Contents

[4.1] Fear-based decision paralysis

1. Operational Definition: A state of cognitive shutdown and behavioral inaction triggered by intense fear or anxiety associated with a security threat, preventing an analyst from executing critical response procedures.

2. Main Metric & Algorithm:

- **Metric:** Fear-based Decision Paralysis Rate (FDPR). Formula: `FDPR = (Number of incidents with no action taken despite high confidence) / (Total number of high-confidence incidents)`.
- **Pseudocode:**

  python

  ```
  def calculate_fdpr(incident_logs, comms_logs, confidence_threshold=0.8):
      """
      incident_logs: from SIEM/SOAR, with analyst actions
      comms_logs: from chat/email, to gauge sentiment
      """
      # 1. Identify high-confidence incidents (e.g., confidence score from ML model > thresh
      high_confidence_incidents = [i for i in incident_logs if i.confidence_score >= confide

      paralyzed_count = 0
      for incident in high_confidence_incidents:
          # 2. Check if no decisive action was taken within a critical time window (e.g., 1
          actions_taken = get_actions_for_incident(incident.id, incident.created_time + time
          decisive_actions = [a for a in actions_taken if a.type in ['contain', 'isolate', '

          # 3. Check comms for fear indicators if no action was taken
          if len(decisive_actions) == 0:
              relevant_comms = get_comms_for_incident(comms_logs, incident.id)
              # Use sentiment analysis or keyword matching on fear/anxiety
              if contains_fear_indicators(relevant_comms):
                  paralyzed_count += 1

      # 4. Calculate FDPR
      total_high_conf = len(high_confidence_incidents)
      FDPR = paralyzed_count / total_high_conf if total_high_conf > 0 else 0
      return FDPR
  ```

- **Alert Threshold:** `FDPR > 0.15` (Paralysis occurs in >15% of high-confidence incidents)

3. Digital Data Sources (Algorithm Input):

- **SIEM/SOAR APIs:** For incident logs, confidence scores, and action timelines.

- **Communication Platform APIs (Slack, Teams):** To access channel/message data related to incident IDs for sentiment analysis.
- **Natural Language Processing (NLP) Model:** Pre-trained model for sentiment analysis (e.g., detecting anxiety, fear) or a keyword list (`"panic"`, `"don't know what to do"`, `"scared"`, `"overwhelmed"`).

4. Human-to-Human Audit Protocol: During tabletop exercises or red team debriefs, use guided questioning: “When the [simulated critical event] occurred, describe what you were feeling. What was going through your mind in the first few minutes? What prevented you from taking immediate action?” Look for verbal and non-verbal cues of freeze response.

5. Recommended Mitigation Actions:

- **Technical/Digital Mitigation:** Implement “panic button” scripts in the SOAR platform that, when triggered, execute a standardized containment playbook to automate initial response steps and break the paralysis.
- **Human/Organizational Mitigation:** Integrate stress inoculation training into tabletop exercises, deliberately simulating high-pressure scenarios in a controlled environment to build resilience.
- **Process Mitigation:** Design incident response playbooks with very simple, binary decision trees for the first 15 minutes (e.g., “If X, then run Containment Script A. If Y, then immediately call Supervisor.”) to reduce cognitive load during a crisis.