**AI-Powered Security Onion - VM-Based Deployment Plan**

**1. Introduction**

**1.1 Purpose**

This document provides a plan for deploying an **AI-powered Security Onion alternative** on a **virtual machine (VM)** to **import network PCAPs** and showcase AI-driven **threat intelligence, predictive analytics, and risk scoring**.

**1.2 Scope**

* Install and configure a **lightweight Security Onion alternative** on a single VM.
* **Import and analyze PCAP files** instead of live network traffic.
* Demonstrate AI-powered **threat intelligence extraction, risk scoring, and predictive analytics**.
* Deploy a **simple SOC dashboard** for visualizing attack data.

**1.3 Target Audience**

* **Cybersecurity Students & Researchers** – To experiment with AI-driven threat intelligence.
* **Developers & Data Scientists** – To implement machine learning techniques for cybersecurity.
* **Security Operations Analysts** – To automate attack analysis and prioritization.

**2. System Architecture**

**2.1 High-Level Components**

1. **PCAP Import & Parsing**
   * **Tools:** Zeek, Suricata
   * **Function:** Extracts metadata (IP, DNS, HTTP, SSL, Protocols, Attack Signatures, etc.).
2. **AI-Driven Threat Intelligence & Analysis**
   * **NLP-Based Threat Extraction** – Uses **BERT/RoBERTa** to analyze OSINT sources and extract relevant cyber threat intelligence.
   * **Predictive Analytics** – Uses **LSTM, ARIMA** models to forecast future attack trends.
   * **Risk Scoring System** – Uses **Random Forest, XGBoost, and Deep Learning Models** to score threats based on severity and exploitability.
3. **Data Storage & Management**
   * **Elasticsearch** – Stores indexed logs & AI insights.
   * **MongoDB/PostgreSQL** – Stores structured metadata for deeper analysis.
4. **SOC Dashboard**
   * **Framework:** React.js + Kibana + D3.js (for visualizations)
   * **Function:** Displays AI insights, attack paths, and risk scores.

**3. Deployment Plan on Virtual Machine**

**3.1 VM Requirements**

* **Host Machine:** Any system with virtualization support.
* **VM Software:** VirtualBox / VMware / Proxmox / KVM.
* **Guest OS:** Ubuntu 20.04 LTS.
* **Minimum VM Specs:**
  + **CPU:** 4 cores.
  + **RAM:** 8 GB.
  + **Storage:** 100 GB.
  + **GPU:** Not required (unless training AI models).

**3.2 VM Setup & Software Installation**

**3.2.1 Setting Up the VM**

1. **Install Ubuntu 20.04 on VirtualBox/VMware.**
2. **Enable Bridged Network Adapter** to allow external network access.
3. **Update system packages:**
4. sudo apt update && sudo apt upgrade -y

**3.2.2 Installing Core Security Tools**

1. **Install Suricata & Zeek:**
2. sudo apt install -y suricata zeek
3. **Install Elasticsearch & Kibana:**
4. sudo apt install -y elasticsearch kibana
5. **Install Python & AI Dependencies:**
6. sudo apt install -y python3-pip
7. pip3 install torch transformers numpy pandas scikit-learn

**3.3 Importing & Processing PCAP Files**

**3.3.1 Uploading PCAP to VM**

* Transfer PCAP files to the VM using SCP or a shared folder.
* Example:
* scp sample.pcap user@vm\_ip:/home/user/pcap\_imports/

**3.3.2 Processing PCAP Files**

1. **Run Zeek for metadata extraction:**
2. zeek -r /home/user/pcap\_imports/sample.pcap
3. **Run Suricata for threat detection:**
4. suricata -r /home/user/pcap\_imports/sample.pcap -l /var/log/suricata/
5. **Store parsed data in Elasticsearch:**
6. python3 import\_to\_elasticsearch.py --input /var/log/suricata/fast.log

**4. AI Feature Demonstration**

**4.1 Threat Intelligence Extraction**

* **Uses NLP to process threat feeds** from security blogs, dark web data.

**4.2 Predictive Analytics**

* **Trains on past attack trends** to predict emerging threats.

**4.3 Risk Scoring**

* **AI ranks threats by severity** to prioritize investigations.

**5. SOC Dashboard Configuration**

**5.1 Kibana Enhancements**

* **Create a dashboard** for visualizing:
  + PCAP-imported metadata.
  + AI-generated risk scores.
  + Attack trends over time.

**5.2 Alerting Mechanism**

* **Configure Elasticsearch alerts** for high-risk threats.
* Display AI-driven attack predictions in Kibana.

**6. Testing & Validation**

**6.1 Testing AI Analysis on Sample PCAPs**

1. **Run AI Model on Imported PCAP:**
2. python3 analyze\_pcap.py --input /home/user/pcap\_imports/sample.pcap
3. **Verify Risk Scores & Predictions:**
   * Check logs for AI-generated insights.
   * Cross-check with known CVEs.

**6.2 Demonstration Checklist**

* **Import PCAP into VM.**
* **Analyze data using Zeek & Suricata.**
* **Run AI models for predictions.**
* **Display results in Kibana dashboard.**
* **Showcase risk prioritization based on AI.**

**7. Conclusion**

This **student-friendly VM deployment plan** allows AI-enhanced Security Onion to be used for **PCAP-based cybersecurity research** without requiring extensive resources. The system will showcase AI-powered insights using imported packet capture data.

**8. Next Steps**

1. **Test the system with multiple PCAP datasets.**
2. **Refine AI models for better accuracy.**
3. **Improve dashboard with real-time analytics.**
4. **Prepare a demo walkthrough for presentation.**