**Functional and Performance Testing Phase**

Exploring Cyber Security: Understanding Threats and Solutions in the Digital Age

Team Members:

1. Dhanraj Pawar
2. Kaushal Chougule
3. Athrav Pawar
4. Balaji Patil

**Functional and Performance Testing Phase Summary**

The Functional and Performance Testing Phase ensures that the system operates as expected and performs efficiently under anticipated workloads. This stage involves evaluating system functionality, scalability, and efficiency to detect and resolve issues before deployment.

**Objectives:**

* Confirm that the system functions as intended.
* Validate the system's capability to manage the anticipated workload.
* Identify and address any performance-related constraints.

**Functional Testing**

Functional testing is conducted to verify that the system aligns with functional requirements. Below are test cases for each requirement:

**1. Vulnerability Assessment**

* **Test Case 1:** Confirm that Nessus successfully conducts a vulnerability scan on the target system.
  + **Steps:**
    1. Install and configure Nessus.
    2. Execute a vulnerability scan on the system.
    3. Verify successful completion and generation of a scan report.
  + **Expected Outcome:** The report should display detected vulnerabilities with severity classifications.
* **Test Case 2:** Ensure the system prioritizes vulnerabilities based on severity.
  + **Steps:**
    1. Analyze scan findings.
    2. Categorize vulnerabilities into critical, high, medium, or low priority.
  + **Expected Outcome:** Vulnerabilities should be accurately prioritized based on severity levels.

**2. Threat Hunting**

* **Test Case 3:** Validate that Splunk monitors SIEM logs for suspicious activities.
  + **Steps:**
    1. Install and configure Splunk.
    2. Observe SIEM logs for unauthorized access or abnormal login attempts.
  + **Expected Outcome:** Splunk should successfully detect and log suspicious activities.
* **Test Case 4:** Verify that the system investigates and escalates potential threats.
  + **Steps:**
    1. Examine a suspicious login attempt.
    2. Escalate the issue to the incident response team upon confirming a threat.
  + **Expected Outcome:** The threat should be escalated and documented in an incident report.

**3. Incident Response**

* **Test Case 5:** Confirm the system’s ability to analyze phishing emails for Indicators of Compromise (IOCs).
  + **Steps:**
    1. Gather phishing emails from spam folders or simulated attacks.
    2. Analyze email headers and content for phishing signs.
  + **Expected Outcome:** The system should successfully identify IOCs in phishing emails.
* **Test Case 6:** Ensure that the system can generate a comprehensive incident report.
  + **Steps:**
    1. Document an incident report detailing a phishing attack.
    2. Include IOCs and suggested mitigation steps.
  + **Expected Outcome:** The incident report should be complete and precise.

**Performance Testing**

Performance testing ensures that the system operates optimally under expected conditions. Below are the test cases:

**1. Scalability Testing**

* **Test Case 7:** Validate the system’s capability to handle large volumes of data.
  + **Steps:**
    1. Simulate a substantial volume of logs and scan data.
    2. Monitor system performance.
  + **Expected Outcome:** The system should process large data volumes without performance degradation.

**2. Response Time Testing**

* **Test Case 8:** Ensure real-time alerts are generated for suspicious activity.
  + **Steps:**
    1. Simulate unusual activities, such as abnormal login attempts.
    2. Measure the time taken to generate an alert.
  + **Expected Outcome:** The system should trigger alerts in real time.

**3. Load Testing**

* **Test Case 9:** Verify the system’s ability to support multiple concurrent users.
  + **Steps:**
    1. Simulate multiple users accessing the system simultaneously.
    2. Monitor the system’s response and performance.
  + **Expected Outcome:** The system should manage concurrent users without performance loss.

**Test Environment**

Testing will take place in a controlled environment that replicates the production setup. The environment consists of:

1. **Virtual Machines** – Running Nessus, Splunk, and other tools.
2. **Test Data** – Simulated logs, scan results, and phishing emails.
3. **Monitoring Tools** – Used to track system performance throughout testing.

**Deliverables**

1. **Test Cases:**
   * A comprehensive list of functional and performance test scenarios.
2. **Test Results:**
   * Documented results from functional and performance evaluations.
3. **Defect Report:**
   * A log of identified defects and their resolutions.
4. **Performance Metrics:**
   * Key system performance indicators (e.g., response times, scalability levels).

**Next Steps**

1. **Documentation & Demonstration:**
   * Compile test documentation and prepare a system demonstration.
2. **Deployment:**
   * Move the system to the production environment.