Malware Detection with ChatGPT

Massimiliano Capparuccia

CONTENUTI

01. Introduzione

OBIETTIVI del PROGETTO

- **Rilevazione di malware:** L'obiettivo del progetto è la rilevazione affidabile di malware in un sistema informatico, attraverso l'utilizzo di ChatGPT come strumento di analisi dei log prodotti dalle macchine esaminate.

- **Valutazione delle capacità:** L'altro obiettivo del progetto è quello di esaminare le capacità di ChatGPT, in generale degli LLM, nell'ambito della rilevazione dei malware e nel suo possibile contributo nella sicurezza informatica

MOTIVAZIONI della RICERCA

- **Uso di ChatGPT:** innanzitutto ChatGPT è facilmente accessibile da tutti, garantendo una vasta diffusione e utilizzo. Inoltre, la sua notevole capacità di comprensione del linguaggio naturale e di analisi testuale lo rende potenzialmente uno strumento efficace per l'analisi dei log per la rilevazione dei malware

- **Sfruttare le Tecnologie Avanzate:** la scelta di guardare questo ambito di tecnologie in rapidissima evoluzione, come i LLM, è motivata dalla necessità di affrontare in modo efficace le sfide sempre più complesse nell'ambito della sicurezza informatica.

02. LLM & Malware Detection

LARGE LANGUAGE MODELS (LLM)

- **Generative Deep Learning:** Il Generative Deep Learning è una sottocategoria del Deep Learning che si concentra sulla generazione di contenuti come testo, immagini e altro. Questi modelli sono progettati per apprendere dai dati di addestramento e generare nuovi dati che abbiano lo stesso stile, la stessa struttura e le stesse caratteristiche dei dati con cui sono stati addestrati.

- **Large Language Models:** Sono modelli di deep learning, addestrati su enormi set di dati per comprendere, generare e predire contenuti basati su testo. Utilizzano il Generative Deep Learning per svolgere il loro ruolo nell'elaborazione del linguaggio naturale.

ARCHITETTURA TRANSFORMER

- **Architettura Transformer:** I LLM si appoggiano sull'architettura Transformer, che consiste in reti neurali composte da encoder e decoder con abilità di self-attention. Utilizzano "token" come unità di base per elaborare il testo. Un token può rappresentare una singola parola, un carattere, a seconda del modello.

- **Self-Attention:** La self-attention è un concetto chiave nei modelli di linguaggio come i Transformer. Consente a ciascun token di valutare quanto gli altri token di input sono rilevanti all'interno di un contesto. Questa tecnica è cruciale per catturare relazioni e contesto tra le parole nei modelli di linguaggio, migliorando la comprensione e la generazione del testo.

LEARNING dei LLM

- **Learning:** L'addestramento delle reti neurali basate su Transformer avviene su ampi corpus di dati di alta qualità. Durante l'addestramento, il modello regola i parametri in modo che possa prevedere con precisione il token successivo sulla base delle sequenze precedenti di token di input. Questo processo fa uso di tecniche di self-supervised learning e unsupervised learning.

- **Fine-Tuning:** Una volta addestrati, i LLM possono essere facilmente adattati per eseguire un compito specifico utilizzando un set di dati relativamente piccolo, attraverso un processo noto come fine-tuning.

MALWARE DETECTION

Le tecniche più utilizzate di rilevamento dei malware:

- **Signature based detection:** E' il metodo più utilizzato attualmente nelle soluzioni antivirus. Questo approccio richiede la manutenzione di un database di firme di malware noti che deve essere aggiornato regolarmente all'evolversi delle minacce, se non è presente il malware non viene rilevato. Contro questo metodo vengono utilizzate tecniche di offuscamento.

- **Behaviour based detection:** Questo approccio si concentra sulle azioni compiute dal malware durante l'esecuzione. In sistemi basati sul comportamento, nella fase di learning vengono fatti esaminare i comportamenti di malware e file benigni, nella fase di test il modello cerca di distinguerli

- **Statistical based detection:** Il rilevamento del malware si basa sulle proprietà statistiche derivate dalle caratteristiche dei programmi. Questa tecnica viene utilizzata per lo più nella ricerca dei malware metamorfici.

MALWARE ANALYSIS

- **Static Analysis:** L'analisi statica del software viene effettuata senza eseguire il programma. Esempi di informazioni ottenute dall'analisi statica includono sequenze di opcode, grafi di flusso di controllo, ecc.

- **Dynamic Analysis:** L'analisi dinamica richiede l'esecuzione del programma, spesso in un ambiente virtuale. Le informazioni che vengono cercate sono: chiamate API, chiamate di sistema, tracce di istruzioni, modifiche del registro, scritture in memoria, ecc.

- **Hybrid Analysis:** Le tecniche ibride combinano aspetti dell'analisi statica e dinamica. Alcuni approcci ibridi estraggono caratteristiche dinamiche durante la fase di learning, ma utilizzano l'analisi statica per la fase di test. Altri approcci ibridi definiscono caratteristiche di malware utilizzando un approccio denominato "Malware DNA" (Mal-DNA) e poi estraggono dinamicamente le caratteristiche comportamentali.

03. Prompt Engineering

04. Metodologie

IPOTESI di MACRO FASI del SISTEMA

Extraction

Estrazione, dalla macchina infetta, di tutti i log di sistema e di sicurezza.

Filtering

Filtraggio e strutturazione dei log, conservando solo quelli rilevanti.

Prompt Engineering

Ottimizzazione del prompt di ChatGPT per massimizzare l'efficacia nella generazione di risposte pertinenti.

Validation

Sottoporre i log estratti a ChatGPT ed effettuare la validazione dei risultati ottenuti al fine di verificare l'accuratezza delle informazioni

RISORSE

- **STIX:** STIX, acronimo di "Structured Threat Information eXpression," è uno standard aperto nel campo della cybersecurity progettato per rappresentare e condividere informazioni sulle minacce informatiche in modo strutturato e coerente. Mira a fornire un linguaggio comune per la descrizione e lo scambio di dati sulle minacce informatiche tra organizzazioni e strumenti di sicurezza. Un possibile utilizzo di STIX potrebbe essere per standardizzare la rappresentazione delle informazioni nei log relativi alle attività del sistema e alle minacce potenziali, quindi mappare i dettagli dei log alle strutture di dati STIX.

(*https://oasis-open.github.io/cti-documentation*)

05. Fonti

PAPERS

Questi sono i papers da cui ho preso le informazioni

- **Improving Language Understanding by Generative Pre-Training, OpenAI (2018) :** Abbastanza dettagliato su LLM e NLP, spiega Transformers, learning e Fine-Tuning
- **Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study (2023):** Molto carino spiega come bypassare filtri di ChatGPT, in sostanza mostra il potenziale e spiega il modo con cui fare prompt engineering.
- **Attention Is All You Need, Google (2017):** Spiega in dettaglio l'architettura dei transformer, encoder e decoder
- **A comparison of static, dynamic, and hybrid analysis for malware detection (2017):** Spiega quali sono le tecniche utilizzate per la malware detection e analysis

PAPERS

- **BERT-Log: Anomaly Detection for System Logs Based on Pre-trained Language Model (2022):** Paper fondamentale a mio parere per il progetto, mostra come hanno utilizzato un pre-trained LLM (BERT) con il fine-tuning al fine di avere un LLM che rileva anomalie nei log. Molti spunti carini sul log parsing, spiegano come hanno strutturato questi dati
- **Revolutionizing Cyber Threat Detection with Large Language Models (2023):** Spiega come hanno introdotto il SecurityLLM model, configurato, testato e quali sono stati i risultati. Viene utilizzato per la detection delle minacce in generale, in alcuni punti cita la malware detection ad esempio di BERT-Log. Fa confronti fra LLM e altri modelli di Deep Learning

PAPERS

- **LAnoBERT : System Log Anomaly Detection based on BERT Masked Language Model (2021):** Molto bello, spiegano come sono riusciti a creare, addestrare e testare LAnoBERT, una variante del modello di BERT per anomaly detection from logs (L'ho trovato abbastanza complesso)
- **DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning:** Introducono DeepLog, un modello di rete neurale che impara automaticamente i pattern standard dei log e segnala le anomalie quando i log deviano dal pattern della normale esecuzione. Pieno di spunti su come gestire i log da utilizzare ma l'ho trovato complesso
- **MalBERT: Using Transformers for Cybersecurity and Malicious Software Detection:** Malware detection su Android utilizzando BERT, fa una bella panoramica sui Transformers. Risultati trovati molto buoni, bel paper da cui prendere spunto
- **GPT-3: Its Nature, Scope, Limits, and Consequences (2020):** Fa una panoramica sulla natura di ChatGPT e sui suoi limiti. Dimostrano che non supera il test di Turing, cioè un criterio per determinare se è in grado di esibire un comportamento intelligente

LINK INTERESSANTI

- https://softteco.com/blog/bert-vs-chatgpt
- https://securelist.com/ioc-detection-experiments-with-chatgpt/108756/
- https://threatmon.io/blog/chatgpt-and-malware-analysis-threatmon/
- https://medium.com/@tonydain9_78432/word-embedding-chat-gpt-c96702f864e0
- https://book.premai.io/state-of-open-source-ai/
- https://www.gptsecurity.info/security-papers
- https://ieeexplore.ieee.org/document/9486307 (Mancante)
- https://systemweakness.com/chatgpt-for-threat-hunting-automation-995f9022c6b3
- https://oasis-open.github.io/cti-documentation/stix/intro
- https://otx.alienvault.com/endpoint-security/welcome
- https://openai.com/blog/openai-cybersecurity-grant-program