Software Engineer Test (3-Days)

Part 1 – Systems Design (40 pts)

Scenario:
Your organization runs a **hybrid SOC** using **Wazuh** (integrated with Elastic Search & Kibana). Analysts report that **over 60 % of generated alerts are false positives** from detection rules. You must design a **microservice-based automation module** to reduce false alerts while maintaining detection precision.

Tasks:

1. **Architecture Design (15 pts)**
   Describe (or diagram) a modular architecture that integrates with **Wazuh Manager REST API**.
   - Include services for rule management, log enrichment, and alert correlation.
   - Show interfaces between Wazuh and your automation engine.
   - Explain how feedback loops from analyst validation improve future precision.
2. **Data Flow Explanation (10 pts)**
   Trace the full pipeline:
   Agent log → Wazuh Manager (decoders & rules) → Elastic index → automation module → alert scoring → analyst review.
   Explain how enriched context (e.g., threat intel, geolocation, asset tags) reduces false positives.
3. **Scalability & Fault Tolerance (10 pts)**
   Propose strategies to:
   - Distribute event processing across nodes (Filebeat queues, Kafka buffers).
   - Handle API rate limits and node failures gracefully.

- Maintain system state consistency between Wazuh and the automation layer.

4. **Ethical & Operational Constraints (5 pts)**
   Define safeguards to avoid suppressing true positives and ensure auditable actions (e.g., SOAR playbook logging, rollback mechanisms).

Part 2 – Coding Challenge (30 pts)

Implement a simplified **False-Positive Handler Script** that automates Wazuh alert triage.

Requirements:

- Language: Python / Go
- Use the **Wazuh API (/alerts, /rules, /decoders)** endpoints.
- Implement:
  1. Fetch open alerts with severity ≤ 3 and tags like "test" or "internal".
  2. Automatically mark them as closed or update the corresponding rule with a whitelist entry.
  3. Add a note to the alert record explaining the reason for closure.
  4. Log every action (alert ID, timestamp, reason) to a local SQLite DB or JSON file.

Part 3 – Analytical Case Study (20 pts)

Using metrics from the confusion matrix (Precision, Recall, FPR, FNR):

1. Given: TP (True Positive) = 320, FP (False Positive) = 80, FN (False Negative) = 40, TN (True Negative) = 560.
   Compute Precision, Recall, and F1 Score.
2. In a Wazuh context, explain how doubling the rule sensitivity (e.g., lower threshold on syscheck alerts) impacts precision and analyst fatigue.

3. Suggest one quantitative metric (SOC MTTR, alert volume ratio) and one qualitative method (analyst survey) to measure alert-fatigue reduction after automation deployment.

Part 4 – Behavioral & Design Reasoning (10 pts)

Answer briefly:

- Describe a time you improved a detection pipeline or log processing system.
- How would you validate the impact of automation on SOC performance?
- A stakeholder wants to disable auto-closure after a missed alert incident, how would you handle it technically and diplomatically?