Cahier des Charges : Projet de Fin d'Études (PFE)

Titre du Projet : Détection des Menaces par Correspondance des Règles YARA et SIGMA

1. Contexte et Objectifs du Projet

1.1 Contexte

Dans un monde numérique en constante évolution, la cybersécurité est devenue une priorité absolue pour les organisations. Les menaces informatiques deviennent de plus en plus sophistiquées, nécessitant des outils de détection efficaces et adaptables. Parmi ces outils, les règles YARA et SIGMA se distinguent comme des standards pour l'identification de codes malveillants (malware) et de comportements suspects dans les systèmes.

- **YARA** : Un outil open-source permettant d'identifier et de classer des logiciels malveillants en fonction de motifs spécifiques (signatures).
- **SIGMA** : Un langage générique pour la description de règles de détection de menaces basé sur les logs système.

Ce projet vise à concevoir et développer un système capable de détecter des menaces en temps réel ou quasi-réel en utilisant ces deux technologies.

1.2 Objectifs

Le projet a pour objectifs principaux :

1. **Intégration des Règles YARA et SIGMA** : Développer un système qui combine la puissance de YARA pour analyser des fichiers suspects et SIGMA pour surveiller les logs système.
2. **Automatisation de la Détection** : Créer un pipeline automatisé capable de scanner des fichiers et des logs pour détecter des comportements malveillants.
3. **Performance et Évolutivité** : Optimiser le système pour traiter de grandes quantités de données tout en restant performant.
4. **Interface Utilisateur** : Proposer une interface conviviale pour visualiser les résultats de la détection et gérer les alertes.
5. **Documentation et Tests** : Fournir une documentation complète et effectuer des tests rigoureux pour valider l'efficacité du système.

2. Description Fonctionnelle

2.1 Fonctionnalités Principales

1. **Analyse des Fichiers avec YARA** :
    - Scanner des fichiers pour détecter des signatures de malware définies dans des règles YARA.
    - Générer des alertes en cas de correspondance avec des règles spécifiques.
2. **Surveillance des Logs avec SIGMA** :
    - Analyser les logs système (par exemple, logs Windows, Linux, ou application) pour détecter des comportements suspects décrits dans des règles SIGMA.
    - Convertir les règles SIGMA en formats compatibles avec les SIEM (Security Information and Event Management) courants.
3. **Corrélation des Données** :
    - Corréler les résultats des analyses YARA et SIGMA pour identifier des menaces complexes impliquant plusieurs vecteurs.
4. **Gestion des Règles** :
    - Permettre l'ajout, la modification et la suppression de règles YARA et SIGMA via une interface utilisateur.
5. **Notifications et Rapports** :
    - Envoyer des notifications en temps réel en cas de détection de menaces.
    - Générer des rapports détaillés sur les incidents détectés.

2.2 Architecture Proposée

1. **Module de Collecte de Données** :
    - Collecte des fichiers à analyser.
    - Collecte des logs système depuis différentes sources.
2. **Module de Traitement** :
    - Application des règles YARA aux fichiers.
    - Application des règles SIGMA aux logs.
3. **Module de Corrélation** :
    - Analyse croisée des résultats pour identifier des menaces complexes.
4. **Module de Notification** :
    - Envoi d'alertes par email, SMS ou via une interface web.
5. **Interface Utilisateur** :
    - Tableau de bord pour visualiser les alertes, gérer les règles et consulter les rapports.

3. Contraintes Techniques

3.1 Langages et Outils

- **Langages de Programmation** : Python (pour l'intégration de YARA et SIGMA), JavaScript/HTML/CSS (pour l'interface utilisateur).
- **Bibliothèques/Frameworks** :
    - YARA : yara-python
    - SIGMA : sigma-tools, pySigma
    - Interface Web : Flask/Django (backend), React/Vue.js (frontend).

- **Base de Données** : PostgreSQL ou MongoDB pour stocker les résultats de détection et les règles.
- **Outils de Surveillance** : Elastic Stack (ELK) pour la gestion des logs.

3.2 Plateformes Supportées

- Systèmes d'exploitation : Linux, Windows.
- Compatibilité avec des SIEM existants (Splunk, QRadar, etc.).

3.3 Performance

- Temps de réponse optimisé pour l'analyse des fichiers et des logs.
- Capacité à traiter des volumes importants de données sans dégradation des performances.

3.4 Sécurité

- Chiffrement des communications entre les modules.
- Authentification et autorisation pour accéder à l'interface utilisateur.

4. Livrables Attendus

1. **Documentation** :
    - Cahier des charges détaillé.
    - Documentation technique du système développé.
    - Guide d'utilisation pour les administrateurs et utilisateurs finaux.
2. **Code Source** :
    - Code complet et commenté, accompagné d'une structure claire.
3. **Interface Utilisateur** :
    - Une interface web responsive pour la gestion des règles et la visualisation des alertes.
4. **Rapport Final** :
    - Analyse des résultats obtenus lors des tests.
    - Propositions d'améliorations futures.

5. Planning Prévisionnel

Recherche et analyse	2 Semaines
Conception de l'architecture	2 Semaines
Développement des modules	6 Semaines
Intégration et tests	3 Semaines
Documentation et rapport final	2 Semaines

6. Critères de Réussite

1. **Fonctionnalité** : Le système doit être capable de détecter efficacement les menaces en utilisant les règles YARA et SIGMA.

2. **Performance** : Le système doit traiter des volumes importants de données sans latence significative.
3. **Utilisabilité** : L'interface utilisateur doit être intuitive et facile à utiliser.
4. **Scalabilité** : Le système doit être conçu pour s'adapter à des environnements de grande envergure.
5. **Sécurité** : Les données doivent être protégées contre tout accès non autorisé.

7. Conclusion

Ce projet vise à combiner deux outils puissants de détection de menaces (YARA et SIGMA) pour créer un système robuste et évolutif. Il s'agit d'une opportunité unique d'explorer les aspects techniques de la cybersécurité tout en répondant à des besoins réels des entreprises.

8. Annexes

- **Références** :
  - Documentation officielle de YARA : https://yara.readthedocs.io
  - Documentation officielle de SIGMA : https://github.com/SigmaHQ/sigma