# 

### **Objective**

Automate **vulnerability detection and remediation** from Software Composition Analysis (SCA) tools (e.g., **BlackDuck**) using an **Agentic AI system** that integrates with **ServiceNow** for ticket management and **Git** for automated code fixes with **Human-in-the-Loop (HIL)** validation.**Phase 1 – Problem Definition & Scope Alignment**

**Goal:** Establish a clear understanding of objectives, agent roles, and success metrics.

### 

| **#** | **Task** | **Description** |
| --- | --- | --- |
| 1.1 | Define POC objectives | Identify the problem, scope, and measurable outcomes |
| 1.2 | Identify agent roles | Define Parser, Analyzer, Remediator, Git, ServiceNow, and HIL Supervisor agents |
| 1.3 | Establish KPIs | Define performance, accuracy, and autonomy metrics |
| 1.4 | Risk & boundary analysis | Determine safety, compliance, and code modification limits |

### 

## **Phase 2 – Data & Knowledge Preparation**

**Goal:** Prepare vulnerability data and knowledge sources for reasoning and context retrieval.

| **#** | **Task** | **Description** |
| --- | --- | --- |
| 2.1 | Collect SCA reports | Export sample BlackDuck reports (JSON/XML/CSV) |
| 2.2 | Define vulnerability schema | Identify key fields (CVE ID, package, version, severity, fix info) |
| 2.3 | Create preprocessing pipeline | Write parsers to standardize and clean data |
| 2.4 | Build vulnerability embeddings | Use CVE descriptions and fix data to generate vector representations |
| 2.5 | Connect CVE knowledge base | Integrate with public NVD / OSV databases for contextual enrichment |
| 2.6 | Configure ADK data connectors | Feed processed data into the agent environment |

## **Phase 3 – Agent Architecture Design (Google ADK)**

**Goal:** Design the autonomous agent system for parsing, analyzing, remediating, and integrating actions.

### **✅ Tasks**

| **#** | **Task** | **Description** |
| --- | --- | --- |
| 3.1 | Design multi-agent structure | Define agents (Parser, Analyzer, Remediator, Git, ServiceNow, HIL) and their interactions |
| 3.2 | Define reasoning and memory flow | Configure ADK memory (short-term, episodic, long-term) |
| 3.3 | Build agent prompt templates | Create system prompts, roles, and behaviors |
| 3.4 | Integrate tools and APIs | Register APIs (Git, ServiceNow) within ADK |
| 3.5 | Define safety & guardrails | Implement constraints for code edits and workflow execution |
| 3.6 | Document orchestration flow | Map sequence: Parse → Analyze → Remediate → Validate → Integrate |

## **Phase 4 – Prototype Development (FastAPI Backend)**

**Goal:** Implement and deploy backend pipelines to operationalize agent interactions.

### **✅ Tasks**

| **#** | **Task** | **Description** |
| --- | --- | --- |
| 4.1 | Setup FastAPI project | Scaffold backend with versioned routes |
| 4.2 | Implement ADK orchestration | Connect ADK agents to FastAPI endpoints |
| 4.3 | Integrate Git API | Enable branch creation, code commits, and PR generation |
| 4.4 | Integrate ServiceNow API | Enable automated ticket creation & updates |
| 4.5 | Implement HIL workflow | Build approval interface for human validation |
| 4.6 | Add observability hooks | Integrate logging, traces, and agent reasoning monitoring |
| 4.7 | Unit and API testing | Validate agent endpoints & responses |

## **Phase 5 – Experimentation & Simulation**

**Goal:** Validate end-to-end agent workflows in controlled experiments.

### 

| **#** | **Task** | **Description** |
| --- | --- | --- |
| 5.1 | Test vulnerability parsing | Evaluate Parser Agent on various report formats |
| 5.2 | Validate severity classification | Check Analyzer Agent accuracy vs known CVSS levels |
| 5.3 | Test remediation generation | Assess fix suggestions and dependency updates |
| 5.4 | Conduct HIL simulations | Run human feedback cycles for generated patches |
| 5.5 | Measure performance metrics | Record latency, cost, and success rate |
| 5.6 | Iterate improvements | Refine prompts, logic, and API calls |

## **Phase 6 – Evaluation & Validation**

**Goal:** Ensure the POC meets functional, security, and business expectations.

| **#** | **Task** | **Description** |
| --- | --- | --- |
| 6.1 | Evaluate accuracy & reliability | Compare outputs with ground-truth and manual baselines |
| 6.2 | Validate security guardrails | Test restricted Git actions and controlled execution |
| 6.3 | Conduct stress tests | Run agents on large SCA data sets |
| 6.4 | Generate technical report | Consolidate evaluation data & business metrics |
| 6.5 | Go/No-Go decision | Stakeholder review of readiness for MVP phase |

## **Phase 7 – Streamlit Frontend & Demonstration**

**Goal:** Build an interactive, user-facing dashboard to showcase the POC.

### **✅ Tasks**

| **#** | **Task** | **Description** |
| --- | --- | --- |
| 7.1 | Design Streamlit UI layout | Build upload, view, and approval sections |
| 7.2 | Integrate FastAPI endpoints | Connect UI with backend agent endpoints |
| 7.3 | Add real-time visualizations | Display vulnerability trends, severity graphs, PR & ticket status |
| 7.4 | Implement HIL control UI | Add approve/reject workflow for remediations |
| 7.5 | Conduct demo session | Live run from scan to PR generation |
| 7.6 | Gather stakeholder feedback | Capture feedback and improvement areas |