User Interface Specification

AMI Patch Evidence Tracker (Synthetic Data)

Student: Euris Garcia

Date: January 2026

1. Overview

The AMI Patch Evidence Tracker is a small web application that helps patch engineers simulate the lifecycle of AMI-based patch events across DEV, STAGE, and PROD. It focuses on:

- Creating patch events for a service, environment, and AMI ID.
- Generating synthetic BEFORE and AFTER vulnerability scan snapshots.
- Computing which vulnerabilities were fixed (DEV evidence).
- Managing the patch lifecycle using a State Pattern.
- Generating STAGE and PROD change-request (CR) summaries.

The application is server-rendered using FastAPI and Jinja2 templates and runs in a web browser. All data displayed is synthetic.

Main user roles:

- Patch engineer / stakeholder (single role for this project).

The interface consists of two main screens:

- Dashboard
- Patch Event Detail

2. Screen 1 – Dashboard

2.1 Purpose

The dashboard provides a high-level overview of all patch events and their lifecycle states, with filtering options and entry points to create or inspect individual events.

2.2 Layout and Elements

- Synthetic data disclaimer banner:
- A prominent banner at the top of the page stating that all data is synthetic and that the application does not connect to real systems.

- Summary statistics bar:
- A horizontal stats section showing:
- Total number of patch events.
- Number of events per environment: DEV, STAGE, PROD.
- Number of events per lifecycle phase: DEV*, STAGE*, PROD*, CLOSED.
- These values are derived from the patch events stored in the local SQLite database and give a quick “at-a-glance” view.

- Filters section:

- **Service filter (dropdown):**

- Lists all synthetic services (e.g., “Nessus Manager”, “Trend Micro”).

- “All” / blank option to disable filtering by service.

- **Environment filter (dropdown):**

- Options: DEV, STAGE, PROD, plus an “All” / blank option.

- State filter (dropdown):

- **Options:**

- DEV_EVIDENCE_CAPTURED

- DEV_VERIFIED

- STAGE_CR_READY

- STAGE_PATCHED

- PROD_CR_READY

- PROD_PATCHED

- CLOSED

- “All” / blank option to disable filtering by state.

- A “Filter” button that reloads the dashboard list with the selected criteria.

- Patch events table:

- Columns:

- Service name

- Environment (DEV/STAGE/PROD)

- AMI ID

- Patch date

- Current lifecycle state (colored badge)

- Link or button to “View details”

- Each row represents one patch event.

- Clicking “View details” opens the Patch Event Detail screen.

- Create Patch Event button:

- A button labeled “Create Patch Event”.

- Navigates to the Patch Event creation form.

2.3 User Interactions

- Filter patch events using any combination of the three filters and click “Filter” to refresh the list.
- Clear filters (by selecting blank / “All”) to see all events.
- Click “Create Patch Event” to add a new event.
- Click a row’s “View details” or equivalent link to open the detail screen.

Inputs:

- Service, environment, and state filter values.

Outputs:

- Updated list of patch events and updated summary stats.

3. Screen 2 – Patch Event Detail

3.1 Purpose

The Patch Event Detail screen shows the full lifecycle and synthetic evidence for a single patch event and allows the user to:

- Generate BEFORE and AFTER synthetic snapshots.

- Compute fixed vulnerabilities (DEV evidence).
- View severity breakdowns.
- Transition the lifecycle state.
- Generate STAGE and PROD CR summaries.

3.2 Layout and Sections

The detail view is divided into structured sections:

A. Patch metadata section

- Displays:
  - Service name
  - Environment (DEV, STAGE, or PROD)
  - AMI ID
  - Patch date
  - Free-text notes
- Provides context for the rest of the information on the page.

B. Lifecycle section

- Displays:
  - Current lifecycle state as a colored badge.
  - A “Next state” dropdown listing only valid next states according to the State Pattern.
  - An “Apply transition” button to commit a state change.
- Behavior:
  - Only allowed transitions appear in the dropdown.

- If an invalid transition is attempted:
  - The system rolls back the change.
  - An error message is shown (e.g., cannot close unless in PROD_PATCHED).

C. Synthetic Scan Evidence section

- BEFORE / AFTER snapshot status:
  - Shows counts such as:
    - “BEFORE snapshot: N vulnerabilities”
    - “AFTER snapshot: M vulnerabilities”
- Buttons:
  - “Generate BEFORE (synthetic)”:
    - Deletes any previous BEFORE snapshot for this event.
    - Creates a new synthetic BEFORE snapshot with fake vulnerabilities (CVE, plugin ID, host, severity).
  - “Generate AFTER (synthetic)”:
    - If a BEFORE snapshot exists:
      - Chooses a random subset of BEFORE vulnerabilities to simulate remaining issues.
    - If BEFORE is missing:
      - A smaller synthetic set is generated.
    - Backend guard:
      - The route ensures a BEFORE snapshot exists and returns a message if called out of order.
  - “Compute fixed vulnerabilities”:
    - Enabled only when both BEFORE and AFTER snapshots exist.
    - Computes the set of fixed vulnerabilities (in BEFORE but not in AFTER).

- Marks DEV evidence as available for this event.

- UI gating:

- "Generate AFTER" button is disabled until a BEFORE snapshot exists.
- "Compute fixed vulnerabilities" button is disabled until both BEFORE and AFTER exist.
- Short helper text below the buttons explains these conditions.

D. Fixed vulnerabilities and severity breakdown

- Once DEV evidence has been computed:
  - A table lists all fixed vulnerabilities with columns:
    - Synthetic ID
    - CVE
    - Plugin ID
    - Severity
    - Host
  - A severity breakdown is displayed:
    - Counts of fixed Critical, High, Medium, and Low vulnerabilities.
  - A simple visual representation (e.g., CSS-based bars) helps users quickly see the distribution by severity.

E. CR summary section

- STAGE CR summary:
  - Button "Generate STAGE CR summary":
    - Enabled when DEV evidence exists and both BEFORE and AFTER snapshots are present.

- Generates synthetic STAGE CR text using the fixed vulnerability data and severity breakdown.
- The generated text is displayed in a read-only text area for easy copying.

- PROD CR summary:

- Button “Generate PROD CR summary”:
  - Enabled only when the event is in an appropriate lifecycle state (for example, after STAGE has been patched).
  - Also requires DEV evidence and both BEFORE and AFTER snapshots.
  - Generates synthetic PROD CR text assuming STAGE validation is complete.
- The text is displayed similarly to the STAGE CR summary.

3.3 User Interactions (Detail Screen)

Typical user flow:

1. Review metadata and initial state.
2. Click “Generate BEFORE (synthetic)” to create a BEFORE snapshot.
3. Click “Generate AFTER (synthetic)” to create an AFTER snapshot.
4. Click “Compute fixed vulnerabilities” to derive DEV evidence.
5. Review:
   - Fixed vulnerabilities table.
   - Severity breakdown.
6. Use the “Next state” dropdown and “Apply transition” to promote the event through the lifecycle.
7. Generate STAGE and then PROD CR summaries when allowed.

Inputs:

- Button clicks for snapshot generation and evidence computation.
- Selected target state for lifecycle transitions.

Outputs:

- Updated counts, tables, severity breakdowns.
- Dev evidence flag.
- STAGE and PROD CR summary text.
- Status messages indicating success or invalid actions.

4. Non-Functional and Safety Considerations

- All data shown in the UI is synthetic:
  - No real environment connections.
  - No production scanners or ticketing systems.
- The UI prominently displays a synthetic data disclaimer in the shared layout.
- Errors (such as invalid lifecycle transitions or missing snapshots) are surfaced as clear user-facing messages on the detail page.