# AI Security Dashboard – Why We Need It

## Introduction

Our current security dashboard only displays scan results and scores, leaving teams to figure out what matters and how to fix it. AI will make the dashboard intelligent — it will prioritize the most critical issues based on real business impact, remove duplicates across tools, provide exact fix steps, and highlight when one fix can solve multiple problems. This turns the dashboard from a static report into a smart assistant that helps us reduce risk faster, save time, and lower costs.

“AI transforms our security dashboard from a list of problems into a smart assistant that tells us what to fix first, how to fix it, and how to solve the most issues with the least effort.”

## 1. From Results to Action

Current Tool:

“CVE-2023-12345 – SQL Injection – Critical – Score: 9.2”

AI Dashboard:

“CVE-2023-12345 – SQL Injection in /user/login – Critical – Score: 9.2 – Exploitable via login form.”  
Suggested Fix: Use parameterized queries in user.js function.

const query = "SELECT \* FROM users WHERE id = ?";  
db.query(query, [req.params.id], callback);

Value: Devs can copy/paste or adapt the fix immediately — no research time.

## 2. Risk Prioritization

Current Tool:

Two “Critical” vulnerabilities — no guidance on which matters more.

AI Dashboard:

Ranks by business impact:  
1️⃣ Payment API vulnerability – customer card data at risk – exploit in the wild (CISA KEV)  
2️⃣ Internal demo service vulnerability – low business impact

Value: Teams fix the issue that can cause the most real damage first.

## 3. Cross-Tool Correlation & De-Duplication

Current Tool:

Black Duck: lodash@4.17.19 vulnerable.  
  
42Crunch: /processPayment API insecure.  
  
They appear as unrelated issues.

AI Dashboard:

Correlates findings:  
“/processPayment API uses lodash@4.17.19 (CVE-2021-23337). Exploiting the API allows access to payment data.”

Value: One fix resolves both findings — no duplicate work.

## 4. AI-Driven Suggestions

Current Tool:

“Upgrade library to fix vulnerability” — no indication which option is best.

AI Dashboard:

Lists possible fixes, then recommends the safest, fastest, most stable:  
✅ Upgrade to v4.17.21 – low risk, stable (Recommended)  
🔄 Replace with alternative library – high migration effort  
🛑 Patch locally – temporary, not recommended

Value: No guesswork — devs know the best path forward instantly.

## 5. Friendly, Interactive UI

Current Tool:

Static vulnerability list with filters.

AI Dashboard:

Natural language search:  
“Show me all Critical API issues found in the last 30 days in mobile apps.”  
  
Instantly shows:  
- Filtered results  
- Heatmap of risk areas  
- One-click “Fix It” cards

Value: Anyone (security, dev, management) can get insights without learning query syntax.

## 6. Fix-One, Resolve-Many

AI Dashboard:

“Upgrading spring-core fixes 10 transitive dependency CVEs at once.”

Value: Saves hours of manually patching each one individually.

## Audience

Security Engineers → Faster triage, less noise

Developers → Clear fixes, fewer duplicate tickets

Managers → Better tracking and prioritization

Executives / Compliance → Posture at a glance, instant audit reports