Cahier des Charges Détaillé

Projet : Système Big Data de Surveillance et d'Analyse de Logs Réseau pour la Cybersécurité

1. Introduction

1.1 Contexte

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.

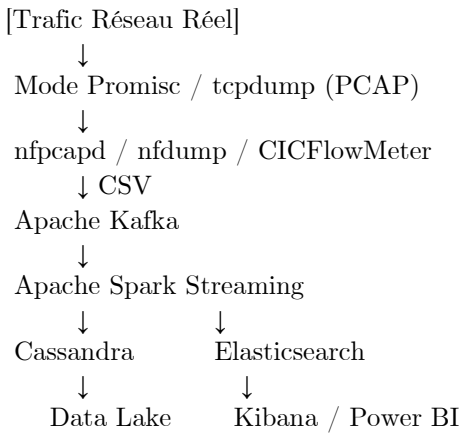
Ce projet vise à concevoir et mettre en place un Système de Surveillance et d'Analyse de Logs basé sur des technologies Big Data (Kafka, Spark, Cassandra, Elasticsearch, Power BI) afin de garantir une collecte, un traitement et une visualisation en temps réel des événements liés à la sécurité.

1.2 Objectifs

- Capturer le trafic réseau brut via le mode promiscuous.
- Convertir les paquets capturés (PCAP) en données de flux analytiques (NetFlow/CSV).
- Intégrer les données collectées dans un pipeline Big Data (Kafka, Spark, Cassandra, Elasticsearch).
- Appliquer des traitements analytiques et de Machine Learning pour détecter les comportements anormaux.
- Visualiser les résultats et alertes via Power BI.

2. Architecture Globale du Système

2.1 Vue d'ensemble



2.2 Composants Principaux

Composant	Rôle
Interface Promisc	Capture des paquets réseau circulant sur le réseau.
tcpdump	Enregistrement continu des paquets au format .pcap.
nfpcapd / nfdump / CICFlowMeter	Conversion des fichiers PCAP en flux NetFlow et extraction des statistiques CSV.
Apache Kafka	Transmission asynchrone et scalable des flux de logs vers Spark.
Apache Spark	Traitement temps réel et batch, analyse statistique et apprentissage automatique.
Apache Cassandra	Stockage distribué haute disponibilité des flux traités.
Elasticsearch + Kibana	Indexation rapide et visualisation en temps réel.
Power BI	Tableaux de bord analytiques et rapports décisionnels.

2.3 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.

3. Environnement Technique

Élément	Technologie
Capture	tcpdump, nfpcapd, nfdump, CICFlowMeter
Traitement	Apache Spark, PySpark, Kafka

4. Spécifications Fonctionnelles

4.1 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.

4.2 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

4.3 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.

4.4 Module de Recherche (Elasticsearch)

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

4.5 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.

5. 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.

6. Étapes Techniques Détaillées

6.1 Capture du trafic réseau

Activation du mode promiscuité :

```
sudo ifconfig eth0 promisc
```

Mode monitor pour carte Wi-Fi :

```
sudo ip link set wlan0 down
sudo iw wlan0 set monitor control
sudo ip link set wlan0 up
```

Capture et rotation automatique :

```
sudo tcpdump -i eth0 -s 0 -W 10 -C 100 -w /var/log/pcap/capture-%Y%m%d%H%M.pcap
```

6.2 Conversion PCAP → NetFlow / CSV

Extraction via nfpcapd :

```
nfpcapd -r /tmp/capture.pcap -l /var/log/netflow/
```

Export CSV :

nfdump -r /var/log/netflow/nfcapd.YYYYMMDDHHMM -o csv > /tmp/flows.csv

Extraction des features avancées :

java -jar CICFlowMeter.jar -r /tmp/capture.pcap -f /tmp/flows.csv

6.3 Ingestion via Apache Kafka

Chaque CSV est lu et publié sur un topic Kafka (netflow-data) :

```
from kafka import KafkaProducer
import pandas as pd, json

producer = KafkaProducer(bootstrap_servers=['localhost:9092'])
df = pd.read_csv('/tmp/flows.csv')

for _, row in df.iterrows():
    producer.send('netflow-data', value=row.to_json().encode('utf-8'))
```

6.4 Traitement en Temps Réel avec Apache Spark

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import *

spark = SparkSession.builder \
    .appName("NetFlowPipeline") \
    .config("spark.cassandra.connection.host", "127.0.0.1") \
    .getOrCreate()

df = spark.readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "localhost:9092") \
    .option("subscribe", "netflow-data") \
    .load()

cleaned = df.selectExpr("CAST(value AS STRING)")
query = cleaned.writeStream \
    .format("org.apache.spark.sql.cassandra") \
    .options(table="flows", keyspace="netflow") \
    .outputMode("append") \
    .start()

query.awaitTermination()
```

6.5 Stockage et Indexation

Cassandra

Structure de table :

```
CREATE TABLE netflow.flows (
    id UUID PRIMARY KEY,
    src_ip text,
    dest_ip text,
    src_port int,
    dest_port int,
    protocol int,
    bytes_in bigint,
    bytes_out bigint,
    num_pkts_in bigint,
    num_pkts_out bigint,
    start_time timestamp,
    end_time timestamp
);
```

Elasticsearch

Connexion via Spark :

```
df.writeStream \
    .format("es") \
    .option("es.nodes", "localhost") \
    .option("es.resource", "netflow/_doc") \
    .outputMode("append") \
    .start()
```

6.6 Visualisation avec Power BI

- Connexion directe à Elasticsearch via le connecteur officiel.
- Dashboards dynamiques :
    - 🔹 Volume de trafic par protocole
    - 🔹 Top IP sources/destinations
    - 🔹 Activité anormale détectée par Spark ML
    - 🔹 Statistiques temporelles (trafic horaire, pics, incidents)

7. Spécifications Fonctionnelles

Module	Fonctionnalités clés
Capture	Mode promisc, rotation PCAP, export NetFlow
Kafka	Collecte des logs en streaming
Spark	Nettoyage, transformation, ML
Cassandra	Stockage distribué des données de flux
Elasticsearch	Indexation et recherche
Power BI	Visualisation et alertes décisionnelles

8. Spécifications Non Fonctionnelles

Critère	Exigence
Performance	≥ 10 000 paquets/s
Scalabilité	Cluster Kafka/Spark/Cassandra extensible
Sécurité	Authentification, TLS, RBAC Power BI
Disponibilité	24/7 avec reprise automatique
Interopérabilité	Formats PCAP, NetFlow, CSV, JSON
Monitoring	Logs centralisés, alertes automatiques

9. Cas d'Utilisation

Acteur	Cas d'utilisation	Description
Administrateur Réseau	Lancer capture réseau	Activation manuelle de tcpdump ou script automatique
Système	Transformation PCAP → CSV	Conversion périodique via nfpcapd

Acteur	Cas d'utilisation	Description
Pipeline Kafka/Spark	Ingestion & traitement temps réel	Nettoyage et analyse de flux réseau
Analyste SOC	Visualiser et filtrer alertes	Power BI et Kibana
Moteur ML	Détection d'anomalies	Algorithmes Spark MLlib

10. Livrables

1. Scripts Bash pour capture et conversion PCAP/NetFlow.
2. Pipeline Kafka & Spark Streaming complet.
3. Base de données Cassandra structurée.
4. Index Elasticsearch configuré.
5. Dashboard Power BI et Kibana.
6. Documentation technique et guide d'installation.

11. Planification Prévisionnelle

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

12. Conclusion

Ce système combine l'ingénierie réseau (tcpdump, nfpcapd, NetFlow) avec la puissance du Big Data (Kafka, Spark, Cassandra, Elasticsearch).
L'ajout de Power BI permet une visualisation décisionnelle complète, facilitant la détection proactive des anomalies réseau et la surveillance continue de la cybersécurité.