MODULO 06

LLM e agenti nel SOC

Copiloti investigativi, generazione di query e regole di detection, sintesi degli incidenti e prospettiva dell'agentic SOC, con la verifica umana come requisito costante.

Security Copilot | KQL / SPL / Sigma | Ollama

SCHEMA · SPECIALIZZAZIONI

Il modulo comune a due specializzazioni

SCHEDA DEL MODULO

Abstract e obiettivi formativi

Abstract

Il modulo studia l'integrazione dei modelli linguistici nel Security Operations Center: architetture dei copiloti commerciali e loro grounding sui dati di tenant, prompt engineering investigativo, generazione e validazione di regole, sintesi degli incidenti, modelli open weight in locale e prospettiva dell'agentic SOC.

Obiettivi formativi (Bloom)

- **Applicare**: usare un copilota per accelerare un'indagine mantenendo la verifica umana
- **Creare**: generare e validare query KQL/SPL e regole Sigma prodotte da un LLM
- **Valutare**: giudicare benefici e rischi dell'autonomia agentica nel triage

Traccia core: 01 02 07

Detection & Response: 03 04 05 **06** o Digital Forensics: 08 09 10 11 o Sicurezza dell'AI: **06** 12 13 14

Traccia core: 15 16

Letture di raccordo per Sicurezza dell'AI: § 3.6, 3.9, 5.1

SCHEMA · TEORIA

Impieghi dell'LLM ad autonomia crescente

Capacità e limiti dei copiloti investigativi

Capacità mature

- Sintesi di un incidente da alert e log eterogenei
- Spiegazione di script e comandi sospetti
- Traduzione di richieste in linguaggio naturale in query (KQL, SPL)
- Grounding sui dati del tenant: risposte ancorate alla telemetria dell'organizzazione

Limiti strutturali

- Allucinazioni plausibili: entità o eventi mai esistiti nei log
- Groundedness da misurare caso per caso
- Dati riservati verso cloud di terze parti: questione contrattuale e GDPR
- Alternativa: modelli open weight in locale (Ollama, vLLM) per casi d'uso sensibili

TEORIA · DETECTION ENGINEERING ASSISTITO

Regole di detection generate da un LLM

01 · GENERA

L'LLM produce la regola Sigma dalla descrizione della tecnica

→

02 · CONVERTI

sigma-cli traduce verso il SIEM di destinazione

→

03 · VALIDA

Test su log noti: attacco simulato e traffico benigno

→

04 · RILASCIA

Versionamento, revisione tra pari, monitoraggio dei FP

Errori ricorrenti delle regole generate

Campi inesistenti nello schema dei log, condizioni troppo larghe, che producono molti falsi positivi, e logiche troppo strette, adattate al solo esempio. In altri casi la sintassi è valida ma la semantica è errata. La validazione del passo 03 è quindi obbligatoria.

TEORIA · AGENTIC SOC

Agenti nel triage di livello 1

Potenzialità

- L'agente pianifica l'indagine in più passi: interroga il SIEM, arricchisce, conclude
- Compiti adatti: deduplicazione, chiusura dei falsi positivi ricorrenti, raccolta di contesto
- Gli analisti si spostano su hunting e casi complessi

Condizioni d'uso

- Strumenti in sola lettura per impostazione predefinita; azioni di scrittura subordinate ad approvazione
- Privilegio minimo per ogni strumento dell'agente
- Log completo di piani, chiamate e motivazioni, ripreso come AI forensics (Modulo 14)
- Misura della groundedness e controllo umano a campione degli esiti

SCHEMA · APPROFONDIMENTI

Uso controllato dei modelli linguistici

APPROFONDIMENTO 1

Struttura del prompt operativo

Un copilota SOC si interroga con un prompt strutturato in tre parti. Il **grounding** fornisce le evidenze su cui il modello deve basarsi: log, alert, estratti di regole, inseriti nel prompt o recuperati con tecniche di retrieval (Modulo 14). Il contesto di ruolo descrive il compito richiesto, per esempio triage, formulazione di ipotesi o query di verifica. Il **formato vincolato** stabilisce i campi dell'output, la citazione delle righe di log usate e il livello di confidenza. Se manca il grounding, il modello completa la risposta con conoscenze generali apprese in addestramento, e un output privo di formato vincolato non si può verificare né passare ad altri strumenti.

Esempio

«Sei un analista di secondo livello. Dati questi 40 eventi Sysmon, indica il processo sospetto, la tecnica ATT&CK candidata e tre query KQL di verifica. Cita gli identificativi degli eventi usati. Se le evidenze non bastano, dichiaralo.» Con l'ultima istruzione il modello può astenersi quando mancano elementi, pratica raccomandata per limitare le risposte inventate.

Metodo

Il prompt fa parte della configurazione: va versionato, provato su casi noti e sottoposto a revisione come il codice dei playbook. Prompt modificati liberamente dai singoli analisti producono triage non riproducibili, con conseguenze anche sul piano probatorio.

APPROFONDIMENTO 2

Allucinazioni dei modelli e loro misura

Un modello linguistico genera testo scegliendo, token dopo token, continuazioni probabili dato il contesto, senza verificarne la correttezza. Quando il contesto non contiene la risposta, il modello la completa con schemi appresi in addestramento: nel SOC ne derivano indicatori mai osservati, identificativi CVE inesistenti ma plausibili, comandi di remediation errati con sintassi corretta. La contromisura consiste nel misurare la **groundedness**: ogni affermazione dell'output deve essere riconducibile a un'evidenza presente nel contesto, e ciò che non è riconducibile si tratta come ipotesi da verificare.

Scenario didattico

Chiesto di riassumere un incidente, il copilota aggiunge che l'host ha contattato un indirizzo IP associato alla rete Tor. L'indirizzo non compare in alcun log: riproduce uno schema frequente nei report usati per l'addestramento. Se l'indicatore entra nel report finale, nelle indagini successive viene trattato come un fatto accertato.

Metodo

Ogni indicatore, hash o CVE presente nell'output viene cercato nei dati originali prima dell'uso. I riassunti destinati a report formali restano bozze finché non passano dalla revisione umana con le fonti a fronte.

APPROFONDIMENTO 3

Modello locale o servizio cloud

Ogni prompt inviato a un servizio esterno è una **comunicazione di dati** al fornitore, e un trasferimento ai sensi del capo V GDPR se il fornitore li tratta fuori dallo Spazio economico europeo: log con indirizzi interni, nomi utente, estratti di posta elettronica, talvolta frammenti di malware. La scelta tra un modello in cloud, più capace e senza oneri di gestione, e un modello eseguito in locale con Ollama o vLLM, che lascia i dati sotto il controllo dell'organizzazione con capacità minori e costi infrastrutturali, è una decisione di governance. Richiede la classificazione dei dati che possono uscire, contratti e accordi sul trattamento con il fornitore e informazioni sulla conservazione dei prompt presso il provider. Un'architettura ibrida assegna i dati d'indagine a un modello locale e riserva il cloud a compiti su dati pubblici o anonimizzati.

Scenario didattico

Un analista incolla in un chatbot pubblico il testo di una e-mail di spear phishing per farsela spiegare. Il testo contiene nome, ruolo e numero diretto del dirigente bersaglio, che finiscono nei sistemi di un fornitore situato fuori dallo Spazio economico europeo. Si configura un nuovo incidente, con possibili obblighi ai sensi dell'art. 33 GDPR.

Metodo

Una policy scritta stabilisce quali dati possono uscire, e il registro dei prompt fa parte dell'audit trail del SOC. La pseudonimizzazione automatica dei prompt sostituisce indirizzi e identità, mentre i dati d'indagine forense si trattano solo con modelli locali.

APPROFONDIMENTO 4

Prompt, RAG e fine-tuning a confronto

Per adattare un modello linguistico a un SOC esistono tre leve, da considerare in quest'ordine. Il **prompt engineering**, cioè istruzioni, esempi e formato, ha costi contenuti e si modifica senza riaddestrare il modello, purché ogni modifica sia versionata e provata (approfondimento 1). La RAG aggiunge la conoscenza dell'organizzazione senza modificare il modello: si aggiorna reindicizzando i documenti, e le risposte possono citare le fonti (Modulo 14). Il **fine-tuning** modifica i pesi ed è utile per formati di output molto specifici o per il lessico di dominio, ma non per fornire al modello informazioni fattuali aggiornate. Richiede inoltre dati curati, risorse di calcolo e manutenzione, fissa la conoscenza alla data dell'addestramento e rende più difficile stabilire che cosa il modello abbia appreso.

Scenario didattico

Obiettivo: note di triage uniformi, in italiano, con campi fissi. Un prompt con tre esempi copre la maggior parte dei casi; l'aggiunta della RAG sui playbook fa sì che le note citino la procedura corretta. Il fine-tuning, valutato, viene scartato perché il miglioramento atteso non compensa il costo di riaddestrare il modello a ogni modifica delle procedure.

Cautele

Il fine-tuning su dati di indagini effettive incorpora quei dati nei pesi, e il modello può riprodurli: la distribuzione del modello può quindi comportare la comunicazione di quei dati (Modulo 15). La RAG, che mantiene i dati fuori dal modello, semplifica cancellazioni e controllo degli accessi.

APPROFONDIMENTO 5

Strumenti e permessi di un agente

Un **agente** è un modello linguistico inserito in un ciclo: osserva, pianifica, chiama uno strumento, osserva il risultato e prosegue. Gli strumenti concessi, per esempio una query al SIEM, una verifica di reputazione o l'apertura di un ticket, delimitano ciò che l'agente può fare e quindi i danni che un'allucinazione o una prompt injection (Modulo 13) possono provocare attraverso di esso. Le regole di progetto sono il **privilegio minimo** per ogni strumento, la separazione tra lettura e azione con le azioni subordinate ad approvazione (Modulo 5), account di servizio dedicati, limiti al numero di chiamate, al tempo e al costo di ogni esecuzione e la registrazione di ogni chiamata con input e output.

Scenario didattico

Si consideri un agente di arricchimento degli alert che dispone di tre strumenti: ricerca nel SIEM in sola lettura, reputazione di indirizzi e domini, consultazione dell'inventario degli asset. Per ogni alert produce una scheda con le evidenze raccolte e una proposta motivata, senza poter chiudere l'alert né bloccare risorse, e un limite di 20 chiamate per alert ne interrompe i cicli.

Metodo

In fase di progetto si valuta, per ogni strumento concesso, che cosa potrebbe fare l'agente se un attaccante controllasse interamente il testo che legge. Se la valutazione indica un danno rilevante, lo strumento va limitato o subordinato ad approvazione.

APPROFONDIMENTO 6

Casi d'uso dei modelli linguistici nel SOC

Il modello è utile quando produce bozze che un esperto verifica in meno tempo di quanto impiegherebbe a scriverle. Si prestano a questo uso la sintesi e la riformulazione, come il riassunto di un caso per il passaggio di turno o la versione di un report tecnico per il vertice. Vi rientrano anche la generazione di query e regole da descrizioni in linguaggio naturale, purché seguita da test, la prima stesura di report e timeline, la spiegazione di comandi e script offuscati e la consultazione della documentazione interna tramite RAG. Presentano invece rischi elevati le decisioni autonome di contenimento, i giudizi definitivi su benignità o malevolenza senza verifica, i calcoli e le statistiche e ogni compito in cui un errore passerebbe inosservato.

Scenario didattico

Al passaggio di turno il copilota riassume le ore precedenti a partire da ticket e chat operative: casi aperti, azioni svolte, attività in sospeso. Il turno entrante verifica il riassunto scorrendo i casi citati, con un costo basso rispetto al tempo risparmiato, anche se il rischio di errore resta.

Limiti

Chiedere al modello se un allegato è malevolo, senza sandbox né analisi, produce risposte formulate con sicurezza in entrambe le direzioni e difficili da controllare. Il verdetto si costruisce con gli strumenti dei Moduli 9 e 11; al modello si può affidare la spiegazione del risultato.

LABORATORIO · ESERCITAZIONE

Triage e detection con un LLM locale

Svolgimento (Ollama)

1. Triage guidato di un alert esportato dal SIEM di laboratorio
2. Generazione di una regola Sigma per la tecnica individuata
3. Conversione con sigma-cli e verifica su log di test con un attacco simulato e traffico benigno
4. Confronto critico tra i contributi utili del modello e le allucinazioni riscontrate

Consegna

- Regola validata con esiti dei test
- Registro delle allucinazioni riscontrate
- Prompt usati, con una nota su cosa ha funzionato

CHIUSURA DEL MODULO

Letture e autovalutazione

Letture obbligatorie

- Documentazione di Microsoft Security Copilot
- NIST AI 600-1, *Generative Artificial Intelligence Profile* (2024)
- OWASP, *LLM AI Cybersecurity & Governance Checklist*

Domande di autovalutazione

1. Cosa significa grounding di un copilota sui dati del tenant?
2. Come si valida una regola generata da un LLM prima del deployment?
3. Quali dati non dovrebbero transitare verso un LLM cloud?
4. Quali compiti di livello 1 sono candidati all'autonomia agentica?
5. Come si misura l'allucinazione in un output investigativo?

FINE DELLA LEZIONE · MODULO 06

Grazie

LLM e agenti nel SOC · Corso «Incident Response potenziata dall'AI»

Domande e materiali: ricevimento del docente. Profilo e contatti: vincenzocalabro.it

Vincenzo Calabrò

CYBER SECURITY · DIGITAL FORENSICS

VINCENZOCALABRO.IT