Contents

[3.2] Trappole di Escalation dell'Impegno

1. Definizione Operativa: Un bias cognitivo in cui gli individui, dopo aver fatto un impegno iniziale piccolo a una richiesta (anche benigna), sono più propensi a conformarsi a una richiesta successiva più grande che viola i protocolli di sicurezza. Questo crea una trappola in cui la sicurezza viene gradualmente erosa.

2. Metrica Principale e Algoritmo:

- **Metrica: Rapporto di Richiesta Escalata (ERR)**. Formula: `ERR = N_escalata / N_iniziale`, dove `N_iniziale` è il numero di richieste iniziali concesse e `N_escalata` è il numero di quelle seguite da una richiesta escalata rilevante per la sicurezza.
- **Pseudocodice:**

  python

```
def calculate_err(access_logs, chat_logs, user_id, time_window='7d'):
    """
    access_logs: Log da IAM, security endpoint, o cloud console.
    chat_logs: Log dalle piattaforme di comunicazione come Slack o Teams.
    time_window: Il periodo da analizzare.
    """
    # 1. Trova richieste iniziali apparentemente benigne (ad esempio, "puoi condividere qu
    initial_requests = query_chat_logs(chat_logs, keywords=["puoi condividere", "mandami",

    escalated_count = 0
    # 2. Per ogni richiesta iniziale, verifica se è stata seguita da un'escalation legata
    for req in initial_requests:
        subsequent_msgs = get_messages_after(chat_logs, req.timestamp, window_hours=48)
        # Cerca richieste escalate (ad esempio, privilegi più alti, disabilita MFA, cambia
        if contains_escalation_keywords(subsequent_msgs, ["diritti admin", "disabilita MFA
            # 3. Verifica se l'azione escalata è stata eseguita controllando i log di acce
            if action_performed(access_logs, user_id, escalated_action, req.timestamp + de
                escalated_count += 1

    initial_count = len(initial_requests)
    ERR = escalated_count / initial_count if initial_count > 0 else 0
    return ERR
```

- **Soglia di Allerta:** `ERR > 0.2` (Più del 20% delle concessioni iniziali portano a un'escalation).

3. Fonti di Dati Digitali (Input dell'Algoritmo):

- **API Piattaforma di Comunicazione (Slack/Teams):** Per cercare pattern di richieste in messaggi diretti e chat di gruppo. Campi: `user`, `message_text`, `timestamp`, `channel_type`.
- **Log Identity & Access Management (IAM) (ad esempio, Azure AD, Okta):** Per verificare i cambiamenti di privilegi. Campi: `user`, `action` (ad esempio, `assignRole`,

`disableMFA`), `target_resource`, `timestamp`.

- **Log Endpoint/Cloud Security (ad esempio, CrowdStrike, AWS CloudTrail):** Per verificare le azioni che disabilitano la sicurezza. Campi: `user`, `event_name` (ad esempio, `StopLogging`, `DeleteAlarm`), `timestamp`.

4. Protocollo di Audit Umano-Umano: Condurre un esercizio di social engineering simulato (con approvazione etica precedente) in cui un membro del red team effettua una piccola richiesta iniziale (ad esempio, chiedere il link di un documento) e in seguito segue con una richiesta che richiede un minore bypass della policy. Misura il tasso di successo. In alternativa, nei colloqui, chiedi: "Qualcuno ti ha mai chiesto un piccolo favore che poi si è trasformato in una richiesta molto più grande che riguarda la sicurezza? Puoi descrivere quello che hai fatto?".

5. Azioni di Mitigazione Consigliate:

- **Mitigazione Tecnica/Digitale:** Implementa l'automazione del flusso di lavoro che richiede l'approvazione del manager per azioni specifiche ad alto rischio (ad esempio, assegnazione di ruoli, modifiche alle policy di sicurezza) indipendentemente da come è stata effettuata la richiesta.
- **Mitigazione Umana/Organizzativa:** Condurre formazione incentrata sul principio di "impegno e coerenza", insegnando al personale a stare attento alle richieste che escalano in privilegi o sensibilità e a segnalarle.
- **Mitigazione del Processo:** Stabilisci un chiaro protocollo che qualsiasi richiesta legata alla sicurezza, indipendentemente da quanto piccola o da chi provenga, deve essere instradata attraverso il sistema di ticketing ufficiale (Jira, ServiceNow) per il tracciamento e l'approvazione.