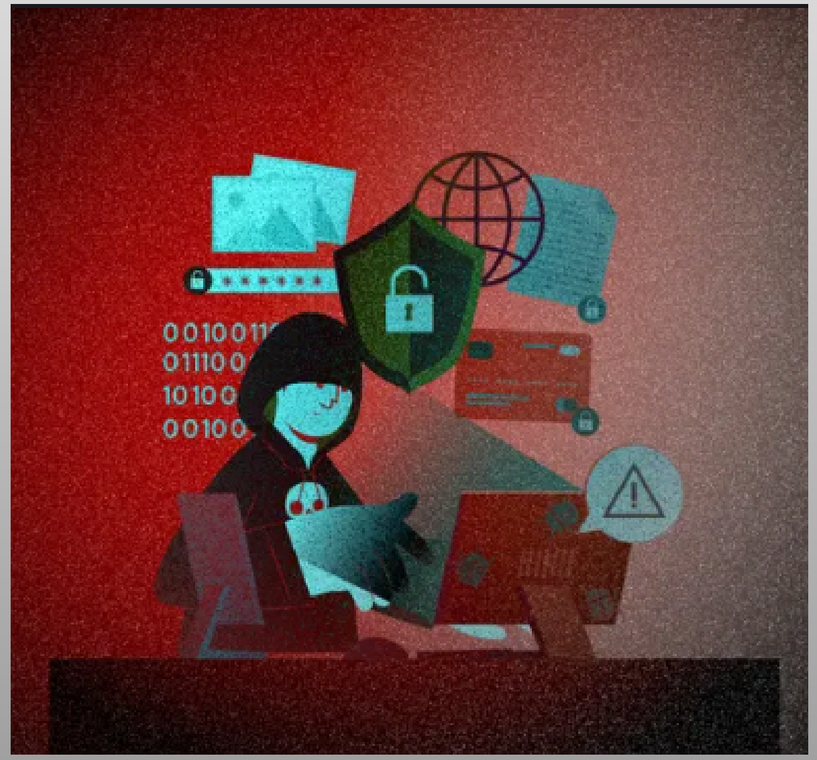
AI-Enhanced Breach and Attack Simulator



By Neha Kasera

Background and motivation

- The growing sophistication and frequency of cyberattacks.
- Organizations need to continuously test and improve their defenses.
- Manual testing is time-consuming and resource-intensive.

Objective

To develop an automated tool that simulates real-world cyberattacks, generates synthetic security logs, and leverages AI for detection and analysis.

Problem Statement

Challenges in Cybersecurity

- Security teams struggle to keep up with evolving attack techniques.
- Traditional penetration testing is not continuous and can miss new threats.
- Lack of realistic, labeled data for testing detection systems.
- Difficulty in mapping security events to frameworks like MITRE ATT&CK.

Impact

- Increased risk of undetected breaches.
- Delayed incident response.
- Compliance and reporting challenges.

Solution Overview

Project Overview

- Developed a Python-based simulation tool for generating attack scenarios and logs.
- Mapped events to MITRE ATT&CK tactics and techniques.
- Integrated AI for breach narration and anomaly detection.

Key features

- Scenario-based Attack Simulation: Predefined attack chains (e.g., Credential Theft, Ransomware, Phishing).
- Synthetic Log Generation: Realistic logs with contextual metadata (timestamp, IP, user, event, MITRE mapping).
- AI-Powered Analysis:
- Attack narrative generation using language models.
- Anomaly detection using machine learning.
- Visualization: Interactive dashboards for log review and analysis.

anomaly_detector.py > ...

```python
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import IsolationForest

class AnomalyDetector:
    def __init__(self):
        self.vectorizer = CountVectorizer()
        self.model = IsolationForest(contamination=0.2)

    def train(self, logs):
        X = self.vectorizer.fit_transform(logs['event'])
        self.model.fit(X)

    def predict(self, logs):
        X = self.vectorizer.transform(logs['event'])
        logs['anomaly'] = self.model.predict(X)
        logs['anomaly'] = logs['anomaly'].map({1: 'Normal', -1: 'Anomaly'})
        return logs

```

anomaly_detection.py

breach_narrator.py > ...

```python
from transformers import pipeline

# Use FLAN-T5 (lightweight) or any summarization model
generator = pipeline("text2text-generation", model="google/flan-t5-base")

def narrate_breach(events):
    input_text = "Explain this attack in detail: " + " -> ".join(events)
    result = generator(input_text, max_length=200, do_sample=True)
    return result[0]['generated_text']

```

breach_narrator.py

app.py > ...

```python
import streamlit as st
from attack_scenarios import SCENARIOS
from log_generator import generate_logs
from breach_narrator import narrate_breach
from anomaly_detector import AnomalyDetector

# Streamlit Page Configuration
st.set_page_config(page_title="AI Breach Simulator", layout="wide")
st.title("AI-Enhanced Breach and Attack Simulator")

# Scenario Selection
scenario = st.selectbox("Choose a breach scenario:", list(SCENARIOS.keys()))

if st.button("Simulate Attack"):
    events = SCENARIOS[scenario]

    # Generate synthetic logs with MITRE tactic mapping
    logs = generate_logs(events)

    # Display Logs with MITRE Tactic Mapping
    st.subheader("Generated Logs (with MITRE tactics)")
    st.dataframe(logs, use_container_width=True)

    # AI-generated Attack Narrative
    st.subheader("AI Attack Narrative")
    story = narrate_breach(events)
    st.write(story)

    # Anomaly Detection
    st.subheader("Anomaly Detection")
    detector = AnomalyDetector()
    detector.train(logs)
    annotated_logs = detector.predict(logs)

    # Display Anomalous Logs with Detection Results
    st.dataframe(annotated_logs, use_container_width=True)

```

app.py

attack_scenarios.py > ...

```python
SCENARIOS = {
    "Credential Theft": [
        "User logs in from new location",
        "Multiple failed login attempts",
        "MFA disabled",
        "New login from IP 192.168.56.101",
        "PowerShell used to list AD users",
        "Password reset request from unfamiliar location",
        "Brute-force attack detected",
        "Suspicious login pattern (same time every day)",
        "VPN login from unusual IP",
        "Login from TOR network"
    ],
    "Lateral Movement": [
        "Remote Desktop session started",
        "Accessed C$ share on another machine",
        "Used PSExec to run remote command",
        "Account used on multiple systems",
        "Suspicious admin tool downloaded",
        "Remote PowerShell command execution",
        "Admin account used on non-administrative systems",
        "Account compromise followed by lateral movement",
        "Credential dump from system memory",
        "Mimikatz or other credential dump tool detected"
    ],
    "Data Exfiltration": [
        "Large file compressed",
        "File uploaded to unknown domain",
        "Tor process detected",
        "External device plugged in",
        "Traffic to Dropbox observed",
        "Massive data download initiated",
        "Large volume of data being transferred at odd hours",
        "Unusual cloud storage access",
        "Unusual HTTP/FTP traffic detected",
        "Data transferred to external IP address"
    ],
    "Privilege Escalation": [
        "Exploit found in system to gain root/admin access",
        "Elevation of privileges detected using local exploit tools",
        "Sudo command executed by non-privileged user",
        "Unauthorized use of admin credentials",
        "System security settings changed unexpectedly",
        "Zero-day exploit attempt detected",
        "Modification of /etc/sudoers or similar privilege files",
        "Privilege escalation via unpatched vulnerability",
        "User added to an admin group without proper authorization"
    ],
    "Ransomware Attack": [
        "Unusual file encryption activity detected",
        "Suspicious file extensions being added to files",
        "Ransomware communication detected (e.g., C2 server traffic)",
        "Files being renamed with a specific ransom extension",
        "Suspicious process named after a known ransomware family",
        "Encrypted files reported by users",
        "Ransom note found on multiple systems",
        "Malicious attachment in email opened",
        "Email received with suspicious subject line and attachment"
    ],
    "Phishing Attack": [
        "Suspicious email with unknown attachment received",
        "Email from seemingly legitimate source requesting credentials",
        "Unusual domain name in email sender's address",
        "User clicked on link from phishing email",
        "Spoofed email masquerading as IT support",
        "Suspicious email with a sense of urgency (e.g., 'Immediate Action Required')",
        "Unusual file type attached to email (e.g., .exe, .zip)",
        "Redirect from known site to a phishing page",
        "Fake login page detected"
    ],
    "Denial of Service (DoS)": [
        "Unexpected increase in network traffic from one source",
        "Server is unreachable due to high traffic",
        "High rate of HTTP requests detected",
        "ICMP flood detected",
        "UDP flood observed",
        "Server performance degraded due to excessive requests",
        "Network device performance impacted due to DDoS",
        "Suspicious volume of SYN packets being sent"
    ],
    "Supply Chain Attack": [
        "Malicious code inserted into third-party software update",
        "Unexpected changes in software dependencies",
        "Abnormal activity in a trusted vendor's network",
        "Compromised update downloaded from a trusted source",
        "Suspicious connection made to vendor's server",
        "Unexpected change in software version from trusted repository",
        "Legitimate software package modified with malicious code"
    ],
    "Insider Threat": [
        "Employee accessing sensitive data without authorization",
        "Employee transferring files to an external device",
        "Unusual pattern of access to critical infrastructure",
        "Employee leaves organization with sensitive data",
        "Increased use of privileged access by non-administrators",
        "Employees accessing files not relevant to their job role",
        "Unauthorised user gaining access via compromised employee account",
        "User logs in after hours without valid reason"
    ]
}
```

attack_scenarios.py

log_generator.py > generate_logs

```python
from faker import Faker
import pandas as pd
from mitre_mapping import MITRE_MAP  # Make sure this file exists and is correctly mapped

fake = Faker()

def generate_logs(events, user="intern"):
    logs = []
    for event in events:
        # Try to find a matching MITRE tactic
        tactic = "Unknown"
        for key in MITRE_MAP:
            if key in event:
                tactic = MITRE_MAP[key]
                break

        # Create log entry
        logs.append({
            "timestamp": fake.iso8601(),
            "user": user,
            "event": event,
            "tactic": tactic,
            "ip": fake.ipv4(),
            "location": fake.city(),
            "hostname": fake.hostname(),
            "os": fake.random_element(["Windows", "Linux", "MacOS", "Ubuntu", "Android"]),
            "user_agent": fake.user_agent(),
            "device": fake.random_element(["Desktop", "Laptop", "Mobile", "Server"]),
            "status": fake.random_element(["Success", "Failure"])
        })
    return pd.DataFrame(logs)

```

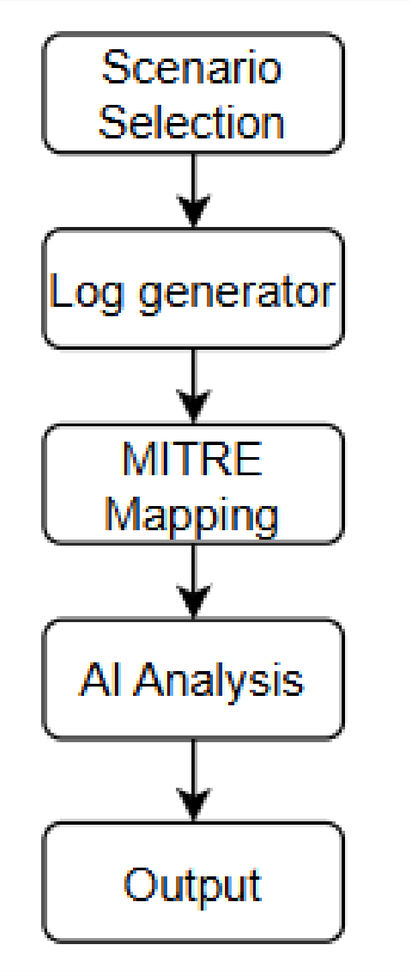
log_generator.py

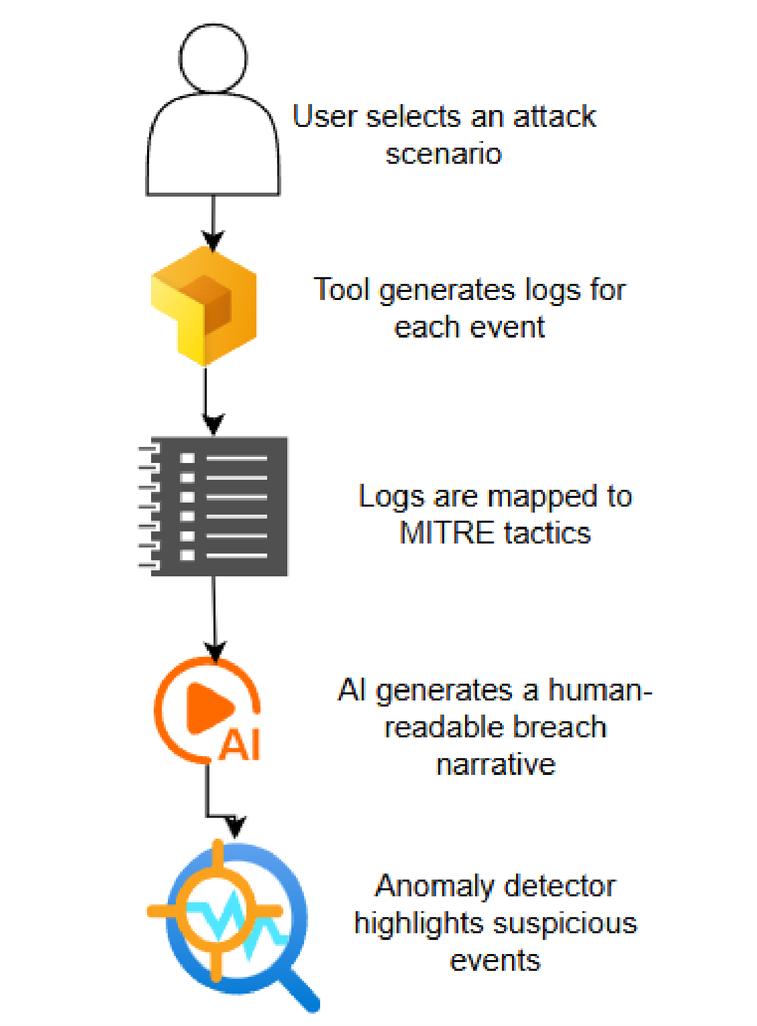
mitre_mapping.py > ...

```python
MITRE_MAP = {
    "User logs in from new location": "Initial Access",
    "Multiple failed login attempts": "Credential Access",
    "MFA disabled": "Defense Evasion",
    "New login from IP": "Initial Access",
    "PowerShell used to list AD users": "Discovery",
    "Password reset request from unfamiliar location": "Credential Access",
    "Brute-force attack detected": "Credential Access",
    "Suspicious login pattern (same time every day)": "Credential Access",
    "Login from TOR network": "Initial Access",
    "Remote Desktop session started": "Lateral Movement",
    "Accessed C$ share on another machine": "Lateral Movement",
    "Used PSExec to run remote command": "Lateral Movement",
    "Account used on multiple systems": "Credential Access",
    "Suspicious admin tool downloaded": "Execution",
    "Remote PowerShell command execution": "Lateral Movement",
    "Admin account used on non-administrative systems": "Lateral Movement",
    "Credential dump from system memory": "Credential Access",
    "Mimikatz or other credential dump tool detected": "Credential Access",
    "Large file compressed": "Collection",
    "File uploaded to unknown domain": "Exfiltration",
    "Tor process detected": "Command and Control",
    "External device plugged in": "Collection",
    "Traffic to Dropbox observed": "Exfiltration",
    "Massive data download initiated": "Exfiltration",
    "Large volume of data being transferred at odd hours": "Exfiltration",
    "Unusual cloud storage access": "Exfiltration",
    "Data transferred to external IP address": "Exfiltration",
    "Exploit found in system to gain root/admin access": "Privilege Escalation",
    "Sudo command executed by non-privileged user": "Privilege Escalation",
    "Unauthorized use of admin credentials": "Privilege Escalation",
    "System security settings changed unexpectedly": "Defense Evasion",
    "Zero-day exploit attempt detected": "Initial Access",
    "Modification of /etc/sudoers or similar privilege files": "Privilege Escalation",
    "Privilege escalation via unpatched vulnerability": "Privilege Escalation",
    "Files being renamed with a specific ransom extension": "Impact",
    "Unusual file encryption activity detected": "Impact",
```

Code / Tool breakdown

Flowchart

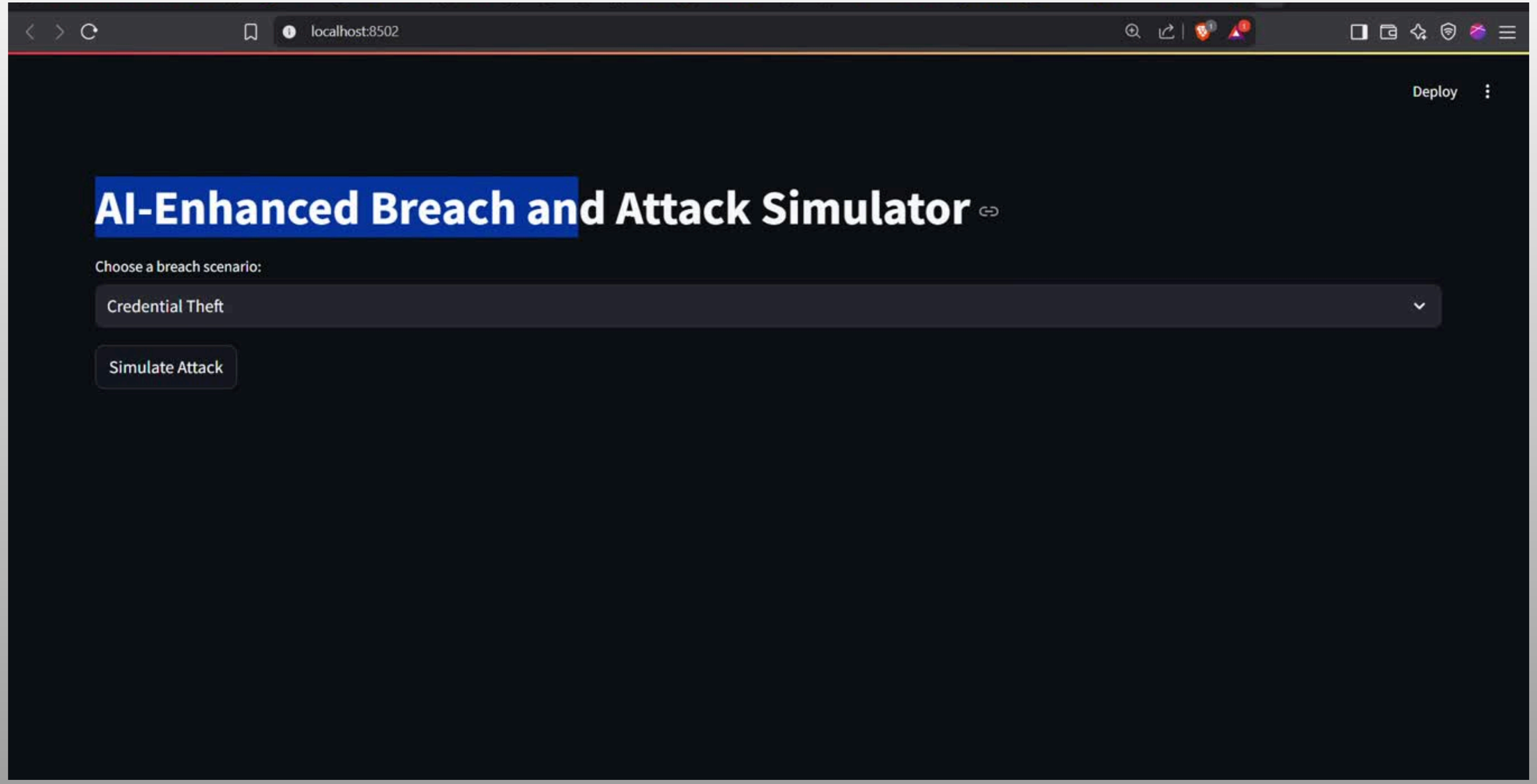




Module_Breakdown

- mitre_mapping.py: Maps security events to MITRE ATT&CK tactics and techniques.
- log_generator.py: Generates synthetic logs for selected attack scenarios.
- attack_scenarios.py: Defines multiple real-world attack chains.
- breach_narrator.py: Uses AI (e.g., FLAN-T5) to generate a narrative of the attack.
- anomaly_detector.py: Detects unusual events using machine learning (Isolation Forest).
- app.py: Provides an interactive interface for scenario simulation and analysis.

Demo video


localhost:8502
Deploy
AI-Enhanced Breach and Attack Simulator
Choose a breach scenario:
Credential Theft
Simulate Attack

Real world use cases

- Security Operations Center (SOC) :

  Use synthetic logs to test SIEM rules and analyst workflows.
- Red Team/Blue Team Exercises :

  Simulate attack chains to train defenders and test detection capabilities.
- Tool Benchmarking :

  Evaluate and tune anomaly detection or alerting systems using labeled, realistic data.
- Compliance & Reporting :

  Demonstrate continuous security validation and MITRE ATT&CK alignment for audits.

Future Enhancements

- Expand Attack Library :
  Add more sophisticated and diverse attack scenarios (e.g., supply chain, cloud attacks).
- Integrate with Real SIEMs :
  Directly feed synthetic logs into tools like Splunk, ELK, or QRadar for live testing.
- Advanced AI Integration:
  Use fine-tuned language models for more accurate breach narratives.
  Implement deep learning for anomaly detection.
- User Customization :
  Allow users to define custom attack chains and log formats.
- Visualization Improvements :
  Add MITRE ATT&CK matrix heatmaps and time-series dashboards.
- Automated Reporting :
  Generate PDF/HTML reports summarizing each simulation.

Conclusion

This project demonstrates the design and implementation of an AI-Enhanced Breach and Attack Simulation Tool that automates the process of simulating cyberattacks, generating realistic security logs, mapping events to the MITRE ATT&CK framework, and leveraging AI for breach narration and anomaly detection.

Thank you