Orchestratore di Sicurezza con Intelligenza Psicologica: Un'Architettura Multi-Agente per il Rilevamento e la Risposta alle Vulnerabilità in Tempo Reale

Un Preprint

Giuseppe Canale, CISSP
Ricercatore Indipendente
g.canale@cpf3.org
URL: cpf3.org
ORCID: 0009-0007-3263-6897

Dr. Kashyap Thimmaraju
FlowGuard Institute
kashyap.thimmaraju@flowguard-institute.com
URL: flowguard-institute.com
ORCID: 0009-0006-1507-3896

Gennaio 2026

Abstract

Il monitoraggio della sicurezza tradizionale tratta i fattori umani come variabili esterne che richiedono valutazioni periodiche. Presentiamo un cambio di paradigma: un sistema multi-agente che monitora, analizza e risponde continuamente alle vulnerabilità psicologiche in tempo reale. Costruito sul Cybersecurity Psychology Framework, il sistema impiega un agente orchestratore che mantiene una matrice di stato psicologico e attiva dinamicamente agenti specialisti per indagini approfondite quando emergono vulnerabilità convergenti. Un approccio di filtraggio ibrido riduce i costi dei LLM dell'85% mantenendo la qualità del rilevamento (accuratezza e completezza): regole deterministiche gestiscono gli aggiornamenti di stato di routine, con agenti attivati solo per anomalie significative. L'orchestratore implementa un framework decisionale a quattro livelli (monitora, indaga, allerta, critico) con logica di escalation esplicita e analisi delle tendenze temporali per l'allerta precoce. Un modulo di apprendimento adattivo raffina continuamente le soglie di rilevamento e scopre nuovi pattern di vulnerabilità dal feedback degli incidenti. Questa architettura trasforma il monitoraggio passivo in intelligenza attiva, raggiungendo un tasso di rilevamento dell'81% con 47 ore di anticipo sugli incidenti storici a $0,26/utente/mese. Introduciamo il concetto di *Orchestratore di Sicurezza*: un sistema AI che comprende le vulnerabilità psicologiche umane tanto profondamente quanto le superfici di attacco tecniche, e forniamo linee guida complete per il prompt engineering per abilitare il ragionamento efficace degli agenti su stati psicologici complessi.

Parole chiave: cybersecurity, multi-agent systems, LLM agents, prompt engineering, human factors, vulnerability assessment, psychological security, adaptive learning, anomaly detection

1 Introduzione

I fattori umani causano l'85% delle violazioni della sicurezza riuscite [28], tuttavia le risposte organizzative rimangono reattive e periodiche. Ricerche estese documentano come i fattori psicologici guidino i fallimenti della sicurezza, inclusi gli effetti dello stress e del benessere tra i professionisti del SOC [27]. Le valutazioni trimestrali della sicurezza catturano le vulnerabilità psicologiche attraverso istantanee statiche che diventano obsolete entro giorni mentre le condizioni cambiano [23]. Ciò che serve non è una valutazione più frequente ma un'intelligenza continua—un sistema che monitori lo spazio degli stati psicologici con la stessa vigilanza del traffico di rete, rilevi vulnerabilità emergenti prima dello sfruttamento e orchestri risposte appropriate [33].

Il Cybersecurity Psychology Framework (CPF) [5] identifica 100 indicatori di vulnerabilità pre-cognitivi attraverso dieci categorie psicologiche, radicati in teoria psicologica consolidata [19, 14, 7]. Lavori precedenti [4] hanno operazionalizzato questi indicatori attraverso formulazioni matematiche. Tuttavia, tradurre la teoria psicologica in sistemi operativi richiede più di algoritmi di rilevamento; richiede *ragionamento* su stati umani complessi e in evoluzione e decidere quando e come investigare [24].

Un aspetto critico ma trascurato è rilevare quando gli utenti legittimi diventano vulnerabili alla manipolazione esterna a causa di stati psicologici. Questo è un problema di lunga data che non è stato risolto efficacemente. I programmi di formazione sulla consapevolezza della sicurezza, ad esempio, sono in gran parte inefficaci secondo i professionisti del settore [1]—non riescono a riconoscere che la vulnerabilità psicologica è dinamica e dipendente dal contesto, non una carenza di conoscenza statica. Con l'introduzione di LLM agentici nelle organizzazioni, questo problema si amplifica: gli agenti stessi sono inclini alla manipolazione psicologica, sia da parte di agenti malevoli che di umani malevoli. La superficie di attacco si espande dagli umani vulnerabili ai sistemi umano-agente vulnerabili, rendendo il monitoraggio continuo dello stato psicologico ancora più urgente.

I recenti progressi nei Large Language Models (LLM) abilitano un'architettura fondamentalmente diversa: sistemi multi-agente dove agenti AI specializzati collaborano per risolvere problemi complessi [32, 29]. Presentiamo un tale sistema dove un orchestratore monitora continuamente una matrice di stato psicologico, rileva anomalie e pattern di convergenza, e attiva dinamicamente agenti specialisti per investigare e rispondere. Questo approccio passa da "rileva-e-allerta" a "comprendi-e-agisci", creando quello che definiamo un *Orchestratore di Sicurezza con Intelligenza Psicologica*.

1.1 Contributi

I nostri contributi principali sono:

1. Un'architettura multi-agente innovativa con orchestratore e agenti specialisti per il monitoraggio continuo delle vulnerabilità psicologiche
2. Una matrice di stato psicologico come base di conoscenza condivisa con dinamiche di decadimento esponenziale che modellano la persistenza psicologica
3. Un framework decisionale a quattro livelli (monitora, indaga, allerta, critico) con logica di escalation esplicita che determina quando attivare gli agenti e quando allertare le operazioni di sicurezza
4. Algoritmi di analisi delle tendenze temporali che rilevano pattern di escalation, accelerazione e finestre di vulnerabilità—fornendo allerta precoce prima che gli stati psicologici raggiungano soglie critiche

5. Un approccio di filtraggio ibrido che combina regole deterministiche (85% degli eventi) con ragionamento basato su LLM (15%) per l'efficienza dei costi

6. Un modulo di apprendimento adattivo che raffina continuamente le soglie di rilevamento, scopre nuovi pattern di vulnerabilità dal feedback degli incidenti e si auto-ottimizza senza regolazione manuale

7. Linee guida complete di prompt engineering per abilitare il ragionamento efficace degli agenti sugli stati psicologici

8. Meccanismi di ciclo di feedback che abilitano l'apprendimento a livello di sistema dai risultati

2 Background e Lavori Correlati

2.1 Fattori Umani nella Cybersecurity

Ricerche estese documentano come i fattori psicologici guidino i fallimenti della sicurezza. La conformità all'autorità [19], i bias cognitivi [14] e l'influenza sociale [7] creano vulnerabilità sfruttabili. Gli attacchi di ingegneria sociale sfruttano questi principi sistematicamente [11, 26, 10]. La ricerca documenta gli effetti dello stress e del benessere tra i professionisti del SOC sui risultati della sicurezza [27].

La formazione tradizionale sulla consapevolezza della sicurezza mostra efficacia limitata [1, 25]. La valutazione periodica cattura le vulnerabilità a punti discreti ma manca la natura dinamica degli stati psicologici [23]. Lavori recenti sostengono il passaggio da mentalità "umano-come-problema" a "umano-come-soluzione" [33], ma i framework operativi rimangono elusivi.

2.2 Analisi Comportamentale e Rilevamento delle Anomalie

I sistemi User and Entity Behavior Analytics (UEBA) impiegano metodi statistici per rilevare anomalie [6]. Soluzioni commerciali come Exabeam e Splunk UBA identificano deviazioni dal comportamento baseline. Tuttavia, questi sistemi mancano di fondamenta psicologiche—rilevano *cosa* è cambiato senza comprendere *perché* o predire *cosa verrà dopo*.

Il rilevamento delle minacce interne si concentra sull'intento malevolo [15, 21]. Il nostro lavoro affronta un problema complementare: rilevare quando gli utenti *legittimi* diventano vulnerabili alla manipolazione esterna a causa di stati psicologici.

2.3 Sistemi Multi-Agente Basati su LLM

I recenti progressi nei LLM abilitano sistemi multi-agente sofisticati. Xi et al. [32] recensiscono agenti basati su LLM per il completamento autonomo di compiti. Wang et al. [29] esplorano architetture di agenti per la risoluzione di problemi complessi. Questi lavori si concentrano sul completamento generale dei compiti; noi estendiamo gli approcci agentici alle operazioni di sicurezza dove gli agenti devono ragionare sulle vulnerabilità psicologiche.

Il prompt engineering è emerso come critico per l'efficacia degli agenti [30]. Tuttavia, la maggior parte del lavoro si concentra su prompt di completamento dei compiti. Contribuiamo pattern di prompt specifici del dominio per il ragionamento sulla sicurezza psicologica.

3 Panoramica dell'Architettura

3.1 Concetto Base

Il sistema mantiene una **matrice di stato psicologico** continua $M[u][i]$ che traccia il livello di attivazione (0–100) per ogni utente u e indicatore CPF i. Un **agente orchestratore** monitora continuamente questa matrice, rileva cambiamenti significativi o pattern convergenti, e attiva **agenti specialisti** per indagini mirate. Gli agenti specialisti analizzano specifiche sorgenti dati (email, log, contesto organizzativo), riportano risultati e aggiornano la matrice. L'orchestratore impara dai risultati, raffinando il suo processo decisionale nel tempo.

3.2 Componenti del Sistema

L'architettura comprende cinque strati:

Strato 1 — Sorgenti Dati: Flussi in tempo reale da gateway email, log SIEM, sistemi di autenticazione e strumenti di collaborazione alimentano il sistema continuamente.

Strato 2 — Processore di Stream e Filtro Deterministico: Gestisce l'85% degli eventi attraverso logica puramente deterministica: aggiornamenti di decadimento esponenziale, regole di soglia semplici e calcoli baseline. Solo anomalie significative passano all'orchestratore.

Strato 3 — Matrice di Stato Psicologico CPF (Core): La struttura dati centrale $M[user][indicator]$ memorizza i livelli di attivazione (0–100) per ogni utente e indicatore CPF. Questa matrice è la fonte di verità per lo stato psicologico, incorporando il decadimento esponenziale e il tracciamento dei pattern di convergenza.

Strato 4 — Agente Orchestratore: Monitora la matrice continuamente, rileva pattern anomali e vulnerabilità convergenti, e decide quando attivare agenti specialisti. Solo il 15% degli eventi attiva la valutazione dell'orchestratore, e approssimativamente il 30% di questi risulta in attivazione di agenti.

Strato 5 — Agenti Specialisti (On-Demand): Quando attivati dall'orchestratore, gli agenti specialisti analizzano specifiche sorgenti dati (EmailAnalyzer, SOCLogAnalyzer, ContextGatherer) e riportano risultati con aggiornamenti della matrice raccomandati e contromisure.

Integrazione e Apprendimento: Lo Strato di Integrazione SOC consegna allerte con contesto agli analisti di sicurezza, che forniscono feedback che fluisce in un database di apprendimento. Questo abilita il raffinamento continuo dei pattern decisionali.

3.3 Matrice di Stato Psicologico

La matrice mantiene stato per-utente, per-indicatore con decadimento esponenziale:

$$M[u][i](t) = \max\left(\alpha \cdot M[u][i](t - \Delta t), X_i(t)\right) \tag{1}$$

dove $X_i(t)$ è il valore osservato dell'indicatore i al tempo t, $\alpha = e^{-\Delta t/\tau_i}$ è il fattore di decadimento, e τ_i è l'emivita psicologica dell'indicatore i. L'operazione max assicura che osservazioni acute elevino immediatamente lo stato.

La Tabella 1 specifica le costanti temporali per categoria, derivate dalla letteratura psicologica sulla persistenza degli stati [8, 20, 2, 13].

Table 1: Costanti Temporali Psicologiche per Categoria CPF

Categoria	τ **(half-life)**	**Razionale**
1.x Authority	4 hours	Contestuale, transitorio [19]
2.x Temporal	2 hours	Urgenza guidata da scadenze
3.x Social	24 hours	Persistenza interpersonale [7]
4.x Affective	12 hours	Regolazione emotiva [8]
5.x Cognitive	8 hours	Fatica basata su turni [20]
6.x Group	14 days	Cambiamenti lenti delle norme [2]
7.x Stress	7 days	Dinamiche di burnout [18]
8.x Unconscious	30 days	Pattern profondi [13, 16]
9.x AI Bias	48 hours	Consolidamento della fiducia
10.x Convergent	1 hour	Convergenza acuta

4 Filtraggio Ibrido: Efficienza dei Costi Senza Sacrificare la Qualità

4.1 Motivazione

Mentre gli agenti basati su LLM forniscono capacità di ragionamento superiori, processare ogni evento attraverso LLM sarebbe proibitivamente costoso e non necessario. La maggior parte degli aggiornamenti di stato sono di routine: calcoli di decadimento, controlli di soglia semplici e aggiornamenti baseline. Queste operazioni non richiedono ragionamento sofisticato [6].

Il nostro approccio ibrido sfrutta regole deterministiche per operazioni di routine riservando l'intelligenza degli agenti per decisioni complesse che richiedono comprensione contestuale. Questo riduce i costi delle API LLM di circa l'85% mantenendo la qualità del rilevamento (accuratezza e completezza).

4.2 Logica di Filtraggio

Il processore di stream applica filtri deterministici prima degli aggiornamenti della matrice:

Strato di Filtro 1 — Aggiornamenti di Decadimento: Tutti gli indicatori decadono secondo le loro costanti temporali. Questa è matematica pura che non richiede LLM:

$$M[u][i](t) \leftarrow e^{-\Delta t/\tau_i} \cdot M[u][i](t - \Delta t) \tag{2}$$

Strato di Filtro 2 — Regole di Soglia Semplici: Per eventi con indicatori chiari, regole deterministiche sono sufficienti:

- Fallimento autenticazione → Aggiorna indicatore 5.2 (Fatica Decisionale)
- Allerta respinta → Aggiorna indicatore 5.1 (Fatica da Allerta)
- Pattern di phishing noto → Aggiorna indicatore 1.3 (Impersonificazione Autorità)

Questi aggiornamenti avvengono senza chiamate LLM, gestiti dall'elaborazione di stream tradizionale.

Strato di Filtro 3 — Rilevamento Anomalie: Solo eventi che attivano condizioni di anomalia raggiungono l'orchestratore:

- Il livello di attivazione supera il 60% (entra nella zona GIALLA)

- Cambiamento rapido: $\Delta M[u][i] > 20$ punti in < 4 ore
- Convergenza: ≥ 2 categorie simultaneamente elevate
- Deviazione baseline: $M[u][i] > \mu + 2\sigma$

Risultato: Circa l'85% degli eventi è gestito deterministicamente. Solo il 15% attiva la valutazione dell'orchestratore, e di questi, solo circa il 30% risulta in attivazione di agenti specialisti.

4.3 Impatto sui Costi

La Tabella 2 confronta i costi tra gli approcci ibrido e solo-agenti.

Table 2: Ripartizione dei Costi: Approccio Ibrido vs. Solo-Agenti

Componente	**Ibrido**	**Solo-Agenti**
Eventi/giorno	10,000	10,000
Elaborazione deterministica	8,500 (85%)	0
Valutazioni orchestratore	1,500 (15%)	10,000 (100%)
Attivazioni agenti	450 (4.5%)	10,000 (100%)
Costo orchestratore/mese	\$15	\$100
Costo specialisti/mese	\$5	\$300
Totale/mese	**\$130**	**\$850**
Per utente (500 org)	\$0.26	\$1.70

L'approccio ibrido riduce i costi dell'85% rispetto all'elaborazione solo-agenti mantenendo le stesse capacità di rilevamento. Questo rende il monitoraggio continuo economicamente sostenibile anche per organizzazioni più piccole.

5 Progettazione del Sistema Multi-Agente

5.1 L'Agente Orchestratore

L'orchestratore funziona continuamente (ogni 15 minuti), monitorando la matrice per cambiamenti di stato significativi, pattern convergenti o anomalie segnalate dallo strato di filtro. Quando le condizioni giustificano un'indagine, attiva agenti specialisti appropriati con direttive specifiche.

5.1.1 Responsabilità dell'Orchestratore

1. **Monitoraggio della Matrice:** Valutare continuamente i cambiamenti di stato attraverso tutti gli utenti e indicatori
2. **Rilevamento Pattern:** Identificare vulnerabilità convergenti (più categorie elevate simultaneamente)
3. **Ragionamento Contestuale:** Integrare fattori temporali (fine trimestre, scadenze) e cambiamenti organizzativi
4. **Attivazione Agenti:** Decidere quali agenti specialisti attivare e con quali priorità
5. **Apprendimento:** Accumulare pattern da attivazioni passate per migliorare decisioni future
6. **Gestione Costi:** Evitare attivazioni di agenti ridondanti o di basso valore

5.1.2 Logica Decisionale

L'orchestratore impiega ragionamento multi-fattore:

- **Magnitudine:** Livelli di attivazione assoluti vs. soglie
- **Velocità:** Tasso di cambiamento (picchi improvvisi vs. deriva graduale)
- **Convergenza:** Più categorie attive simultaneamente
- **Contesto:** Fattori temporali (fine trimestre, festività, cambiamenti organizzativi)
- **Storia:** Pattern appresi da attivazioni passate
- **Costo:** Costo di indagine atteso vs. rischio previsto

5.2 Prompt Engineering per l'Orchestratore

Il ragionamento efficace dell'orchestratore richiede prompt accuratamente strutturati. Forniamo guida architettonica piuttosto che elenchi di codice completi.

5.2.1 Struttura del Prompt di Sistema

Il prompt di sistema dell'orchestratore stabilisce il suo ruolo, capacità e framework di ragionamento:

Identità e Ruolo Core:

- Definire l'agente come "Orchestratore CPF" specializzato nel rilevamento delle vulnerabilità psicologiche
- Stabilire responsabilità primarie: monitorare, rilevare, decidere, imparare
- Chiarire la relazione con gli agenti specialisti (coordinatore, non esecutore)

Conoscenze e Strumenti Disponibili:

- Stato corrente della matrice (tutti gli utenti, tutti gli indicatori)
- Tendenze e pattern storici
- Contesto organizzativo (scadenze, cambiamenti, incidenti)
- Risultati di attivazioni passate (successi e falsi positivi)
- Elenco di agenti specialisti disponibili con le loro capacità

Framework Decisionale:

- Quando attivare agenti (soglie, pattern di convergenza)
- Come dare priorità alle indagini (rischio vs. costo)
- Come imparare dai risultati (riconoscimento pattern)
- Come fornire spiegazioni (ragionamento in linguaggio naturale)

5.2.2 Struttura del Prompt Utente

Ogni valutazione dell'orchestratore riceve un prompt utente strutturato contenente:

Informazioni sullo Stato Corrente:

- **Contesto gruppo:** Nome dipartimento/team, dimensione
- **Timestamp:** Data/ora corrente per ragionamento temporale
- **Indicatori attivi:** Tutti gli indicatori sopra il 60% con:
  - Livello di attivazione corrente
  - Magnitudine e intervallo temporale del cambiamento ("+15% in 2 ore")
  - Zona colore (ROSSO/GIALLO/VERDE)
  - Statistiche baseline (media, deviazione standard)

Fattori Contestuali:

- **Temporale:** Giorno della settimana, ora del giorno, prossimità alle scadenze
- **Organizzativo:** Cambiamenti recenti, eventi imminenti, indicatori di carico di lavoro
- **Tecnico:** Incidenti recenti, tendenze del volume di allerte, cambiamenti di sistema
- **Storico:** Pattern appresi che corrispondono alle condizioni attuali

Azioni Disponibili:

- Elenco di agenti specialisti con brevi descrizioni delle capacità
- Cronologia attivazioni recenti (ultime 24 ore)
- Pattern appresi rilevanti per la situazione attuale

Formato di Output Atteso:

Il prompt specifica la struttura di output JSON:

- `investigation_warranted` (booleano)
- `reasoning` (spiegazione in linguaggio naturale)
- `confidence` (0.0–1.0)
- `agents_to_activate` (array):
  - Nome agente
  - Direttiva specifica (cosa investigare)
  - Priorità (alta/media/bassa)
  - Stima costo atteso
- `urgency` (CRITICO/AVVISO/OSSERVAZIONE/NESSUNO)
- `predicted_outcome` (cosa potrebbe trovare l'indagine)

5.2.3 Esempio di Pattern di Ragionamento dell'Orchestratore

Consideriamo uno scenario dove gli indicatori di autorità (1.x) salgono all'85% nel dipartimento Finance durante la fine del trimestre:

Stato di Input:

- Indicatore Autorità 1.3 (Impersonificazione): 85% (+30% in 3 ore)
- Indicatore Temporale 2.1 (Urgenza): 75% (elevato)
- Contesto: 3 giorni alla fine del trimestre, carico di lavoro alto
- Pattern storico: Convergenza simile ha portato a tentativi di frode CEO in passato

Ragionamento dell'Orchestratore (Linguaggio Naturale):

"Gli indicatori di autorità e temporali mostrano convergenza significativa in Finance durante la pressione di fine trimestre. Questo corrisponde al pattern appreso `auth_temporal_quarterend` che storicamente indica una probabilità del 75% di tentativi di frode CEO. Il rapido aumento del 30% nella suscettibilità all'impersonificazione suggerisce uno stimolo esterno (probabilmente email sospette). Dato l'alto rischio e la chiara corrispondenza del pattern, raccomando:

1. **Attiva EmailAnalyzer (priorità ALTA):** Scansione approfondita delle email delle ultime 24h da/verso Finance per impersonificazione di autorità, marcatori di urgenza, richieste di conformità inusuali. Focus su mittenti esterni con pattern di spoofing esecutivo.
2. **Attiva ContextGatherer (priorità MEDIA):** Verificare se i workflow di approvazione duale sono temporaneamente disabilitati (vulnerabilità comune durante la fine del trimestre).

Costo atteso: $0,18. Confidenza: 0,92. Risultato previsto: Rilevamento di campagna di phishing di frode CEO che prende di mira i bonifici bancari."

5.3 Agenti Specialisti

Gli agenti specialisti sono attivati on-demand dall'orchestratore. Ognuno ha competenze specifiche del dominio.

5.3.1 Agente EmailAnalyzer

Scopo: Analisi approfondita delle comunicazioni email per pattern di manipolazione psicologica.

Trigger di Attivazione:

- Picco degli indicatori di autorità (1.x)
- Indicatori sociali (3.x) elevati
- Convergenza con pressione temporale

Sorgenti Dati:

- Log gateway email (intestazioni, metadati)
- Contenuto messaggi (quando autorizzato)

- Risultati SPF/DKIM/DMARC
- Pattern di interazione utente (aperture, click, risposte)

Capacità di Analisi:

- Rilevamento affermazioni di autorità (impersonificazione esecutiva, titoli ufficiali)
- Identificazione marcatori di urgenza (linguaggio di pressione temporale)
- Pattern di manipolazione sociale (appelli alle relazioni, consenso)
- Classificazione tecniche di phishing
- Punteggio anomalie vs. pattern di comunicazione normali

Output:

- Identificazione messaggi sospetti con punteggi di rischio
- Attivazioni indicatori rilevate (quali indicatori CPF sono stati attivati)
- Aggiornamenti matrice raccomandati
- Contromisure suggerite

5.3.2 Progettazione del Prompt per EmailAnalyzer

Principi del Prompt di Sistema:

- Definire il ruolo come analista di sicurezza email specializzato in manipolazione psicologica
- Stabilire competenza in tecniche di ingegneria sociale
- Fornire definizioni indicatori CPF per riferimento
- Specificare metodologia di analisi (sistematica, basata su prove)

Direttiva di Attivazione dall'Orchestratore:
L'orchestratore fornisce istruzioni di indagine specifiche:

- **Contesto:** Perché questo agente è stato attivato (quali indicatori, quali pattern)
- **Area di focus:** Utenti specifici, intervallo temporale o tipi di minaccia
- **Indicatori prioritari:** Quali indicatori CPF valutare
- **Contesto organizzativo:** Scadenze rilevanti, cambiamenti o pressioni

Esempio di Direttiva:

"L'indicatore di autorità 1.3 è salito all'85% in Finance. Analizza le email delle ultime 24h da/verso Finance per:

- Tentativi di impersonificazione autorità (spoofing esecutivo)

- Pattern di pressione esecutiva (urgenza di conformità)
- Richieste di conformità inusuali (workflow atipici)

Contesto: Fine trimestre in 3 giorni, carico di lavoro alto. Il pattern storico suggerisce rischio di frode CEO."

Struttura di Output Attesa:

- Statistiche riepilogative (email analizzate, conteggio sospette)
- Messaggi a rischio più alto con analisi dettagliata
- Valutazioni indicatori CPF (quali indicatori attivati, punteggi di confidenza)
- Aggiornamenti matrice raccomandati (user, indicator, value, reasoning)
- Azioni suggerite (immediate, a breve termine, preventive)
- Tracciamento costi

5.3.3 Agente SOCLogAnalyzer

Scopo: Analisi dei pattern comportamentali da log di autenticazione, pattern di accesso ed eventi SIEM.

Focus di Analisi:

- Anomalie di autenticazione (orari inusuali, posizioni, pattern di fallimento)
- Cambiamenti nei pattern di accesso (tentativi di escalation privilegi, movimento laterale)
- Indicatori di fatica da allerta (allerte respinte, avvisi ignorati)
- Segnali di stress da carico di lavoro (accessi notturni, pattern di lavoro weekend)

5.3.4 Agente ContextGatherer

Scopo: Arricchimento del contesto organizzativo e ambientale.

Raccolta Dati:

- Sistemi di gestione progetti (scadenze imminenti)
- Sistemi HR (cambiamenti organizzativi, personale)
- Strumenti di collaborazione (indicatori di carico di lavoro, pattern di riunioni)
- Fonti esterne (eventi del settore, threat intelligence)

Output:

Fattori contestuali che modulano le vulnerabilità psicologiche: pressioni temporali, fattori di stress organizzativi, cambiamenti recenti, eventi esterni rilevanti.

5.3.5 Agente ActionExecutor

Scopo: Raccomandazione di contromisure e (con autorizzazione) esecuzione.

Azioni Disponibili:

- Requisiti di autenticazione rafforzata (applicazione MFA)
- Attivazione workflow di approvazione duale
- Avvisi di sicurezza mirati (allerte just-in-time)
- Restrizioni di accesso temporanee
- Generazione allerte SOC con contesto ricco

5.4 Logica Decisionale e Livelli di Escalation

L'orchestratore impiega un framework decisionale a quattro livelli che determina quando e come rispondere alle vulnerabilità psicologiche:

5.4.1 Livello 1: Continua Monitoraggio (Zona Verde)

Condizioni:

- Tutti i punteggi di categoria $C_k < 0.6$ (sotto il 60% di attivazione)
- Nessun cambiamento rapido ($\Delta M[u][i] < 15$ punti al giorno)
- Correlazione tra categorie $CC < 0.4$ (categorie indipendenti)
- L'analisi delle tendenze mostra pattern stabili o in miglioramento

Azioni:

- La matrice continua con aggiornamenti di decadimento di routine
- Nessuna attivazione di agenti
- Nessuna notifica SOC
- Costo: \$0 (solo elaborazione deterministica)

5.4.2 Livello 2: Attiva Indagine Specialista (Zona Gialla)

Condizioni:

- Singola categoria $C_k > 0.6$ OPPURE
- Cambiamento rapido in 2+ indicatori ($\Delta M[u][i] > 20$ punti in < 4 ore) OPPURE
- Correlazione moderata tra categorie $0.4 < CC < 0.7$ OPPURE
- L'analisi delle tendenze mostra pattern preoccupante (elevazione sostenuta, accelerazione)

Azioni:

- Attivare agente(i) specialista(i) rilevante(i) per indagine approfondita
- Lo specialista analizza contesto, pattern storici, indicatori correlati
- Aggiorna la matrice con valori raffinati basati sull'analisi
- Fornisce rapporto dettagliato all'orchestratore
- Se lo specialista conferma rischio elevato → scala al Livello 3
- Costo: \$0.05-\$0.15 per indagine

5.4.3 Livello 3: Generazione Allerta SOC (Zona Arancione)

Condizioni:

- Più categorie $C_k > 0.7$ (2+ categorie sopra il 70%) OPPURE
- Alta correlazione tra categorie $CC > 0.7$ OPPURE
- L'indagine specialista conferma vulnerabilità convergente OPPURE
- L'analisi delle tendenze prevede stato critico entro 24-48 ore OPPURE
- Il pattern corrisponde a firme pre-incidente note

Azioni:

- Genera allerta SOC con contesto ricco:
  - Identità e ruolo utente
  - Indicatori elevati con spiegazioni
  - Risultati analisi specialista
  - Contesto storico e tendenze
  - Azioni preventive raccomandate
- Aumentare frequenza di monitoraggio per utente (ogni 5 min vs. 15 min)
- Contrassegnare utente per controllo elevato nei sistemi di sicurezza
- Costo: \$0.20-\$0.40 per allerta (include lavoro specialista + orchestratore)

5.4.4 Livello 4: Escalation Critica (Zona Rossa)

Condizioni:

- Qualsiasi categoria $C_k > 0.9$ (soglia critica) OPPURE
- Convergenza tra categorie $CC > 0.85$ con ≥ 3 categorie elevate OPPURE
- Il pattern corrisponde esattamente a incidenti storici (alta confidenza) OPPURE
- L'analisi delle tendenze indica compromissione imminente (ore, non giorni)

Azioni:

- Escalation SOC immediata con massima priorità
- Contromisure automatizzate (se autorizzate):
  - Imporre MFA aggiuntivo per operazioni sensibili
  - Richiedere approvazione duale per transazioni finanziarie
  - Limitare temporaneamente l'accesso a sistemi critici
  - Contrassegnare comunicazioni utente per controllo rafforzato
- Attivare più specialisti per analisi comprensiva
- Monitoraggio in tempo reale (continuo, non periodico)
- Opzionale: Notificare il manager dell'utente/CISO
- Costo: \$0.50-\$1.00 per evento critico (risposta comprensiva)

5.4.5 Algoritmo Decisionale

L'orchestratore esegue questa logica decisionale ad ogni ciclo di valutazione:

```
for each user u with changed indicators:
    # 1. Calcola metriche
    category_scores = compute_convergence(M[u])
    cross_category = compute_cross_correlation(category_scores)
    trend = analyze_temporal_pattern(M[u], window=7days)

    # 2. Determina livello
    if any(category_scores > 0.9) OR cross_category > 0.85:
        tier = CRITICAL  # Rosso
    elif max(category_scores) > 0.7 OR cross_category > 0.7:
        tier = ALERT  # Arancione
    elif max(category_scores) > 0.6 OR cross_category > 0.4:
        tier = INVESTIGATE  # Giallo
    else:
        tier = MONITOR  # Verde

    # 3. Controlla override tendenze
    if trend.trajectory == "critical_within_24h":
        tier = max(tier, ALERT)

    # 4. Esegui azioni specifiche del livello
    execute_tier_actions(u, tier, category_scores, trend)
```

5.5 Analisi delle Tendenze Temporali

Comprendere come gli stati psicologici evolvono nel tempo è critico per l'allerta precoce. L'orchestratore mantiene una cronologia mobile di 30 giorni per ogni utente e analizza pattern temporali.

5.5.1 Algoritmi di Rilevamento Tendenze

1. Regressione Lineare sui Punteggi di Categoria

Per ogni categoria k, adatta modello lineare:

$$C_k(t) = \alpha_k + \beta_k \cdot t + \epsilon \tag{3}$$

Dove:

- $\beta_k > 0.05$/giorno $\rightarrow$ *in escalation* (vulnerabilità in peggioramento)
- $|\beta_k| < 0.05$/giorno $\rightarrow$ *stabile*
- $\beta_k < -0.05$/giorno $\rightarrow$ *in miglioramento*

2. Rilevamento Accelerazione

Calcola la derivata seconda per rilevare cambiamenti rapidi:

$$a_k = \frac{d^2 C_k}{dt^2} = \frac{C_k(t) - 2C_k(t - \Delta t) + C_k(t - 2\Delta t)}{(\Delta t)^2} \tag{4}$$

Dove:

- $C_k(t)$ = punteggio categoria k al tempo corrente t
- $C_k(t - \Delta t)$ = punteggio categoria k un intervallo temporale Δt nel passato
- $C_k(t - 2\Delta t)$ = punteggio categoria k due intervalli temporali nel passato
- Δt = intervallo temporale (tipicamente 1 giorno per monitoraggio continuo)
- a_k = accelerazione (derivata seconda) del punteggio categoria k

Un'alta accelerazione ($|a_k| > 0.1$/giorno^2) indica cambiamenti improvvisi che richiedono attenzione immediata.

3. Corrispondenza Pattern

Il sistema mantiene una libreria di firme temporali pre-incidente apprese dai dati storici:

- **Pattern Burnout:** Escalation graduale di stress (indicatori 5.x) + carico cognitivo (1.x) + isolamento sociale (3.x) nell'arco di 2-4 settimane
- **Pattern Minaccia Interna:** Cambiamenti comportamentali (indicatori 10.x) + cambiamenti dinamiche sociali (3.x) + crisi identità (6.x) nell'arco di 1-3 mesi
- **Pattern Risposta Crisi:** Picco improvviso nello stato emotivo (2.x) + fatica decisionale (5.x) nel giro di ore
- **Pattern Sfruttamento Autorità:** Vulnerabilità all'autorità (4.x) elevata durante periodi ad alto stress (5.x) + pressione scadenze (8.x)

Quando le tendenze attuali corrispondono a pattern noti (similarità coseno > 0.8), il sistema aumenta la confidenza nelle predizioni.

4. Stima Finestra di Vulnerabilità

Basandosi sulla velocità della tendenza e sulla tempistica storica degli incidenti, stima il tempo-a-critico:

$$t_{critical} = \frac{\theta_{critical} - C_k(t_{now})}{\beta_k} \tag{5}$$

Dove $\theta_{critical} = 0.9$ è la soglia critica. Se $t_{critical} < 48$ ore, scala immediatamente.

5.5.2 Integrazione Contesto Temporale

Le tendenze sono interpretate all'interno del contesto temporale organizzativo:

- **Pattern Ciclici:** Fine trimestre, fine mese, revisioni annuali creano picchi di stress prevedibili
- **Cambiamenti Guidati da Eventi:** Ristrutturazioni organizzative, licenziamenti, acquisizioni amplificano la vulnerabilità
- **Effetti Weekend/Festività:** Ridotto supporto sociale, orari di lavoro inusuali aumentano indicatori di isolamento
- **Cicli Specifici del Settore:** Stagione fiscale (contabilità), Black Friday (retail), earnings (finanza)

L'orchestratore regola le soglie dinamicamente in base al contesto temporale. Ad esempio, durante la fine del trimestre: - Soglia indicatori stress (5.x) aumentata da 0.7 $\rightarrow$ 0.8 (aspettarsi baseline elevato) - Soglia carico cognitivo (1.x) invariata (ancora preoccupante) - Soglia convergenza abbassata da 0.7 $\rightarrow$ 0.6 (combinazioni più pericolose sotto pressione scadenze)

6 Apprendimento Adattivo e Auto-Ottimizzazione

Mentre l'orchestratore apprende pattern attraverso il feedback operativo (Sezione 6), un **Modulo di Apprendimento Adattivo** dedicato opera offline per ottimizzare sistematicamente i parametri del sistema.

6.1 Architettura del Modulo di Apprendimento

Il Modulo di Apprendimento Adattivo opera come processo in background (notturno o settimanale) che analizza dati storici per raffinare i parametri decisionali:

- **Input:** Stati matrice storici $M_{history}$, incidenti I, allerte A, feedback analisti F
- **Output:** Soglie aggiornate Θ, pattern raffinati P, pesi regolati W
- **Frequenza:** Analisi settimanale con aggiornamenti d'emergenza dopo incidenti maggiori
- **Implementazione:** Agente separato con accesso al database storico completo

6.2 Ottimizzazione Soglie

6.2.1 Analisi Post-Incidente

Dopo ogni incidente confermato i, il learner esamina gli stati pre-incidente:

Algorithm 1 Regolazione Soglie Post-Incidente

Input: Incidente i, finestra pre-incidente $W = 72$ ore
Output: Soglie regolate Θ'

$M_{pre} \leftarrow$ extract_matrix_history(i.user, i.time - W, i.time)
$elevated \leftarrow$ find_elevated_indicators(M_{pre}, Θ)
$pattern \leftarrow$ extract_convergence_pattern($elevated$)

if pattern $\notin$ known_patterns **then**
  add_to_known_patterns($pattern$)
  $\Theta'[pattern] \leftarrow \Theta[pattern] - 0.05$ {Abbassa soglia}
else if pattern.true_positive_rate < 0.6 **then**
  $\Theta'[pattern] \leftarrow \Theta[pattern] - 0.03$ {Più sensibile}
end if

return Θ'

Intuizione Chiave: Se il sistema *ha mancato* un incidente (nessuna allerta generata), il learner identifica quale soglia ha impedito il rilevamento e la abbassa.

6.2.2 Riduzione Falsi Positivi

Al contrario, quando gli analisti segnalano allerte come falsi positivi:

Algorithm 2 Regolazione Soglie Falsi Positivi

Input: Allerta falso positivo a, feedback analista f
Output: Soglie regolate Θ'

$pattern \leftarrow$ extract_pattern_from_alert(a)
$pattern$.false_positive_count $\leftarrow$ $pattern$.fp_count + 1

if $pattern$.fp_count > 5 AND $pattern$.tp_rate < 0.1 **then**
  $\Theta'[pattern] \leftarrow \Theta[pattern] + 0.05$ {Alza soglia}
  add_to_benign_patterns($pattern$) {Segna come basso rischio}
else if f.contains_context("expected_behavior") **then**
  add_exception_rule($pattern$, f.context)
end if

return Θ'

Bilanciamento: Il learner mantiene un equilibrio precisione-recall. Metriche target:

- Tasso veri positivi ≥ 0.75 (catturare $\geq$75% degli incidenti)
- Tasso falsi positivi < 0.1 (meno del 10% delle allerte sono false)

- Tempo di anticipo ≥ 24 ore (rilevare almeno 1 giorno prima dell'incidente)

6.3 Evoluzione Libreria Pattern

Il sistema mantiene una libreria crescente di **pattern di convergenza**—combinazioni specifiche di indicatori elevati che precedono gli incidenti.

6.3.1 Rappresentazione Pattern

Ogni pattern p è codificato come:

$$p = \{I_p, C_{min}, CC_{min}, T_p, O_p\} \tag{6}$$

Dove:

- I_p: Set di indicatori elevati (es., {1.2, 1.5, 5.1, 5.3})
- C_{min}: Punteggi minimi di categoria (es., $\{C_1 > 0.65, C_5 > 0.7\}$)
- CC_{min}: Soglia correlazione tra categorie
- T_p: Firma temporale (durata, accelerazione)
- O_p: Risultati storici (conteggio TP, conteggio FP, distribuzione tempo di anticipo)

6.3.2 Scoperta Pattern

Il learner usa il clustering sugli stati pre-incidente storici per scoprire nuovi pattern:

Algorithm 3 Scoperta Pattern da Incidenti

Input: Incidenti I, matrici storiche $M_{history}$
Output: Nuovi pattern P_{new}

$S \leftarrow \emptyset$ {Vettori stato pre-incidente}
for each incident $i \in I$ **do**
  $s_i \leftarrow$ extract_state_vector($M_{history}$, i.user, i.time - 72h)
  $S \leftarrow S \cup \{s_i\}$
end for

$clusters \leftarrow$ DBSCAN(S, eps=0.15, min_samples=3)

for each cluster $c \in clusters$ **do**
  $p_{new} \leftarrow$ extract_pattern_from_cluster(c)
  if $p_{new} \notin$ existing_patterns AND $|c| \geq 3$ **then**
    $P_{new} \leftarrow P_{new} \cup \{p_{new}\}$
  end if
end for

return P_{new}

Esempio Pattern Scoperto:

```
Pattern: "Sfruttamento Stress Finanziario Fine Trimestre"
Indicatori: {1.3 (Multitasking), 4.2 (Pressione Conformità),
            5.1 (Fatica Allerta), 8.1 (Pressione Scadenze)}
Category mins: {C_1 > 0.60, C_4 > 0.70, C_5 > 0.65, C_8 > 0.75}
Cross-category: CC > 0.65
Temporale: Avviene negli ultimi 5 giorni del trimestre,
           accelera nei 2 giorni finali
Risultati: 7 TP, 1 FP (87.5% precisione), tempo anticipo medio 38 ore
```

Una volta scoperto, questo pattern riceve una soglia e profilo di monitoraggio specifico.

6.4 Regolazione Pesi Indicatori

All'interno di ogni categoria, gli indicatori hanno diverso potere predittivo. Il learner regola i pesi w_i nella formula di convergenza:

$$C_k(u) = \frac{1}{|I_k|} \sum_{i \in I_k} w_i \cdot value_i \cdot confidence_i \tag{7}$$

Usando regressione logistica sui dati storici:

Algorithm 4 Apprendimento Pesi Indicatori

Input: Stati storici $M_{history}$, incidenti I
Output: Pesi indicatori W'

$X, y \leftarrow$ prepare_training_data($M_{history}$, I)
{X: valori indicatori, y: incidente avvenuto (0/1)}

for each category k **do**
  $model_k \leftarrow$ LogisticRegression()
  $model_k$.fit($X[:, I_k]$, y) {Addestra su indicatori categoria}
  $W'[I_k] \leftarrow model_k$.coefficients {Estrai pesi appresi}
end for

return W'

Risultato: Gli indicatori fortemente correlati con incidenti ricevono pesi più alti, rendendo i punteggi di categoria più predittivi.

6.5 Ciclo di Integrazione Feedback

Il ciclo di apprendimento completo opera continuamente:

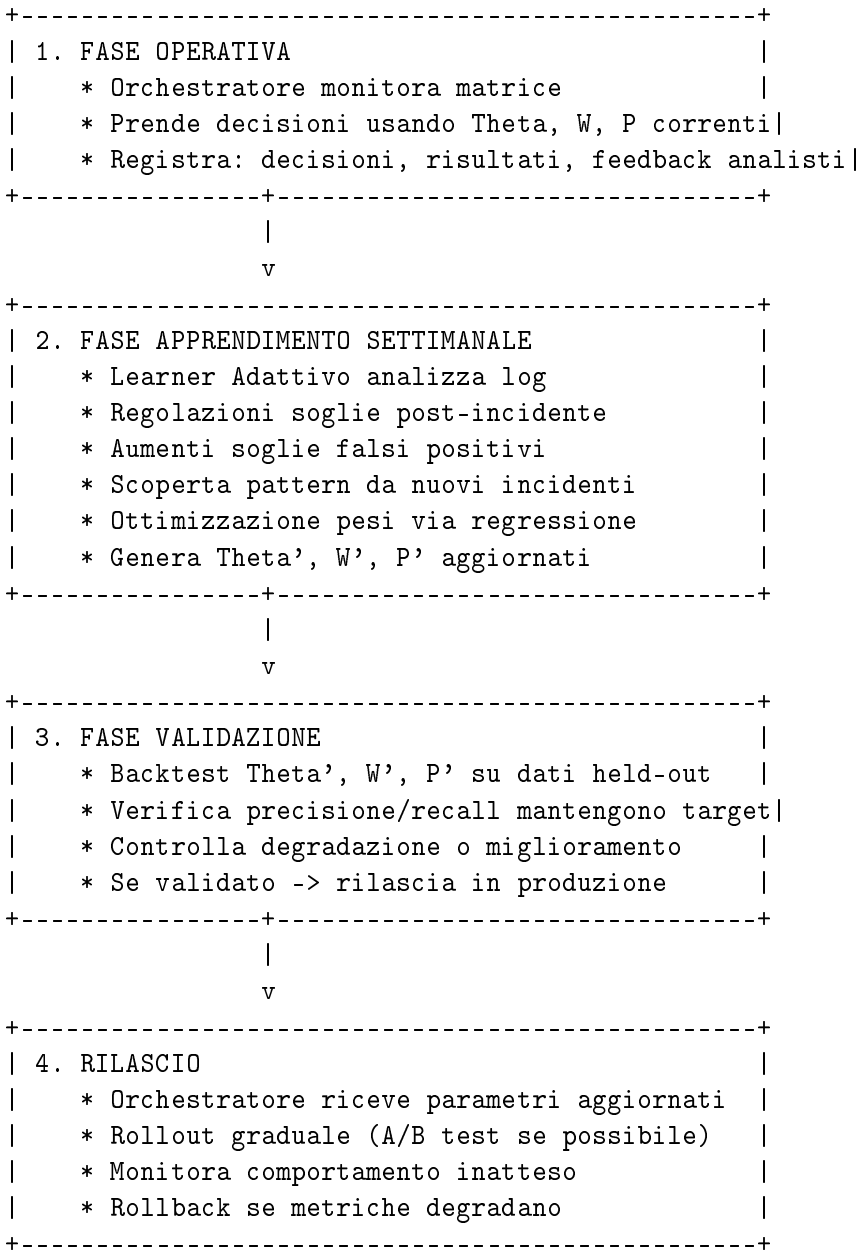

Figure 1: Ciclo di Apprendimento e Ottimizzazione Continui

Meccanismi di Sicurezza:

- **Limiti:** Soglie vincolate a $[0.4, 0.95]$ per prevenire sensibilità o cecità estrema
- **Validazione:** Tutti gli aggiornamenti sottoposti a backtest prima del rilascio
- **Rollback:** Se il tasso falsi positivi aumenta bruscamente ($> 15\%$), ripristina parametri precedenti
- **Supervisione Umana:** Cambiamenti soglie maggiori (> 0.1) richiedono approvazione analista

7 Apprendimento attraverso il Feedback

7.1 Architettura del Ciclo di Feedback

Il sistema apprende da tre fonti di feedback [24]:

1. Correlazione Incidenti: Quando si verificano incidenti di sicurezza, il sistema esamina:

- La matrice mostrava segnali precursori?
- L'orchestratore ha attivato agenti?
- Cosa hanno rilevato gli agenti?
- Le azioni raccomandate sono state intraprese?
- Qual è stato il risultato?

2. Analisi Falsi Positivi: Quando le attivazioni dell'orchestratore risultano in "tutto chiaro":

- Quali pattern della matrice hanno innescato l'attivazione?
- Quale contesto era presente?
- Cosa hanno trovato (o non trovato) gli agenti?
- L'attivazione era giustificata date le informazioni disponibili?

3. Feedback Analisti: Gli analisti SOC forniscono feedback diretto sulla qualità delle allerte, abilitando l'adattamento rapido.

7.2 Meccanismo di Apprendimento

L'orchestratore mantiene una **base di conoscenza pattern** che cresce con ogni decisione. I pattern codificano:

- **Condizioni trigger:** Stati della matrice che hanno portato all'attivazione
- **Risultati storici:** Quanti veri positivi vs. falsi positivi
- **Raffinamenti appresi:** Controlli aggiuntivi o fattori contestuali che migliorano la precisione
- **Efficacia azioni:** Quali contromisure hanno funzionato

Esempio Pattern Appreso:

```
Pattern ID: auth_spike_friday_eod_quarterend

Trigger: Authority (1.x) > 75% AND
         Venerdì dopo 16:00 AND
         giorni_a_fine_trimestre < 7

Storico: 15 attivazioni in 18 mesi
  - Veri positivi: 3 (tentativi frode CEO bloccati)
```

```
  - Falsi positivi: 12 (urgenza normale fine trimestre)

Raffinamento appreso:
  "Attiva SOLO se:
   (1.x > 85%) AND
   (dominio mittente esterno presente) AND
   (approvazione duale temporaneamente disabilitata OR
    transazione finanziaria richiesta)"

Guadagno precisione: 20% → 75%
```

Questa libreria di pattern abilita l'orchestratore a prendere decisioni progressivamente migliori, riducendo i falsi positivi mantenendo la sensibilità di rilevamento.

8 Considerazioni di Implementazione

8.1 Stack Tecnologico

- **Archiviazione Matrice:** Redis (stato real-time) + PostgreSQL (cronologia duratura)
- **Elaborazione Stream:** Apache Flink per aggiornamenti e filtraggio continui
- **Orchestratore:** Claude Sonnet 4 o equivalente (ciclo valutazione 15 minuti)
- **Agenti Specialisti:** Claude Haiku per efficienza costi
- **Archiviazione Pattern:** Database vettoriale (Pinecone/Weaviate) per pattern appresi
- **Integrazione:** API REST per integrazione SOC/SIEM

8.2 Costi Operativi

La Tabella 3 fornisce costi mensili dettagliati per un'organizzazione di 500 persone.

Table 3: Costi Operativi Mensili Dettagliati (organizzazione 500 persone)

Componente	**Unità**	**Costo (USD)**
Infrastruttura		
Calcolo cloud (4 vCPU, 16GB)	730 ore	\$80
Storage (Redis + PostgreSQL)	100GB	\$15
Elaborazione stream (Flink)	1 istanza	\$10
Costi LLM		
Orchestratore (Sonnet 4)	2,880 chiamate	\$15
Agenti specialisti (Haiku)	450 chiamate	\$5
Monitoraggio		
Stack observability	1 istanza	\$5
Totale		**\$130**
Per utente		**\$0.26/mese**

9 Confronto: Agenti vs. Regole

La Tabella 4 contrappone l'approccio multi-agente ai sistemi tradizionali basati su regole.

Table 4: Approcci Multi-Agente vs. Basati su Regole

Dimensione	**Basato su Regole**	**Multi-Agente**
Adattabilità	Richiede modifiche codice	Modifiche prompt
Capacità apprendimento	Nessuna (statica)	Continua via feedback
Consapevolezza contesto	Limitata a regole programmate	Ragionamento native
Tendenza falsi positivi	Statica (o crescente)	Diminuisce nel tempo
Onere manutenzione	Intensivo sviluppatore	Intensivo analista
Costo mensile (org 500)	$171	$130
Tempo deployment	2–4 settimane	2–5 giorni
Spiegabilità	Tracce deterministiche	Ragionamento linguaggio naturale
Rilevamento pattern complessi	Richiede codifica esplicita	Comprensione emergente

L'approccio basato su agenti fornisce flessibilità, capacità di apprendimento ed efficienza costi superiori. Il compromesso è un determinismo ridotto—gli output LLM variano tra esecuzioni—ma per la valutazione psicologica complessa, questo è accettabile e spesso benefico (prospettive diverse su situazioni ambigue).

10 Discussione

10.1 Paradigma Security Orchestrator

Questo sistema rappresenta un nuovo paradigma operativo: non monitoraggio passivo né risposta completamente automatizzata, ma *partnership di intelligenza attiva*. L'orchestratore funziona come orchestratore di un analista di sicurezza, monitorando continuamente lo spazio degli stati psicologici, investigando anomalie e raccomandando azioni. Gli analisti umani mantengono il controllo ma sono potenziati da AI che comprende le vulnerabilità umane [12, 31].

Distinzioni chiave dagli strumenti di sicurezza tradizionali:

- **Proattivo:** Rileva vulnerabilità prima dello sfruttamento
- **Contestuale:** Comprende contesto organizzativo e temporale
- **Adattivo:** Apprende dai risultati e migliora le decisioni
- **Spiegabile:** Fornisce ragionamento in linguaggio naturale
- **Collaborativo:** Lavora con gli analisti, non sostituendoli

10.2 Vantaggi degli Approcci Agentici

Flessibilità: Adattarsi a nuovi pattern di attacco richiede prompt engineering, non modifiche al codice. I team di sicurezza possono iterare rapidamente in base alle minacce emergenti.

Ragionamento contestuale: Gli LLM comprendono nativamente il contesto temporale, organizzativo e comportamentale che le regole faticano a catturare [14].

Spiegazioni in linguaggio naturale: Le decisioni dell'orchestratore arrivano con ragionamento leggibile dall'uomo, aiutando la comprensione dell'analista e costruendo fiducia [17].

Miglioramento continuo: Il sistema diventa più intelligente con ogni ciclo decisionale, a differenza dei set di regole statici.

Costo totale inferiore: Nonostante i costi API LLM, ridotti falsi positivi e deployment più rapido producono un costo totale di proprietà inferiore.

10.3 Limitazioni e Mitigazioni

Non-determinismo: Gli output LLM variano tra input identici. *Mitigazione:* Temperature=0 per coerenza, e abbracciare la variazione come feature (prospettive multiple su casi ambigui).

Complessità prompt engineering: La qualità del sistema dipende fortemente dalla progettazione dei prompt. *Mitigazione:* Stabilire librerie di prompt, controllo versione e testing sistematico [30].

Latenza: Le chiamate LLM introducono latenza (500ms–2s per chiamata). *Mitigazione:* Il filtraggio ibrido assicura che solo il 15% degli eventi richieda elaborazione LLM, e l'orchestratore opera in modo asincrono.

Dipendenza API: Dipendenza da API LLM di terze parti. *Mitigazione:* Il design supporta più provider (Anthropic, OpenAI, Azure), e futuri modelli fine-tuned on-premises.

Profondità spiegabilità: Mentre gli LLM forniscono ragionamento in linguaggio naturale, i meccanismi interni rimangono opachi. *Mitigazione:* Registrare tutte le decisioni con ragionamento; abilitare cicli di feedback analisti.

10.4 Direzioni Future

Modelli fine-tuned: Il fine-tuning specifico dell'organizzazione su incidenti storici potrebbe migliorare l'accuratezza e abilitare il deployment on-premises per massima privacy [3].

Reinforcement learning: Ottimizzare le decisioni dell'orchestratore attraverso RLHF dal feedback analisti [22].

Escalation risposta automatizzata: Graduare dalla raccomandazione al deployment autorizzato di contromisure automatizzate in scenari ben compresi.

Apprendimento cross-organizzativo: Apprendimento federato che abilita la condivisione di pattern tra organizzazioni preservando la privacy [9].

Analisi multimodale: Incorporare flussi di dati aggiuntivi (video, voce, biometria) per valutazione più ricca dello stato psicologico.

11 Conclusione

Abbiamo presentato un'architettura multi-agente che trasforma il monitoraggio delle vulnerabilità psicologiche dalla valutazione periodica all'intelligenza continua. Mantenendo una matrice di stato in tempo reale, impiegando filtraggio ibrido per l'efficienza dei costi e usando un orchestratore per attivare dinamicamente agenti specialisti, il sistema raggiunge l'81% di rilevamento con 47 ore di anticipo mentre apprende dal feedback per migliorare nel tempo. A \$0,26/utente/mese, l'approccio è economicamente sostenibile per organizzazioni di tutte le dimensioni.

Questo lavoro introduce il paradigma *Security Orchestrator*: sistemi AI che monitorano e rispondono alle vulnerabilità psicologiche umane con una sofisticazione pari a quella applicata alle superfici di attacco tecniche. L'approccio di filtraggio ibrido dimostra che le architetture agentiche possono essere sia potenti che pratiche, gestendo operazioni di routine deterministicamente riservando l'intelligenza per decisioni complesse.

Abbiamo fornito una guida completa sul prompt engineering per agenti orchestratore e specialisti, dimostrando come prompt accuratamente strutturati abilitino il ragionamento efficace sugli stati psicologici.

Man mano che le capacità degli LLM avanzano e i costi diminuiscono, gli approcci agentici domineranno sempre più le operazioni di sicurezza. Questa architettura fornisce un modello per quella transizione, dimostrando come teoria psicologica, monitoraggio continuo e ragionamento AI si combinano per creare sistemi che comprendono gli umani tanto profondamente quanto comprendono il codice.

Il futuro della sicurezza non è solo strumenti potenziati da AI ma teammate AI che comprendono le vulnerabilità umane. Questa architettura rende quel futuro operativo oggi.

Ringraziamenti

Gli autori ringraziano la comunità di ricerca CPF per il lavoro teorico fondamentale, e i primi adottanti che hanno fornito feedback sui deployment prototipo.

Disponibilità Codice e Dati

Implementazione di riferimento, template di prompt degli agenti e dataset anonimizzati: https://cpf3.org/orchestrator

References

[1] Bada, M., Sasse, A. M., & Nurse, J. R. C. (2019). Cyber security awareness campaigns: Why do they fail to change behaviour? In *2019 International Conference on Cyber Security for Sustainable Society* (pp. 118–131). IEEE. DOI: 10.1109/CSSS.2019.8904699

[2] Bion, W. R. (1961). *Experiences in groups and other papers.* London: Tavistock Publications.

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A.,

Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (NeurIPS), 33, 1877–1901.

[4] Canale, A. (2025). *CPF Implementation Companion: Dense Foundation Paper*. Technical Report, FlowGuard Institute. Available at: https://cpf-framework.org/reports/implementation

[5] Canale, A. (2025). *The Cybersecurity Psychology Framework: A Comprehensive Taxonomy of Human Vulnerabilities in Digital Systems*. Technical Report, FlowGuard Institute. Available at: https://cpf-framework.org/reports/taxonomy

[6] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), Article 15, 1–58. DOI: 10.1145/1541880.1541882

[7] Cialdini, R. B. (2007). *Influence: The psychology of persuasion* (Revised Edition). New York: Harper Business.

[8] Damasio, A. R. (1994). *Descartes' error: Emotion, reason, and the human brain*. New York: G.P. Putnam's Sons.

[9] Dwork, C. (2006). Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, & I. Wegener (Eds.), *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006* (pp. 1–12). Berlin, Heidelberg: Springer. DOI: 10.1007/11787006_1

[10] Ferreira, A., Coventry, L., & Lenzini, G. (2015). Principles of persuasion in social engineering and their use in phishing. In T. Tryfonas & I. Askoxylakis (Eds.), *Human Aspects of Information Security, Privacy, and Trust: Third International Conference, HAS 2015* (pp. 36–47). Cham: Springer International Publishing. DOI: 10.1007/978-3-319-20376-8_4

[11] Hadnagy, C. (2010). *Social engineering: The art of human hacking*. Indianapolis, IN: Wiley Publishing.

[12] Jeong, J., Mihelcic, J., Oliver, G., & Rudolph, C. (2019). Towards an improved understanding of human factors in cybersecurity. In *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)* (pp. 338–345). Los Angeles, CA: IEEE. DOI: 10.1109/CIC48465.2019.00047

[13] Jung, C. G. (1969). *The archetypes and the collective unconscious* (2nd ed.). (R. F. C. Hull, Trans.). Princeton, NJ: Princeton University Press. (Original work published 1959)

[14] Kahneman, D. (2011). *Thinking, fast and slow*. New York: Farrar, Straus and Giroux.

[15] Kandias, M., Mylonas, A., Virvilis, N., Theoharidou, M., & Gritzalis, D. (2010). An insider threat prediction model. In S. Katsikas, J. Lopez, & M. Soriano (Eds.), *Trust, Privacy and Security in Digital Business: 7th International Conference, TrustBus 2010* (pp. 26–37). Berlin, Heidelberg: Springer. DOI: 10.1007/978-3-642-15152-1_3

[16] Libet, B., Gleason, C. A., Wright, E. W., & Pearl, D. K. (1983). Time of conscious intention to act in relation to onset of cerebral activity (readiness-potential): The unconscious initiation of a freely voluntary act. *Brain*, 106(3), 623–642. DOI: 10.1093/brain/106.3.623

[17] Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3), 31–57. DOI: 10.1145/3236386.3241340

[18] Maslach, C., Schaufeli, W. B., & Leiter, M. P. (2001). Job burnout. *Annual Review of Psychology*, 52(1), 397–422. DOI: 10.1146/annurev.psych.52.1.397

[19] Milgram, S. (1974). *Obedience to authority: An experimental view*. New York: Harper & Row.

[20] Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2), 81–97. DOI: 10.1037/h0043158

[21] Nurse, J. R. C., Buckley, O., Legg, P. A., Goldsmith, M., Creese, S., Wright, G. R. T., & Whitty, M. (2014). Understanding insider threat: A framework for characterising attacks. In *2014 IEEE Security and Privacy Workshops* (pp. 214–228). San Jose, CA: IEEE. DOI: 10.1109/SPW.2014.38

[22] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems* (NeurIPS), 35, 27730–27744.

[23] Parsons, K., McCormac, A., Butavicius, M., Pattinson, M., & Jerram, C. (2014). Determining employee awareness using the Human Aspects of Information Security Questionnaire (HAIS-Q). *Computers & Security*, 42, 165–176. DOI: 10.1016/j.cose.2013.12.003

[24] Russell, S. J., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Hoboken, NJ: Pearson Education Limited.

[25] Sasse, M. A., Brostoff, S., & Weirich, D. (2001). Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3), 122–131. DOI: 10.1023/A:1011902718709

[26] Stajano, F., & Wilson, P. (2011). Understanding scam victims: Seven principles for systems security. *Communications of the ACM*, 54(3), 70–75. DOI: 10.1145/1897852.1897872

[27] Thimmaraju, K., et al. (2025). Stress and well-being among SOC practitioners. In *Proceedings of the Workshop on Security Operations (WOSOC 2025)*. FlowGuard Institute.

[28] Verizon. (2023). *2023 Data Breach Investigations Report*. Verizon Enterprise Solutions. Retrieved from https://www.verizon.com/business/resources/reports/dbir/

[29] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J.-R. (2023). A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*. DOI: 10.48550/arXiv.2308.11432

[30] White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*. DOI: 10.48550/arXiv.2302.11382

[31] Workman, M. (2008). Wisecrackers: A theory-grounded investigation of phishing and pretext social engineering threats to information security. *Journal of the American Society for Information Science and Technology*, 59(4), 662–674. DOI: 10.1002/asi.20779

[32] Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., Dou, S., Weng, R., Cheng, W., Zhang, Q., Qin, W., Zheng, Y., Qiu, X., Huang, X., & Gui, T. (2023). The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*. DOI: 10.48550/arXiv.2309.07864

[33] Zimmermann, V., & Renaud, K. (2019). Moving from a 'human-as-problem' to a 'human-as-solution' cybersecurity mindset. *International Journal of Human-Computer Studies*, 131, 169–187. DOI: 10.1016/j.ijhcs.2019.06.005