

Lab 03

Cybersecurity 3

Elastic Stack

Conteneur Docker Compose

Créez le fichier docker-compose.yml :

```
version: "3.8"
```

```
services:
```

```
  elasticsearch:
```

```
    image: docker.elastic.co/elasticsearch/elasticsearch:8.9.0
```

```
    environment:
```

- discovery.type=single-node
- xpack.security.enabled=false
- xpack.monitoring.enabled=false

```
    ports:
```

- "9200:9200"

```
    healthcheck:
```

```
      test: ["CMD-SHELL", "curl -s http://localhost:9200/_cluster/health | grep green"]
```

```
      interval: 30s
```

```
      retries: 20
```

```
  kibana:
```

```
    image: docker.elastic.co/kibana/kibana:8.9.0
```

```
    ports:
```

- "5601:5601"

```
    depends_on:
```

- elasticsearch

filebeat:

image: docker.elastic.co/beats/filebeat:8.9.0

volumes:

- ./data/webserver.log:/usr/share/filebeat/data/webserver.log:ro
- ./data/powershell_eventlog.csv:/usr/share/filebeat/data/powershell_eventlog.csv:ro
- ./filebeat.yml:/usr/share/filebeat/filebeat.yml:ro

depends_on:

- elasticsearch

Etape 1. Préparation des fichiers de logs

Dans ./data/, placez :

- webserver.log (logs HTTP incluant SQLi et brute force)
- powershell_eventlog.csv (événements Windows avec commandes suspectes)

Créez aussi filebeat.yml :

yaml

CopierModifier

filebeat.inputs:

- type: log

enabled: true

paths:

- /usr/share/filebeat/data/webserver.log

fields:

type: web

- type: log

enabled: true

paths:

- /usr/share/filebeat/data/powershell_eventlog.csv

fields:

type: powershell

output.elasticsearch:

hosts: ["http://elasticsearch:9200"]

Etape 2. Démarrage de la stack

`docker-compose up -d`

Surveillez les logs pour vérifier qu'Elasticsearch et Kibana sont prêts :

`docker-compose logs -f elasticsearch`

`docker-compose logs -f kibana`

`docker-compose logs -f filebeat`

Etape 3. Ingestion des logs

- filebeat lit automatiquement les deux fichiers et indexe dans Elasticsearch.
- Vérifiez dans Kibana → "Discover" que les index filebeat-* contiennent bien ces logs.

Etape 4. Requêtes de détection

Dans Kibana → "Discover" ou "Search" :

1. SQL Injection

`type:"web" AND (url:* OR query:*) AND message:"UNION SELECT"`

2. PowerShell suspect

`type:"powershell" AND (message:*"New-Object Net.WebClient"* OR message:*"Invoke-Expression"*)`

3. Brute force (401 fail)

`type:"web" AND status:401`

Puis transformez en visualisation.

Etape 5. Création du Dashboard

Dans Kibana → Visualize / Dashboard :

- Graphique temporel par heure : compte des événements SQLi / PowerShell / 401s
- Top IPs malveillantes : agrégations sur champ client.ip

Etape 6. Alertes automatisées

Avec Alerts and Actions dans Kibana :

- Créez une alerte :
 - Type : Threshold
 - Index : filebeat-*
 - Expression : nombre d'événements >10 dans dernière minute, par client.ip ou global

Configurez une notification (email, webhook, ou log).

Etape Livrables attendus (document PDF)

1. 🖼️ Capture écran du Dashboard (graph + top IP)
2. 📄 Export des requêtes KQL utilisées
3. 📋 Rapport décrivant :
 - Méthodologie d'ingestion
 - Analyse résultat (ex : X requêtes SQLi à HH:MM)
 - Corrélation entre logs (ex : brute force suivi d'exécution PowerShell)
 - Règle d'alerte proposée