Contents

[1.1] Unquestioning compliance with apparent authority

1. Operational Definition: A vulnerability where security personnel automatically comply with requests from individuals perceived as authority figures, without performing necessary verification checks, increasing the risk of successful social engineering attacks like CEO fraud.

2. Main Metric & Algorithm:

- **Metric:** Unverified Authority Request Rate (UARR). Formula: `UARR = (Number of unverified high-impact requests from "authorities") / (Total number of high-impact requests from "authorities")`.
- **Pseudocode:**

  python

```
def calculate_uarr(access_logs, comms_data, start_date, end_date):
    """
    access_logs: List of logs from network/access control systems
    comms_data: List of messages from email/chat platforms
    """
    # 1. Identify potential authority figures (e.g., C-level, VPs) from HR data or comms p
    authority_users = identify_authority_figures(comms_data)

    # 2. Extract all high-impact requests from these users (e.g., password reset, access g
    authority_requests = [
        req for req in extract_requests(comms_data, start_date, end_date)
        if req.sender in authority_users and req.impact == 'high'
    ]

    # 3. Cross-reference with access logs to check for verification (e.g., ticket ID, 2FA)
    unverified_count = 0
    for req in authority_requests:
        # Check if the request was followed by an action without a linked, approved ticket
        corresponding_actions = find_corresponding_actions(access_logs, req)
        if not was_request_verified(corresponding_actions):
            unverified_count += 1

    # 4. Calculate UARR
    total_requests = len(authority_requests)
    UARR = unverified_count / total_requests if total_requests > 0 else 0
    return UARR
```

- **Alert Threshold:** `UARR > 0.2` (More than 20% of high-impact authority requests are executed without verification)

3. Digital Data Sources (Algorithm Input):

- **Email/Chat Platforms (Microsoft Graph API, Slack API):** To extract requests containing high-impact keywords (`"need access to"`, `"reset password for"`, `"urgent"`) from senders identified as authority figures.
- **IAM/Active Directory Logs:** To verify if a requested action was performed.
- **Ticketing System API (Jira, ServiceNow):** To check if an action was linked to an approved, validated ticket.

4. Human-to-Human Audit Protocol: Conduct a simulated phishing exercise targeting SOC analysts with a message impersonating a known company executive (e.g., CISO) requesting a high-impact action like disabling a security control. Measure the percentage of analysts who comply without seeking secondary verification via a different channel (e.g., a phone call).

5. Recommended Mitigation Actions:

- **Technical/Digital Mitigation:** Implement a mandatory technical workflow that blocks high-impact actions (e.g., privileged access grants) requested via chat/email unless they are linked to an approved ticket number in the system.
- **Human/Organizational Mitigation:** Conduct targeted training on authority bias, focusing on established verification procedures for all requests, regardless of the perceived seniority of the requester.
- **Process Mitigation:** Formalize a "4-eyes principle" in the SOC playbook for any action requested outside the ticketing system, requiring mandatory confirmation by a shift supervisor.