Exercise 3: Real-Time Cybersecurity Defense System (Mutex, Semaphore, Message Passing, and Queues)

Scenario

You are tasked with designing and implementing a real-time cybersecurity monitoring and active defense system for a critical infrastructure facility. This system operates in a high-stakes environment where malicious actors constantly target networks, servers, and connected IoT devices. The system must not only detect and respond to cyber threats in real-time but also coordinate actions across multiple layers of defense while avoiding downtime or false positives. The system comprises multiple interdependent components, each handling distinct responsibilities:

High-Speed Packet Sniffer
Monitors multiple network interfaces in real-time. Simultaneously processes packets from three priority streams:

1. Critical (e.g., SCADA control systems).
2. Normal (e.g., employee devices).
3. Low (e.g., visitor networks).

Places packets into separate priority queues for threat analysis.

- **Distributed Threat Analyzer**
  Runs three parallel analyzers (one for each priority stream).
  Uses a mutex-protected database of threat patterns to classify packets as safe, suspicious, or malicious.
  Updates the database in real-time with new intelligence from external threat feeds, triggering a system-wide lock during updates.
  Flagged threats are sent to the response handler with severity levels and metadata.

- **Response Handler**
  Takes immediate action based on flagged threats:
  1. Low Threat: Logs and monitors.
  2. Medium Threat: Quarantines devices or IPs (uses semaphores to avoid simultaneous access to shared quarantine resources).
  3. High Threat: Sends shutdown commands to compromised systems, triggers alarms, and updates the logger.

  Coordinates with external incident response APIs (simulated via message passing).

- **Logger**
  Logs all actions, including sniffed packets, analysis results, responses, and system errors.
  Operates as a dedicated low-priority task to avoid interrupting real-time operations.

- **Watchdog and Error Recovery System**
  Monitors task execution for missed deadlines or crashes & detects:
  1. Queue overflow/underflow.
  2. Deadlocks when multiple analyzers attempt to access the threat database simultaneously.
  3. Packet loss due to sniffer overload.

  Restarts or adjusts tasks dynamically to recover from faults.

Requirements

Multi-Priority Queues:

1. Implement separate FreeRTOS queues for Critical, Normal, and Low priority packets.
2. Ensure the high-priority queue is processed first in the event of congestion.

Use message passing to send flagged threats from the analyzers to the response handler with metadata (packet ID, severity, and suggested actions).

Protect access to the threat database when multiple analyzers need to reference or update it.

Control access to shared quarantine resources to avoid simultaneous write conflicts.

Detect and log queue overflow/underflow conditions dynamically.

Implement backpressure to prevent sniffer overload by dropping low-priority packets when critical queues are full.

Hints for Implementation

Use timeouts on mutexes and semaphores to detect and resolve deadlocks.