**Cahier des Charges Détaillé**

**Projet : Système de Surveillance et d’Analyse de Logs pour la Cybersécurité**

**1. Introduction**

**1.1 Contexte du projet**

Dans un contexte où les attaques informatiques deviennent de plus en plus sophistiquées, la surveillance continue des systèmes informatiques est devenue essentielle. Les entreprises doivent analyser d’énormes volumes de **logs** générés par leurs serveurs, pare-feu, applications et équipements réseau afin de détecter des comportements suspects ou des tentatives d’intrusion.

Ce projet vise à concevoir et mettre en place un **Système de Surveillance et d’Analyse de Logs** basé sur des **technologies Big Data** (Kafka, Spark, Cassandra, Elasticsearch, Power BI) afin de garantir une **collecte, un traitement et une visualisation en temps réel** des événements liés à la sécurité.

**2. Objectifs du projet**

* Collecter en temps réel les logs provenant de différentes sources (serveurs, pare-feu, routeurs, applications).
* Traiter et analyser ces logs massifs pour détecter des anomalies ou des activités suspectes.
* Stocker efficacement les données dans une base adaptée au Big Data.
* Permettre la **visualisation et l’exploration des données de sécurité** via des tableaux de bord dynamiques dans **Power BI**.
* Fournir des **alertes en temps réel** en cas de détection de comportements anormaux.

**3. Portée du système**

Le système couvre :

* **La collecte** de logs en continu depuis plusieurs sources.
* **Le traitement et la normalisation** des données via un pipeline Big Data.
* **Le stockage** distribué et optimisé pour la lecture/écriture intensive.
* **L’analyse et la détection d’anomalies** à l’aide d’algorithmes de traitement de flux.
* **La visualisation et le reporting** des indicateurs de cybersécurité.

**4. Architecture Globale du Système**

**4.1 Composants techniques**

| **Composant** | **Rôle** |
| --- | --- |
| **Apache Kafka** | Collecte et transmission en temps réel des logs sous forme de flux distribués. |
| **Apache Spark (Streaming)** | Traitement en temps réel des données issues de Kafka (filtrage, agrégation, détection d’anomalies). |
| **Apache Cassandra** | Base de données distribuée pour le stockage des logs traités et agrégés. |
| **Elasticsearch** | Indexation et recherche rapide des événements de sécurité. |
| **Power BI** | Visualisation interactive des indicateurs et alertes de sécurité via tableaux de bord. |

**4.2 Schéma d’architecture (description)**

1. Les différentes sources (serveurs, routeurs, firewalls, applications) envoient leurs logs vers **Apache Kafka**.
2. **Spark Streaming** consomme les messages Kafka, nettoie et enrichit les logs, puis les envoie vers **Cassandra** pour stockage permanent.
3. En parallèle, **Elasticsearch** indexe les données pour permettre des recherches rapides.
4. **Power BI** se connecte à **Elasticsearch** ou **Cassandra** pour afficher des rapports et tableaux de bord interactifs.

**5. Spécifications Fonctionnelles**

**5.1 Module de Collecte (Kafka)**

* Collecter les logs de diverses sources (HTTP, syslog, fichiers, API, etc.).
* Assurer une transmission fiable et tolérante aux pannes.
* Regrouper les messages en topics selon le type de source ou de gravité.

**5.2 Module de Traitement (Spark)**

* Nettoyer et transformer les données en temps réel.
* Détecter des anomalies (ex. : nombre élevé d’échecs de connexion).
* Calculer des indicateurs clés :
  + Taux d’erreurs par application
  + Volume de logs par heure
  + Activité utilisateur suspecte

**5.3 Module de Stockage (Cassandra)**

* Stocker les données normalisées sous forme de tables partitionnées.
* Garantir une haute disponibilité et une scalabilité horizontale.
* Conserver l’historique des logs sur une longue période.

**5.4 Module de Recherche (Elasticsearch)**

* Indexer les données pour permettre des recherches multicritères.
* Fournir un accès rapide aux logs filtrés par date, source ou gravité.

**5.5 Module de Visualisation (Power BI)**

* Tableaux de bord dynamiques avec indicateurs clés :
  + **Nombre d’attaques détectées**
  + **Top 10 des adresses IP suspectes**
  + **Taux de réussite/échec d’authentification**
  + **Évolution temporelle des alertes**
* Possibilité de filtrer par source, période, ou type d’événement.

**6. Spécifications Non Fonctionnelles**

| **Critère** | **Exigence** |
| --- | --- |
| **Performance** | Traitement de milliers de logs par seconde. |
| **Scalabilité** | Ajout de nouveaux nœuds Kafka, Spark ou Cassandra sans interruption. |
| **Disponibilité** | Tolérance aux pannes via réplication et partitionnement. |
| **Sécurité** | Authentification, chiffrement TLS, contrôle d’accès. |
| **Intégration** | Connectivité avec Power BI et autres outils SIEM. |

**7. Cas d’utilisation principaux**

| **Acteur** | **Cas d’utilisation** | **Description** |
| --- | --- | --- |
| **Administrateur Sécurité** | Visualiser le tableau de bord | Consulter les indicateurs en temps réel sur Power BI. |
| **Analyste SOC** | Rechercher un log spécifique | Utiliser Elasticsearch pour filtrer par adresse IP, date ou type d’événement. |
| **Système** | Générer une alerte | Envoyer une notification si un seuil d’anomalies est dépassé. |

**8. Technologies et Outils**

* **Langages :** Python, Scala
* **Frameworks :** Apache Spark Streaming
* **Bases de données :** Cassandra, Elasticsearch
* **Messagerie :** Apache Kafka
* **Visualisation :** Power BI
* **Conteneurisation (optionnel) :** Docker
* **Orchestration (optionnel) :** Kubernetes

**9. Livrables**

1. Rapport d’analyse et de conception (diagrammes UML, architecture).
2. Code source du pipeline Big Data.
3. Base de données Cassandra et index Elasticsearch fonctionnels.
4. Tableaux de bord Power BI prêts à l’emploi.
5. Documentation technique et guide d’installation.

**10. Planification Prévisionnelle (Exemple)**

| **Phase** | **Durée** | **Livrables** |
| --- | --- | --- |
| Étude et conception | 2 semaines | Diagrammes, architecture fonctionnelle |
| Déploiement de l’infrastructure Big Data | 3 semaines | Cluster Kafka/Spark/Cassandra |
| Développement du pipeline de traitement | 3 semaines | Scripts Spark Streaming |
| Intégration d’Elasticsearch et Power BI | 2 semaines | Tableaux de bord |
| Tests et validation finale | 1 semaine | Rapport de tests |

**11. Conclusion**

Ce système permettra d’assurer une **supervision proactive de la sécurité** à travers l’analyse en temps réel des logs, tout en offrant une **architecture scalable, résiliente et visuellement exploitable**. Grâce à l’intégration de **Power BI** et des technologies Big Data (Kafka, Spark, Cassandra, Elasticsearch), le système offrira une vision complète et centralisée de l’état de la cybersécurité d’une infrastructure.