Contents

[7.3] Fight Response Aggression

1. Operational Definition: A stress-induced response characterized by hostile or overly forceful interactions with systems, tools, or colleagues, often leading to procedural bypasses, configuration errors, or a toxic team environment.

2. Main Metric & Algorithm:

- **Metric: Aggressive Interaction Rate (AIR)**. Formula: `AIR = (N_aggressive_commands + N_hostile_messages) / N_total_interactions`.
- **Pseudocode:**

  python

  ```
  def calculate_air(employee_id, start_date, end_date):
      # Query command line history for aggressive patterns (e.g., force flags, bulk deletion
      cmd_logs = query_siem(index='linux_audit', search=f'user:{employee_id} (rm * -rf | kil
      n_aggressive_cmds = count(cmd_logs)

      # Query communication platforms for hostile language
      messages = query_teams_api(employee_id, start_date, end_date)
      hostile_keywords = ["idiot", "useless", "why bother", "broken", "fix it now", "dumb"]
      n_hostile_msgs = count_messages_containing(messages, hostile_keywords)

      # Get total interaction count for normalization
      total_cmds = query_siem(index='linux_audit', search=f'user:{employee_id}', result_coun
      total_msgs = count(messages)
      total_interactions = total_cmds + total_msgs

      if total_interactions > 0:
          air = (n_aggressive_cmds + n_hostile_msgs) / total_interactions
      else:
          air = 0
      return air
  ```

- **Alert Threshold:** `AIR > 0.05` (5% of interactions are aggressive) over a one-week period.

3. Digital Data Sources (Algorithm Input):

- **SIEM (e.g., Splunk):** Index `linux_audit` or `windows_events`, fields `user`, `command`.
- **Communication Platform API (e.g., Microsoft Graph API for Teams):** `sender`, `body`, `timestamp`.

4. Human-to-Human Audit Protocol: Team lead observation and anonymous 360-degree feedback from peers. Sample questions: "Have you observed a colleague bypassing security controls out of frustration?" "Does anyone on the team use hostile or demeaning language when systems fail?"

5. Recommended Mitigation Actions:

- **Technical/Digital Mitigation:** Implement technical safeguards like `--no-preserve-root` protection on `rm` commands and require dual approval for critical, destructive actions.
- **Human/Organizational Mitigation:** Provide conflict resolution and stress management training. Foster a blameless post-mortem culture.
- **Process Mitigation:** Introduce a "cool-down" protocol where a frustrated analyst can hand over a task to a colleague without penalty.