Proyecto. Asistente para MITRE ATT&CK utilizando LLMs y RAG (Retrieval-Augmented Generation)

---

1. Descripción y objetivos

Se pretende desarrollar un prototipo de asistente inteligente para ayudar en el trabajo con MITRE ATT&CK (*Adversarial Tactics, Techniques, and Common Knowledge*)

- MITRE ATT&CK es una **base de conocimientos** que agrupa y documenta las **tácticas y técnicas** utilizadas por los atacantes en el ciclo de vida de un ataque.
- ATT&CK define 14 tácticas que representan las finalidades últimas y los objetivos que pretende alcanzar un atacante con sus acciones (explican el **por qué**)
    - ATT&CK organiza las tácticas en tres **matrices** especializadas en diferentes entornos y plataformas (*enterprise*, *mobile*, *industrial control systems(ICS)*)
- Para cada táctica se identifican y documentan las técnicas y subtécnicas empleadas por los atacantes
    - Para cada técnica o subtécnica se incluye una descripción textual, un listado de procedimientos de ataque vinculados con esa técnica, un listado de posibles técnicas de mitigación y de detección, junto con referencias externas adicionales

Objetivos

1. Conocer y utilizar una librería de LLMs para implementar un sistema RAG sencillo (LangChain en este caso)
2. Aplicar las ideas de RAG para implementar un asistente inteligente en un entorno relativamente realista (asistencia con las técnicas de MITRE ATT&CK)

---

2. Herramientas y recursos a utilizar

- Volcado en JSON de un resumen de las 656 técnicas y subtécnicas de la matriz *enterprise* de la versión actual de MITRE ATT&CK: techniques_enterprise_attack.json
    - Cada entrada tiene un identificador, un título, una descripción y la URL de su entrada en la web de MITRE ATT&CK.

- Adicionalmente, se incluye la lista de tácticas que emplean cada técnicas, las plataformas afectadas y el tipo fuentes de datos (*datasources*) que proporcionan información sobre cada técnica.
- Se incluyen una descripción textual de las posibles estrategías de detección y una lista de mitigaciones a emplear con cada técnica.

  Ejemplo:


```
{
    "id": "T1037",
    "url": "https://attack.mitre.org/techniques/T1037",
    "name": "Boot or Logon Initialization Scripts",
    "description": "Adversaries may use scripts automatically executed at
boot ...",
    "detection": "Monitor logon scripts for unusual access ...",
    "tactics": ["Persistence", "Privilege Escalation"],
    "platforms": ["macOS","Windows", "Linux", "Network"],
    "datasources": [
      "File: File Modification",
      "Windows Registry: Windows Registry Key Creation",
      "Command: Command Execution",
      "File: File Creation",
      "Process: Process Creation",
      "Active Directory: Active Directory Object Modification"
    ],
    "mitigation_methods": [
      {
        "id": "M1024",
        "url": "https://attack.mitre.org/mitigations/M1024",
        "name": "Restrict Registry Permissions",
        "description": "Restrict the ability to modify certain hives or
keys in the Windows Registry."
      },
      {
        "id": "M1022",
        "url": "https://attack.mitre.org/mitigations/M1022",
        "name": "Restrict File and Directory Permissions",
        "description": "Restrict access by setting directory and file
permissions that are not ..."
      }
    ]
  }
```


- Dataset `dattaraj/security-attacks-MITRE` disponible en Huggingfaces con ejemplos del tipo de entradas y salidas a utilizar: [security-attacks-MITRE]
  - Ejemplo:

```
ENTRADA DEL USUARIO:
An administrator noticed that a user account, which belongs to an
employee on leave, was used to access sensitive files at odd hours

RESPUESTA DEL ASISTENTE:
This scenario matches MITRE technique T1078 - Valid Accounts, part of the
Defense Evasion tactic.
It could indicate account compromise.
Mitigation steps include implementing the principle of least privilege,
regularly reviewing and disabling unnecessary accounts, using multi-
factor authentication, and monitoring for unusual account activity.
```

- **Importante:** este dataset se aporta como ejemplo de las entradas y salidas del asistente, no es necesario utilizarlo, ni se espera que las salidas proporcionadas por el asistente desarrollado sean idénticas a las de estos ejemplos

---

3. Desarrollo de trabajo

Se pide implementar un Chatbot que aplique RAG (*Retrieval-Augmented Generation*) sobre la colección de técnicas de MITRE ATT&CK para implementar un asistente capaz (1) de diagnosticar qué técnica o técnicas de MITRE ATT&CK se están utilizando en el escenario descrito en la entrada del usuario y (2) de ofrecer una lista de mitigaciones contramedidas adaptadas al escenario descrito y las técnicas de ataque implicadas.

Se proponen dos alternativas, con diferentes calificaciones máximas.

1. Sistema **RAG simple** (replicando el esquema del ejemplo incluido en el resumen de LangChain de Moovi)
   - No tiene historia y proporciona una respuesta/solución al escenario descrito sin tener en cuenta el contexto
   - **Evaluación: hasta 7,5 puntos**
2. Sistema **RAG conversacional**
   - Mantiene un histórico de la conversación y una vez proporcionada la respuesta/solución al escenario descrito, permitirá preguntas y aclaraciones al respecto de la respuesta/solución propuesta.
   - **Evaluación: hasta 10 puntos**

4. Tareas a realizar

Las tareas concretas a realiza en la práctica serán las siguientes:

1. Diseñar e implementar la carga y procesamiento del fichero JSON con el listado de técnicas y subtécnicas de MITRE ATT&CK
   - Debe decidirse qué elementos de los objetos JSON incluir en el *vector store* y cuales en el *metadata* vinculado a los *Document*
   - **Importante:** hay varios atributos de los objetos JSON que ofrecen descripciones textuales que pueden ser relevantes para proporcionar contexto al RAG (atributos `description`, `detection` y las descripciones de los `mitigation_methods`)

2. Implementar el Chatbot, bien siguiendo el esquema **RAG simple** o el **RAG conversacional** descrito en el punto anterior
    - En ambos casos es necesario **definir un *Prompt* específico** que proporcione el tipo de salida deseado (diagnóstico + propuesta de mitigaciones) y decidir **qué elementos** de los documentos originales se van a incluir **como contexto**.
    - **Importante:** Respecto a las descripciones de los `mitigation_methods`, si no se incluyen como `page_content` en los *Document* generados, sus descripciones se pueden añadir *a posteriori* en el momento de confeccionar el contexto del *prompt* a enviar al LLM.
3. Ejecución del sistema y pruebas
    - Realizar y documentar algunas pruebas informales
    - **Nota:** no se pide ningún tipo de evaluación cuantitativa, sólo pruebas informales

---

5. Normas de entrega

Trabajo **individual** o en **parejas**

Entregable:

- Ficheros Python que conforman el sistema desarrollado
- Memoria **breve** con la siguiente estructura:
    - Descripción del problema
    - Descripción de la implementación
    - Ejemplo de funcionamiento y descripción de las pruebas realizadas
    - Conclusiones y problemas encontrados (posibles mejoras, idoneidad de la herramienta, etc)
    - Bibliografía consultada

  Nota: Incluir nombre, DNI y e-mail de todos los alumnos del grupo en la portada

Fecha de entrega: Hasta miércoles, **29/1/2025**, 23:50