Contents

[3.4] Liking-Based Trust Override

1. Operational Definition: The vulnerability where a positive rapport, perceived similarity, or personal liking for a requester causes an individual to bypass security controls or procedures they would normally follow.

2. Main Metric & Algorithm:

- **Metric: Affinity Bypass Rate (ABR)**. Formula: `ABR = N_bypass / N_requests`, where `N_requests` is the number of requests from a "liked" entity (e.g., a frequent collaborator) and `N_bypass` is how many of those led to a policy bypass.
- **Pseudocode:**

  python

```
def calculate_abr(access_logs, chat_logs, hr_data):
    """
    hr_data: Data from HR system to map organizational structure.
    """
    abr_results = {}
    # 1. Identify "high-affinity" pairs: users who frequently interact and are in the same
    affinity_pairs = find_high_affinity_pairs(chat_logs, hr_data, min_interactions=10)

    for user_a, user_b in affinity_pairs:
        # 2. Get requests from user_b to user_a
        requests = get_requests_from_user(chat_logs, user_b, user_a)
        bypass_count = 0
        for req in requests:
            # 3. Check if the request led to a security bypass (e.g., file share without a
            if led_to_bypass(access_logs, user_a, req):
                bypass_count += 1

        total_requests = len(requests)
        ABR = bypass_count / total_requests if total_requests > 0 else 0
        abr_results[(user_a, user_b)] = ABR
    return abr_results
```

- **Alert Threshold:** `ABR > 0.3` (Over 30% of requests from a high-affinity colleague result in a bypass).

3. Digital Data Sources (Algorithm Input):

- **Communication Platform API (Slack/Teams):** To measure interaction frequency and content between users. Fields: `user_from`, `user_to`, `message_count`, `sentiment_score` (if available).
- **HR Database API:** To understand formal reporting structures and team memberships.

- **Access Logs (SharePoint, GitHub, File Servers):** To verify if sharing actions complied with policy (e.g., was an approval ticket referenced?). Fields: `user`, `action`, `target`, `timestamp`.

4. Human-to-Human Audit Protocol: Review access approval tickets. For a sample of approvals, interview the grantor: "You approved X request for [Colleague Y]. Can you walk me through your thought process? Did the fact that you know and work with them closely make the approval easier or faster than if it came from a stranger?".

5. Recommended Mitigation Actions:

- **Technical/Digital Mitigation:** Implement mandatory fields in access request workflows that force the grantor to cite the business justification and policy rule that applies, regardless of the requester.
- **Human/Organizational Mitigation:** Role-play training scenarios where a well-liked colleague makes an inappropriate request, teaching staff how to say "no" gracefully while maintaining rapport.
- **Process Mitigation:** Introduce a "four-eyes" principle for certain high-risk actions requested by individuals outside of a predefined, formal workflow (like a ticketing system).