UTIP Stabilization & Enhancement Roadmap

Version: 2.0 (Recovery & Polish)

Date: January 20, 2026

Theme: "Midnight Vulture"

Classification: INTERNAL USE ONLY

---

1. Executive Summary

The Unified Threat Intelligence Platform (UTIP) has achieved a robust backend foundation, with Phases 1–7 (Core API, Vulnerability Pipeline, Intel Worker, Correlation, Attribution, and Remediation) fully operational. However, the project encountered critical friction during Phase 8 (Frontend Integration), specifically regarding authentication "plumbing," basic usability (scrolling), and visualization complexity.

This document outlines the **Corrective Action Plan** to stabilize the platform immediately and the **UX Enhancement Specification** to elevate the interface to a "Wiz-like" standard using the Midnight Vulture design system.

Part 1: The Stabilization Plan ("Getting Back on Track")

We are pivoting from a "feature-complete" goal to a "data-flowing" goal. The priority is to unblock user workflows by fixing plumbing and scoping down complex visualizations.

1.1 Fix the Critical Plumbing (Immediate Priority)

The frontend is currently disconnected from the backend due to authentication mismatches and blocked by a CSS layout bug.

- **Fix JWT Audience Mismatch:**
    - **Issue:** Backend expects `utip-api` audience; Frontend client issues `utip-frontend`. Result: 401 Unauthorized loops.

    - **Remediation:** Update `backend/app/auth/keycloak.py` to accept tokens from the `utip-frontend` client by modifying the `verify_token` function.
- **Resolve the "Glass Ceiling" Scroll Bug:**
    - **Issue:** The `.app-container` class applies `height: 100vh` and `overflow: hidden`, effectively locking the viewport and hiding content below the fold.
    - **Remediation:** Modify `frontend/src/app/app.component.ts` to remove `overflow: hidden` and change `height` to `min-height: 100vh`.
- **Enable Workflow Actions:**
    - **Issue:** "Upload Intel" and "Upload Scan" buttons are disabled in the UI, preventing users from testing the pipeline.
    - **Remediation:** Bind these buttons to the existing API endpoints (`/api/v1/intel/upload` and `/api/v1/vuln/upload`) immediately.

1.2 Security Hardening (Critical Debt)

The Engineering Review flagged critical security lapses that must be addressed before any further feature development.

- **Rotate Compromised Secrets:**
    - **Issue:** `KEYCLOAK_CLIENT_SECRET` and test credentials were committed to documentation.
    - **Action:** Generate new secrets in Keycloak. Remove `PHASE1_COMPLETION_REPORT.md` and `DEPLOYMENT.md` from the repo history or scrub them.
- **Implement Rate Limiting:**
    - **Action:** Add `slowapi` to the FastAPI backend to prevent brute-force attacks on the login and upload endpoints.
- **Enable TLS/HTTPS:**
    - **Action:** Configure Nginx (Phase 8 container) to handle SSL termination. Do not transmit JWTs over plaintext HTTP.

1.3 Strategic Pivot: Visualization Scope

We are pausing the development of the complex "Matrix Grid" (14-column heat map) to focus on immediate data visibility.

- **Descope:** Stop work on the custom MITRE Matrix grid component.
- **New Target:** Enhance the existing **Technique List** view.
    - Group techniques by Tactic (Initial Access, Execution, etc.).
    - Use the existing color coding (Red/Yellow/Blue) which is already implemented.
    - This delivers 80% of the value with 20% of the effort, unblocking the project.

Part 2: UI/UX Enhancement Specification

Once stabilization is complete, we will apply the "Midnight Vulture" design system to achieve a high-fidelity, "cyber-tactical" aesthetic inspired by Wiz and CrowdStrike.

2.1 The "Midnight Vulture" Design System

Philosophy: High-stakes decision-making cockpit. Dark mode only. Data-dense but breathable.

- **Color Palette:**
  - **Void (Background):** `#020617` (Slate-950).
  - **Surface (Cards):** `#0f172a` (Slate-900).
  - **Semantic Intelligence:**
    - 🔴 **Critical (Red):** Overlap (Intel + Vuln) - `#EF4444`.
    - 🟡 **Intel (Amber):** Threat Intel only - `#F59E0B`.
    - 🔵 **Vuln (Blue):** Vulnerability only - `#3B82F6`.
- **Typography:**
  - **UI Elements:** *Inter* (Clean, readable sans-serif).
  - **Data Points:** *JetBrains Mono* (For IPs, Hashes, CVE IDs, Technique IDs).

2.2 Usability Enhancements (The "Wiz" Feel)

To match the fluidity of Wiz, we will implement the following interaction patterns:

- **Contextual Side-Drawers (Not Modals):**
  - Never navigate the user away from the dashboard.
  - **Action:** Clicking a technique opens the **Remediation Sidebar** (already partially implemented). Ensure this sidebar overlays the content smoothly using the "Slide In" animation.
- **Asynchronous "Agentless" Feel:**
  - Uploads should not block the UI.
  - **Action:** When a user uploads a scan, show a "Processing..." toast notification (using the new `ToastComponent`) and let them continue working. Notify them when the layer is ready.
- **Empty State Illustrations:**
  - Replace blank screens with actionable empty states.
  - **Design:** "No Layers Yet? Start by uploading a Nessus scan.".

2.3 Component-Level Polish

Specific improvements identified in the usability audit:

- **Scroll-to-Top Button:** A floating button (`bottom: 2rem; right: 2rem`) that appears after scrolling 300px.
- **Enhanced Focus States:** Replace default browser outlines with a `2px solid var(--color-accent)` glow for keyboard accessibility.
- **Skeleton Loaders:** Refine `app-skeleton-loader` to match the exact layout of the Stats Grid and Layer Cards for a smoother loading experience.
- **Search Input Polish:** Add a "Clear" (X) button inside the search bar and a magnifying glass icon to improve affordance.

---

3. Implementation Roadmap (Next 4 Sprints)

Sprint 1: Stabilization & Plumbing (Week 1)

- **Goal:** User can log in, upload data, and see it in a list.
- [ ] **Fix:** Apply JWT audience patch in Backend.
- [ ] **Fix:** Apply CSS scroll fix in `app.component.ts`.
- [ ] **Fix:** Wire up "Upload" buttons to API endpoints.
- [ ] **Security:** Rotate all secrets and sanitize repo history.

Sprint 2: Core UX Polish (Week 2)

- **Goal:** The application feels professional and responsive.
- [ ] **Feature:** Implement `ToastComponent` for success/error feedback.
- [ ] **Feature:** Add "Scroll to Top" button and enhanced focus states.
- [ ] **Feature:** Refine Skeleton Loaders for Dashboard.
- [ ] **Design:** Enforce "Midnight Vulture" tokens globally (Glassmorphism on all cards).

Sprint 3: Advanced Visualization (Week 3)

- **Goal:** Replace basic lists with richer visual contexts.
- [ ] **Feature:** Enhance "Technique List" to group by Tactic (Columnar view).
- [ ] **Feature:** Polish the "Attribution Panel" with confidence score pulse animations.
- [ ] **Feature:** Polish "Remediation Sidebar" with tabbed views for Mitigations vs. Detection Rules.

Sprint 4: Production Hardening (Week 4)

- **Goal:** Ready for "Internal Use Only" deployment.
- [ ] **Ops:** Implement Rate Limiting on API.
- [ ] **Ops:** Configure TLS termination.
- [ ] **Ops:** Optimize Docker builds (multi-stage).