Rapport Project Fin d’années sous le sujet

*« Conception d’un Système de Détection d’Attaques Basé sur l’IA pour un Pare-feu Intelligent »*

**1. Introduction**

Dans un environnement où les cyberattaques deviennent de plus en plus sophistiquées, il est impératif d’adopter des solutions de cybersécurité avancées capables d’identifier et de réagir aux menaces en temps réel. Ce projet vise à développer un **système hybride de détection et de réponse aux cyberattaques**, basé sur une combinaison d’outils open-source et d’algorithmes de Machine Learning.

L’objectif est d’intégrer un mécanisme dynamique de **détection des anomalies et des attaques**, capable de générer automatiquement des règles de sécurité appliquées sur un pare-feu intelligent. Ce système sera **déployé sur une machine jouant le rôle de passerelle de sécurité** entre le réseau interne et externe d’une entrepris

2. Architecture du Système

L’architecture du système est conçue pour maximiser la détection et la réactivité face aux menaces en intégrant plusieurs modules spécialisés qui travaillent en synergie. Cette architecture modulaire permet une collecte complète des données, une analyse approfondie des risques, et une application dynamique des politiques de sécurité. Chaque module joue un rôle spécifique dans le cycle de détection et de réponse, assurant une couverture exhaustive des vecteurs d’attaque potentiels.

* **Module de Capture de Trafic** : Ce module est responsable de l’interception du trafic réseau à des points stratégiques, tels que la passerelle réseau ou directement sur le pare-feu de l’entreprise. En utilisant des outils comme tcpdump , ce module collecte des paquets réseau bruts, offrant une visibilité détaillée sur les communications entrantes et sortantes. Ces données brutes sont ensuite stockées temporairement pour être analysées.
* **Module de Prétraitement des Données** : Avant d’être analysées, les données capturées sont filtrées et transformées par ce module. Logstash, un composant clé de la stack ELK, est utilisé pour enrichir les données, en extraire les informations essentielles (telles que les adresses IP, les ports, et les protocoles), et éliminer les éléments superflus. Cette phase garantit que seules les données pertinentes sont transmises aux modèles d’analyse, optimisant ainsi les performances du système.
* **Module d’Analyse et Détection des Menaces :** Ce module utilise des algorithmes de Machine Learning, implémentés en Python avec des bibliothèques comme TensorFlow, pour identifier des patterns suspects dans le trafic réseau et les logs. Il combine des approches supervisées (pour reconnaître des attaques déjà identifiées) et non supervisées (pour détecter des anomalies ou des menaces inconnues). Ce module est essentiel pour transformer des données brutes en informations exploitables sur la sécurité du réseau.
* **Module de Stockage et Visualisation :** Afin d’assurer un suivi efficace des événements de sécurité, ce module stocke toutes les données traitées dans Elasticsearch, une base de données orientée recherche. Kibana est ensuite utilisé pour visualiser ces données à travers des tableaux de bord interactifs, permettant aux administrateurs réseau de surveiller les activités en temps réel, d’identifier rapidement les menaces et de prendre les mesures nécessaires.
* **Module d’Alerte et de Réponse :** Ce module joue un rôle crucial dans la réactivité du système. À l’aide de Watcher dans Elasticsearch, il génère des alertes basées sur des conditions prédéfinies, comme une augmentation anormale du trafic ou la détection de signatures associées à des attaques spécifiques. En complément, des scripts Python peuvent être déclenchés pour automatiser les réponses aux incidents, par exemple en bloquant une adresse IP suspecte ou en isolant une machine compromise.

Cette architecture modulaire permet une scalabilité et une flexibilité accrues, facilitant l’intégration avec d’autres systèmes de sécurité existants et l’adaptation à l’évolution des menaces. Elle offre également la possibilité d’améliorer continuellement les capacités de détection en ajustant les modèles de Machine Learning et les règles de sécurité en fonction des nouvelles données collectées.

**Module 1 :**

**1. Objectif**

Ce module a pour objectif d’assurer la collecte efficace du trafic réseau afin d’identifier les activités suspectes ou malveillantes. Il constitue la première étape du pipeline de détection en capturant des données brutes avant leur traitement et leur analyse dans les modules suivants.

**Les principales fonctions de ce module incluent :**

* Surveillance du trafic en temps réel pour enregistrer les connexions entrantes et sortantes.
* Filtrage des paquets non pertinents afin de limiter le volume de données collectées.
* Génération de logs structurés exploitables par les outils d’analyse et de Machine Learning.
* Transmission des données aux modules de prétraitement et de détection des menaces.

**2. Outils utiliser**

Le choix des outils repose sur leur capacité à capturer, analyser et stocker efficacement le trafic réseau.

**tcpdump (Capture de paquets bruts)**

* Outil en ligne de commande permettant de capturer les paquets réseau sur une interface donnée.
* Filtrage des paquets en fonction de l’adresse IP, du protocole ou du port.
* Sauvegarde des paquets dans des fichiers pcap pour une analyse ultérieure.

**Zeek (Analyse approfondie du trafic réseau)**

* Génération de logs détaillés sur le trafic réseau (connexions, protocoles, activités suspectes).
* Détection d’anomalies basée sur des signatures et des modèles comportementaux.
* Extraction d’informations pertinentes pour leur transmission au module de prétraitement et d’enrichissement des données.

**3. Déploiement et Configuration**

**3.1. Emplacement et Installation**

Ce module est déployé sur un pare-feu, afin de capturer l’ensemble du trafic entrant et sortant. Il est installé sur une machine dédiée, sous un environnement Linux.

**3.2. Capture et Analyse en Temps Réel**

**Une fois installés, les outils sont configurés pour fonctionner en continu et transmettre les données aux modules suivants.**

* tcpdump est utilisé pour capturer les paquets bruts et les enregistrer sous forme de fichiers exploitables.
* Zeek analyse ces fichiers et génère des logs détaillés contenant les informations essentielles à la détection des menaces.

**3.3. Intégration avec le Module 2 (Prétraitement et Enrichissement des Données)**

Les logs générés par Zeek sont collectés par Filebeat, puis transmis à Logstash pour un filtrage avancé. Cette dernière applique des règles de transformation et d’enrichissement avant d’indexer les données dans Elasticsearch, permettant ainsi une recherche et une corrélation efficaces des événements.

**4. Rôle dans le Pipeline de Détection**

Ce module joue un rôle fondamental dans la chaîne de détection en assurant une capture fiable et structurée des communications réseau. Son interaction avec le module de prétraitement garantit un flux de données optimisé, facilitant ainsi l’analyse et l’identification des menaces dans les étapes suivantes.

**Module 2**

**Objectif**

Ce module a pour but de traiter les données collectées afin de les rendre exploitables par les modèles de Machine Learning et les outils d'analyse. Il permet d'assurer la qualité des données en supprimant les informations inutiles, en structurant les logs et en ajoutant des métadonnées essentielles avant leur stockage.

**Outils Recommandés**

L’architecture repose sur plusieurs outils spécialisés, chacun ayant un rôle précis dans le pipeline de traitement des données.

1. **Filebeat**
   * Agent léger permettant la collecte des logs en temps réel depuis différentes sources, notamment des fichiers journaux, des flux réseau et des conteneurs Docker.
   * Transmet les données à Logstash pour transformation ou directement à Elasticsearch pour indexation.
2. **Logstash**
   * Permet le filtrage, l’enrichissement et la transformation des logs collectés.
   * Applique des règles de prétraitement, comme la suppression des entrées non pertinentes et l’ajout de champs supplémentaires (timestamp, adresse IP, protocole).
   * Formate les données en un modèle structuré avant de les envoyer à Elasticsearch.
3. **Elasticsearch**
   * Stocke les logs indexés afin de permettre des recherches rapides et efficaces.
   * Facilite l’analyse et la corrélation des événements grâce à des requêtes avancées.

**Déploiement et Configuration**

Le déploiement de ce module suit plusieurs étapes essentielles pour assurer une ingestion efficace des données :

1. **Déploiement des Conteneurs Docker**
   * Utilisation de Docker Compose pour orchestrer le lancement des services Filebeat, Logstash et Elasticsearch.
   * Configuration des volumes pour persister les données et éviter toute perte d’informations en cas de redémarrage.
2. **Configuration de Filebeat**
   * Définition des sources de logs à surveiller (fichiers journaux des équipements réseau, flux de paquets capturés).
   * Application de filtres pour exclureles entrées redondantes et limiter le volume de données transmises.
3. **Configuration de Logstash**
   * Création de pipelines de transformation des données en fonction des types de logs collectés.
   * Filtrage des événements non pertinents et normalisation des informations essentielles.
   * Ajout de métadonnées pour enrichir les logs et faciliter leur exploitation par les modèles de Machine Learning.
4. **Indexation et Stockage dans Elasticsearch**
   * Définition des schémas d’indexation pour organiser les données et permettre des recherches optimisées.
   * Vérification de la cohérence des logs stockés et ajustement des règles de traitement si nécessaire.

**Module 3 :**

**Objectif :**

Ce module vise à identifier les comportements suspects dans le réseau et les emails à l'aide de modèles de Machine Learning. L'objectif est de détecter les attaques potentielles telles que les DDoS, les tentatives d'intrusion, ainsi que les emails de phishing. Pour ce faire, différentes techniques d'apprentissage automatique seront utilisées, incluant des approches supervisées et non supervisées, adaptées à la nature variée des menaces.

**Attaques viser :**

* Phishing
* DDOS
* Command & Control (C&C) avec Domain Generation Algorithm (DGA)

[ExtraHop/DGA-Detection-Training-Dataset](https://github.com/ExtraHop/DGA-Detection-Training-Dataset)

**Module 4**

**1. Objectif**

Ce module a pour mission d’assurer la visualisation des données issues des analyses de trafic réseau et de la détection des menaces. Il permet de centraliser les logs et d’offrir une interface intuitive pour leur exploitation, sans inclure la gestion des alertes qui sera prise en charge par le module 5.

**Les principales fonctions de ce module incluent :**

* Recherche rapide et efficace des incidents grâce à un moteur d’analyse avancé.
* Création de tableaux de bord interactifs permettant le suivi des menaces en temps réel.
* Analyse approfondie des tendances et des comportements suspects détectés par les modèles de Machine Learning.

**2. Outils**

**Kibana (Visualisation et Tableaux de Bord)**

* Interface graphique permettant la création de visualisations interactives des données de sécurité stockées dans Elasticsearch.
* Génération de tableaux de bord personnalisés pour surveiller les activités suspectes, analyser les tendances et identifier les menaces en temps réel.
* Mise en place de filtres et de graphiques pour mieux comprendre l’évolution des menaces.

**3. Déploiement et Configuration**

**3.1. Intégration avec le Pipeline de Détection**

Ce module s’inscrit dans un processus structuré où il exploite les données collectées et traitées par les modules précédents :

1. **Capture des paquets réseau (Module 1) :** tcpdump et Zeek enregistrent le trafic et génèrent des logs structurés.
2. **Prétraitement et enrichissement des données (Module 2) :** Filebeat collecte les logs, Logstash applique des règles de transformation et ajoute des métadonnées avant d’envoyer les données vers Elasticsearch.
3. **Détection des anomalies et des attaques (Module 3) :** Les modèles de Machine Learning analysent les logs indexés pour identifier les activités suspectes et marquer les événements critiques.
4. **Stockage et visualisation (Module 4) :** Elasticsearch stocke les logs enrichis et permet leur exploitation via Kibana.

**3.2. Configuration des Tableaux de Bord dans Kibana**

* Création de visualisations dynamiques permettant d’identifier rapidement les tendances et les incidents.
* Mise en place de filtres pour analyser les événements par période, type d’attaque ou source du trafic.
* Développement d’indicateurs de performance pour évaluer l’efficacité des mécanismes de détection.

**Module 5**

**1. Objectif**

Ce module a pour but de gérer les alertes et d’automatiser la réponse aux incidents de sécurité détectés par les modèles d’analyse. Il permet de réagir en temps réel aux menaces, en notifiant les administrateurs ou en déclenchant des actions correctives sur les équipements du réseau.

**Les principales fonctionnalités incluent :**

* Surveillance continue des événements de sécurité et détection des incidents critiques.
* Génération d’alertes en fonction des logs et anomalies identifiées dans Elasticsearch.
* Automatisation des réponses aux menaces (ex. blocage d’IP malveillantes, mise en quarantaine d’équipements compromis).

**2. Outils utiliser**

**Gestion des alertes : Watcher (Elasticsearch)**

* Extension native d’Elasticsearch permettant de définir des règles d’alerte en fonction des événements stockés.
* Surveillance des tendances anormales et déclenchement de notifications en cas d’attaque suspecte.
* Intégration avec des systèmes externes pour transmettre les alertes aux administrateurs.

**Automatisation des Réactions : Scripts Python**

* Développement de scripts permettant d’interpréter les alertes et d’initier des actions automatisées.
* Interaction avec les pare-feux et les systèmes de gestion réseau pour appliquer des contre-mesures.

**Application Dynamique des Règles de Pare-feu : Iptables**

* Modification automatique des règles de filtrage en fonction des menaces détectées.
* Blocage en temps réel des adresses IP malveillantes signalées par le système d’alerte.
* Sécurisation proactive du réseau en limitant l’impact des attaques en cours.

**3. Déploiement et Configuration**

**3.1. Configuration des Alertes avec Watcher**

* Définition de règles de surveillance sur les logs de sécurité indexés dans Elasticsearch.
* Déclenchement d’alertes en fonction des seuils critiques définis (ex. taux anormal de connexions depuis une IP unique).
* Intégration avec des systèmes de messagerie pour notifier les administrateurs en temps réel.

**3.2. Automatisation des Réponses avec Python**

* Développement de scripts pour analyser les alertes et générer des réponses automatisées.
* Interaction avec les équipements de sécurité pour prendre des décisions basées sur les incidents détectés.
* Mise en place de mécanismes d’apprentissage permettant d’adapter les réactions en fonction de la nature des attaques.

**3.3. Mise en Place des Règles de Pare-feu Dynamiques**

* Utilisation d’Iptables pour ajuster les règles de filtrage en fonction des alertes.
* Blocage automatique des adresses IP suspectes identifiées par le système d’alerte.
* Configuration d’une liste noire évolutive pour renforcer la sécurité du réseau.

**4. Résultats Attendus**

* **Détection proactive des attaques** grâce à l’apprentissage automatique.
* **Réduction des faux positifs** en affinant les modèles de détection.
* **Automatisation des règles de sécurité** pour bloquer les menaces en temps réel.
* **Visualisation centralisée des événements** pour faciliter l’investigation.