**Fake Architecture Orchestrator**

Immagine che contiene Elementi grafici, design, cerchio, grafica

Descrizione generata automaticamente

**Orlando De Bernardis**

**Pierantonio Carrozzini**

INDICE

1. INTRO PROBLEMI DI SICUREZZA
2. IL NOSTRO PROGETTO IN SINTESI (LA CONSEGNA DEL PROF)
3. IDE PY, DOCKER, PARSEXML, YOLO E TERRAFORM
4. LA SOLUZIONE PROPOSTA
5. CONCLUSIONI

Nel panorama sempre più digitale ed interconnesso del mondo contemporaneo, la protezione dei dati è diventata una priorità critica per le aziende di ogni settore. La rapida evoluzione della tecnologia ha aperto nuove opportunità per l'innovazione e la crescita, ma ha anche introdotto nuove sfide legate alla sicurezza informatica. Al giorno d'oggi, la salvaguardia delle informazioni sensibili è diventata un elemento fondamentale per il successo e la reputazione di un'azienda.

Queste minacce variano da attacchi di malware sofisticati a frodi informatiche, e le conseguenze di un'eventuale violazione della sicurezza possono essere devastanti, causando danni finanziari, perdita di fiducia dei clienti e danneggiando la reputazione aziendale.

Una delle pratiche più diffuse al giorno d’oggi nel panorama della sicurezza informatica è quella di utilizzare dati fittizi per proteggere dati reali e prende il nome di “honeytoken”. Si tratta di record falsi che vengono aggiunti a database legittimi e che forniscono agli amministratori di rete e agli esperti di sicurezza un modo proattivo di monitorare attività sospette o tentativi di violazione della sicurezza prima che possano causare danni reali. Eventuali aggressori, infatti, potrebbero dedicare tempo e risorse a cercare di compromettere questi elementi falsi invece che prendere di mira risorse di valore effettivo. Tuttavia, è importante gestire attentamente l’implementazione di tali honeytoken per evitare false allerte o esposizione involontaria di dati sensibili.

L’obiettivo di questo progetto è, dunque, quello di creare facilmente un’infrastruttura “fake” da un diagramma architetturale e di istanziare un container Docker per ogni tipo di risorsa individuata tramite la creazione di un “terraform plan” adeguato.

La soluzione da noi proposta va a soddisfare tutte le specifiche di progetto richieste, permettendo all’utente di scegliere se fornire il diagramma architetturale tramite un opportuno file “.xml” o tramite una vera e propria immagine in formato “.png” o “.jpeg”.

Il tutto è stato realizzato in Python utilizzando PyCharm CE come IDE, ed inoltre sono stati necessari strumenti aggiuntivi come Docker, Terraform, YoloV5 e Roboflow (questi ultimi necessari per il modello di rete neurale per analizzare i file immagine).

Immagine che contiene schermata, testo, nero, Carattere

Descrizione generata automaticamente

Una volta eseguito lo script viene dapprima caricato da remoto il modello di rete neurale e successivamente è chiesto all’utente come si intende procedere:

* Digitando **“xml”** si sceglierà di istanziare le varie componenti e, dunque, di generare il terraform plan fornendo in input un diagramma in formato xml
* Digitando “**img”** si procederà tramite l’analisi e il riconoscimento delle risorse da un file immagine
* Digitando **“compose”** ….
* PARTE PER XML
* PARTE PER IMG

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Digitando la sigla “***img***” l’utente si troverà dinanzi a questa schermata, nella quale gli verrà chiesto di inserire da tastiera il *path* dell’immagine da analizzare. Digitando “default” verrà invece utilizzata l’immagine di test prevista nella directory di progetto.