Group P06

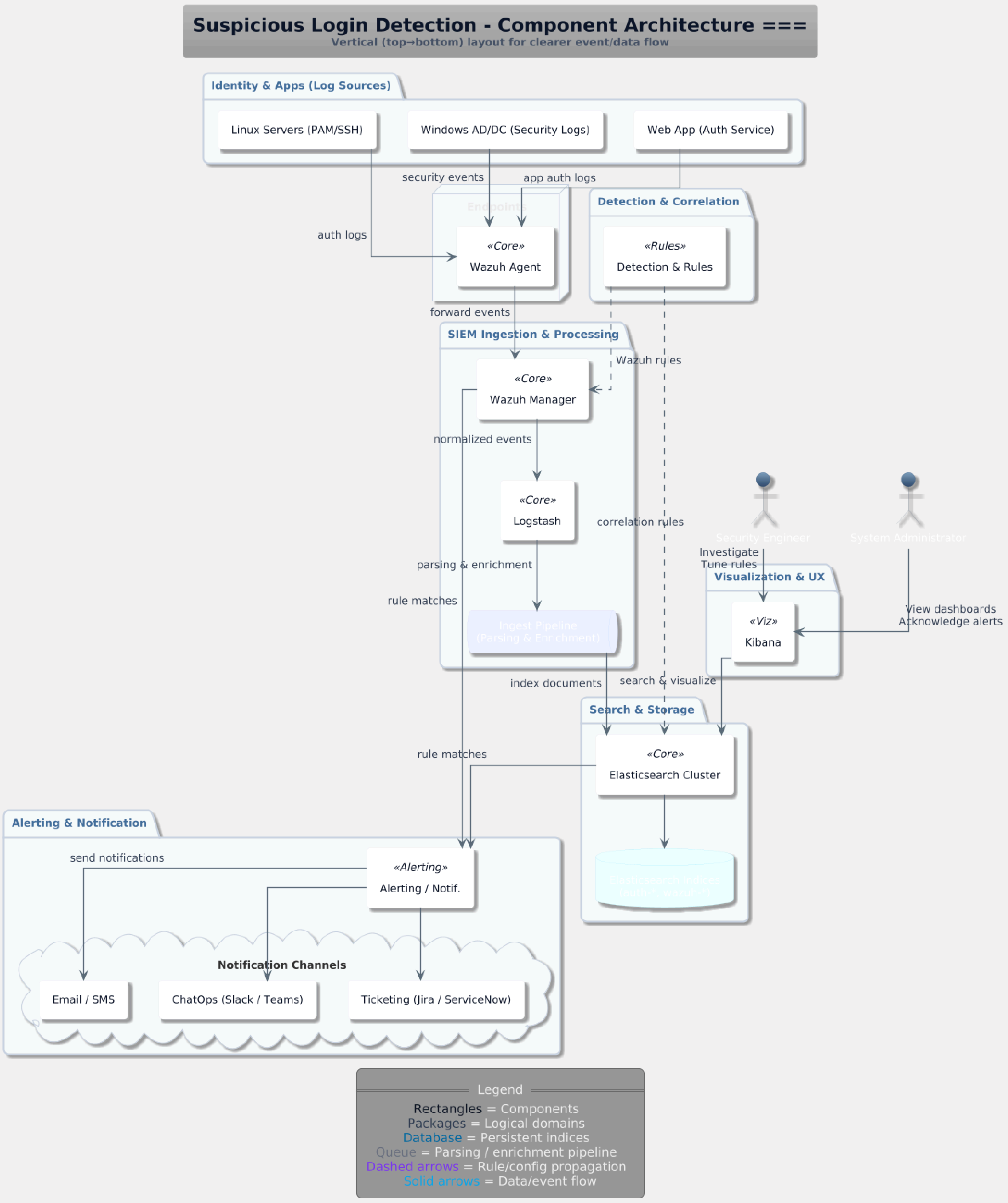
Anomalous Login Detection

Aaffan Niazi, Mustafa Hussain, Muhammad Mustafa, Shehroz Faryad

Overview

- **Objective:** Detect suspicious or unusual login activities in real-time using the ELK Stack (Elasticsearch, Logstash, Kibana) integrated with Wazuh.
- **Function:** Collects authentication logs, detects patterns like multiple failed attempts or logins from abnormal locations.
- **Users:** System administrators and security engineers.
- **Key Benefit:**
- Real-time alerts for abnormal logins.
- Faster detection and response to security threats.
- Centralized dashboard for visualization and monitoring.

System Architecture



System Architecture

- **Core Components:**
- Log Sources (Identity & Apps): Generate authentication/security logs from servers and web apps.
- Wazuh Agent: Collects and forwards logs securely.
- Wazuh Manager + Logstash: Normalize and enrich logs; apply detection rules.
- Elasticsearch Cluster: Stores and indexes events for fast search.
- Kibana: Provides dashboards for analysis and visualization.
- Alerting & Notification System: Sends alerts via Email, SMS, or ChatOps tools.
- **Architecture Features:**
- Modular and scalable design (collection, processing, storage, and visualization layers).
- Real-time data processing pipeline.
- Fault-tolerant and easily extendable structure.

How Architecture Supports Non-Functional Requirements

1. **Performance – Process 10,000 log entries per second**
   - Parallel ingestion from multiple Wazuh Agents.
   - Optimized Logstash enrichment and time-based Elasticsearch indices.
   - Horizontal scalability across ingestion tiers.
2. **Reliability – Recovery within 5 minutes after failure**
   - Fault isolation between collection, processing, and storage tiers.
   - Stateless components enable quick restart and auto-recovery.
   - Health monitoring dashboards in Kibana.
3. **Efficiency – Memory usage under 1 GB**
   - Lightweight event forwarding at the edge.
   - Compact data indices in Elasticsearch.
   - Minimal enrichment to reduce processing load.

How Architecture Supports Security Requirements

1. **Broken Access Control**
   - Kibana is the only user-facing entry point.
   - Role-based access: read-only for analysts, edit rights for engineers.
   - Private network isolation for backend components (Wazuh, Logstash, Elasticsearch).
2. **Input Manipulation Attack**
   - Early validation and normalization by Wazuh Agent and Manager.
   - Strict schema enforcement—reject malformed logs.
   - Alert triggers for suspicious log patterns or volume spikes.
3. **Data Poisoning Attack**
   - Pre-processing and filtering of logs before storage.
   - Separation of recent and archival data to limit contamination.
   - Continuous performance reviews in Kibana to refine detection rules.

Thank You