



**SYSTÈME DE DÉTECTION D'ANOMALIES ET DE GESTION DE LOGS POUR LA  
SÉCURITÉ DES RÉSEAUX**

**PAR LARZUL JULIEN**

**RAPPORT DE RECHERCHE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À  
CHICOUTIMI COMME EXIGENCE PARTIELLE EN VUE DE L'OBTENTION DU  
GRADE DE MAÎTRE ÈS SCIENCES EN INFORMATIQUE**

**QUÉBEC, CANADA**

**© LARZUL JULIEN, 2025**

## TABLE DES MATIÈRES

<b>LISTE DES FIGURES</b> . . . . .	1
<b>INTRODUCTION</b> . . . . .	2
<b>CHAPITRE I – MISE EN PLACE DE L’ENVIRONNEMENT</b> . . . . .	3
1.1 ENVIRONNEMENT DE TRAVAIL . . . . .	3
1.2 BRIQUES LOGICIELLES NÉCESSAIRES . . . . .	3
1.3 INSTALLATION DES COMPOSANTS . . . . .	4
1.4 AUTOMATISATION PAR ALIAS . . . . .	5
<b>CHAPITRE II – CONFIGURATION ET INTÉGRATION</b> . . . . .	6
2.1 CONFIGURATION DE SURICATA . . . . .	6
2.1.1 VÉRIFICATION DU CHEMIN DES RÈGLES . . . . .	6
2.1.2 AJOUT D’UNE RÈGLE LOCALE . . . . .	7
2.1.3 TEST DE CONFIGURATION . . . . .	7
2.1.4 RELANCE DU SERVICE . . . . .	7
2.1.5 GÉNÉRATION DE TRAFIC ICMP . . . . .	8
2.1.6 VÉRIFICATION DES ALERTES GÉNÉRÉES . . . . .	9
2.2 CONFIGURATION DE SYSLOG-NG . . . . .	9
2.2.1 FICHIER DE CONFIGURATION . . . . .	10
2.2.2 DÉMARRAGE ET VÉRIFICATION DU SERVICE . . . . .	10
2.2.3 COPIE LOCALE DES ÉVÉNEMENTS . . . . .	11
2.2.4 VÉRIFICATION CÔTÉ ELASTICSEARCH ET KIBANA . . . . .	11
<b>CHAPITRE III – SCÉNARIOS D’INTRUSION (5 CAS)</b> . . . . .	13
<b>CHAPITRE IV – VISUALISATION ET ALERTES</b> . . . . .	14
<b>CHAPITRE V – ANALYSE ET DISCUSSION</b> . . . . .	15
<b>CONCLUSION</b> . . . . .	16
<b>APPENDICE A – PREMIÈRE ANNEXE</b> . . . . .	17

## LISTE DES FIGURES

FIGURE 2.1 – VÉRIFICATION DU CHEMIN DES RÈGLES ET INCLUSION DE LOCAL.RULES.. . . . .	6
FIGURE 2.2 – RÈGLE ICMP DE TEST AJOUTÉE AU FICHIER LOCAL.RULES. . . .	7
FIGURE 2.3 – VALIDATION DE LA CONFIGURATION DE SURICATA.. . . . .	7
FIGURE 2.4 – RELANCE DE SURICATA EN MODE DÉMON SUR L'INTERFACE ENS160. . . . .	8
FIGURE 2.5 – GÉNÉRATION DE TRAFIC ICMP AVEC LA COMMANDE PING. . . .	8
FIGURE 2.6 – ALERTE GÉNÉRÉE DANS FAST.LOG SUITE AU PING ICMP. . . . .	9
FIGURE 2.7 – FICHIER DE CONFIGURATION SYSLOG-NG POUR L'INTÉGRATION DES ÉVÉNEMENTS SURICATA. . . . .	10
FIGURE 2.8 – VÉRIFICATION DU BON FONCTIONNEMENT DU SERVICE SYSLOG-NG. . . . .	10
FIGURE 2.9 – COPIE LOCALE DES ÉVÉNEMENTS SURICATA DANS SURICATA-COLLECTED.JSON.	11
FIGURE 2.10 – ÉVÉNEMENTS SURICATA VISIBLES DANS KIBANA APRÈS INTÉGRATION AVEC SYSLOG-NG ET ELASTICSEARCH. . . . .	11

## INTRODUCTION

Dans ce contexte, ce projet a pour objectif la conception et le déploiement d'un *système de détection d'anomalies et de gestion de logs*. L'approche consiste à mettre en place une chaîne complète allant de la collecte des journaux jusqu'à leur visualisation et leur analyse via une interface conviviale. L'architecture retenue repose sur quatre briques logicielles complémentaires :

- **Suricata**, un système de détection d'intrusions (IDS/IPS) chargé d'analyser le trafic réseau et de générer des alertes en temps réel ;
- **syslog-ng**, utilisé pour centraliser les journaux du système et des applications ;
- **Elasticsearch**, base de données NoSQL permettant l'indexation et la recherche rapide des événements collectés ;
- **Kibana**, une interface web offrant des fonctionnalités de visualisation et de création de tableaux de bord.

Le projet doit également inclure l'implémentation de plusieurs scénarios d'attaque simulés. Ces cas d'intrusion permettront de valider la capacité du système à détecter différents comportements malveillants et à générer des alertes exploitables par l'administrateur.

Ce rapport présente dans un premier temps l'environnement mis en place et les choix technologiques retenus. Il détaille ensuite la configuration et l'intégration des outils, avant de décrire et d'analyser cinq scénarios d'intrusion représentatifs. Enfin, une partie est consacrée à la visualisation des résultats, à l'évaluation des limites du système et aux perspectives d'amélioration.

# CHAPITRE I

## MISE EN PLACE DE L'ENVIRONNEMENT

### 1.1 ENVIRONNEMENT DE TRAVAIL

Le projet a été réalisé par une équipe de quatre personnes. Chaque membre a travaillé sur une machine hôte différente (Windows ou macOS), mais l'ensemble du projet a été uniformisé à travers l'utilisation d'une machine virtuelle Ubuntu. Ce choix garantit un environnement cohérent, reproductible et isolé, permettant de tester des scénarios d'attaques sans risque pour les machines personnelles.

L'environnement retenu est le suivant :

- **Système d'exploitation invité** : Ubuntu 22.04 LTS (Linux)
- **Hyperviseurs utilisés** : VMware Fusion (macOS) et VMware Workstation/VirtualBox (Windows)
- **Ressources allouées** : 2 vCPU, 4 Go de mémoire vive, 40 Go de disque
- **Interface réseau** : ens160, configurée en mode NAT

Ce choix d'architecture permet de travailler de manière collaborative tout en garantissant que les configurations, les scripts et les fichiers produits sont compatibles sur toutes les machines de l'équipe.

### 1.2 BRIQUES LOGICIELLES NÉCESSAIRES

Le projet repose sur quatre composants principaux :

- **Suricata** : un système de détection et de prévention d'intrusions (IDS/IPS), chargé d'analyser le trafic réseau et de générer des alertes.

- **syslog-ng** : un collecteur de logs, utilisé pour centraliser les journaux générés par le système et par Suricata.
- **Elasticsearch** : une base de données orientée recherche, permettant d'indexer et de stocker les logs collectés.
- **Kibana** : une interface web de visualisation connectée à Elasticsearch, permettant d'explorer et d'analyser les logs.

### 1.3 INSTALLATION DES COMPOSANTS

L'installation a été réalisée en ligne de commande sur Ubuntu. Les principales étapes sont résumées ci-dessous.

#### **SURICATA**

```
sudo apt install -y suricata suricata-update  
sudo suricata-update  
sudo systemctl restart suricata
```

La commande `suricata -build-info` permet de vérifier que l'installation est correcte.

#### **SYSLOG-NG**

```
sudo apt install -y syslog-ng  
systemctl status syslog-ng
```

#### **ELASTICSEARCH**

Téléchargement et extraction de la version ARM64 :

```
curl -LO https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch
↳ -8.15.3-linux-aarch64.tar.gz
tar -xzf elasticsearch-8.15.3-linux-aarch64.tar.gz
cd elasticsearch-8.15.3
./bin/elasticsearch -E discovery.type=single-node -E xpack.security.enabled=
↳ false
```

## **KIBANA**

Téléchargement et extraction :

```
curl -LO https://artifacts.elastic.co/downloads/kibana/kibana-8.15.3-linux-
↳ aarch64.tar.gz
tar -xzf kibana-8.15.3-linux-aarch64.tar.gz
cd kibana-8.15.3
./bin/kibana
```

## **1.4 AUTOMATISATION PAR ALIAS**

Afin de simplifier le lancement des différents composants, des alias ont été définis dans le fichier `/.bashrc`. Ces raccourcis permettent à l'équipe de démarrer ou d'arrêter les services (Elasticsearch, Kibana, Suricata) avec des commandes simples, plutôt que de retaper à chaque fois des lignes longues et complexes. Cette approche améliore la lisibilité, réduit les erreurs de saisie et accélère les tests.

## CHAPITRE II

### CONFIGURATION ET INTÉGRATION

#### 2.1 CONFIGURATION DE SURICATA

##### 2.1.1 VÉRIFICATION DU CHEMIN DES RÈGLES

Pour commencer, nous avons vérifié que le fichier `/etc/suricata/suricata.yaml` pointe bien vers le répertoire `/var/lib/suricata/rules`, et que le fichier `local.rules` est inclus dans la section `rule-files` :

```
$ grep -n "default-rule-path" /etc/suricata/suricata.yaml
```

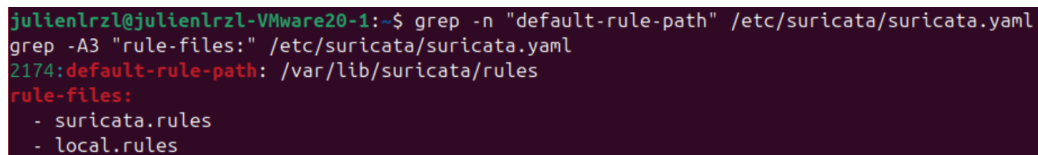
```
2174:default-rule-path: /var/lib/suricata/rules
```

```
$ grep -A3 "rule-files:" /etc/suricata/suricata.yaml
```

```
2176:rule-files:
```

```
2177:  - suricata.rules
```

```
2178:  - local.rules
```



```
julienlrzl@julienlrzl-VMware20-1:~$ grep -n "default-rule-path" /etc/suricata/suricata.yaml
grep -A3 "rule-files:" /etc/suricata/suricata.yaml
2174:default-rule-path: /var/lib/suricata/rules
rule-files:
- suricata.rules
- local.rules
```

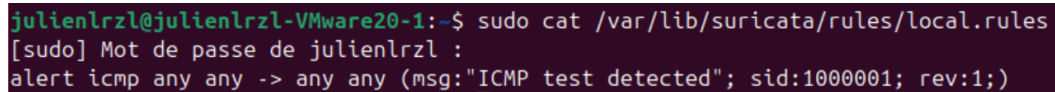
**FIGURE 2.1 : Vérification du chemin des règles et inclusion de `local.rules`.**



### 2.1.2 AJOUT D'UNE RÈGLE LOCALE

Une règle de détection ICMP a été ajoutée dans le fichier `local.rules` :

```
alert icmp any any -> any any (msg:"ICMP test detected"; sid:1000001; rev:1;)
```



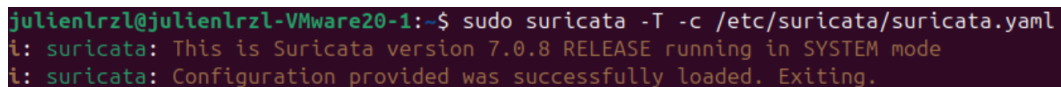
```
julienlrzl@julienlrzl-VMware20-1:~$ sudo cat /var/lib/suricata/rules/local.rules
[sudo] Mot de passe de julienlrzl :
alert icmp any any -> any any (msg:"ICMP test detected"; sid:1000001; rev:1;)
```

**FIGURE 2.2 : Règle ICMP de test ajoutée au fichier `local.rules`.**

### 2.1.3 TEST DE CONFIGURATION

Un test de configuration a permis de vérifier que le fichier YAML est valide et que les règles sont correctement chargées :

```
$ sudo suricata -T -c /etc/suricata/suricata.yaml
-- Configuration OK --
```



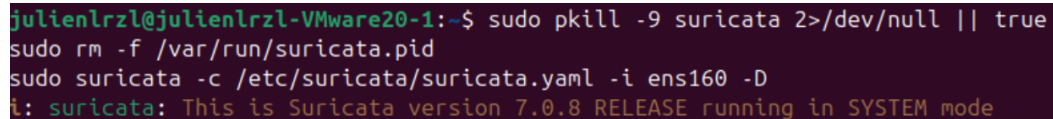
```
julienlrzl@julienlrzl-VMware20-1:~$ sudo suricata -T -c /etc/suricata/suricata.yaml
i: suricata: This is Suricata version 7.0.8 RELEASE running in SYSTEM mode
i: suricata: Configuration provided was successfully loaded. Exiting.
```

**FIGURE 2.3 : Validation de la configuration de Suricata.**

### 2.1.4 RELANCE DU SERVICE

Suricata a ensuite été relancé en mode démon :

```
$ sudo pkill -9 suricata 2>/dev/null || true  
$ sudo rm -f /var/run/suricata.pid  
$ sudo suricata -c /etc/suricata/suricata.yaml -i ens160 -D
```



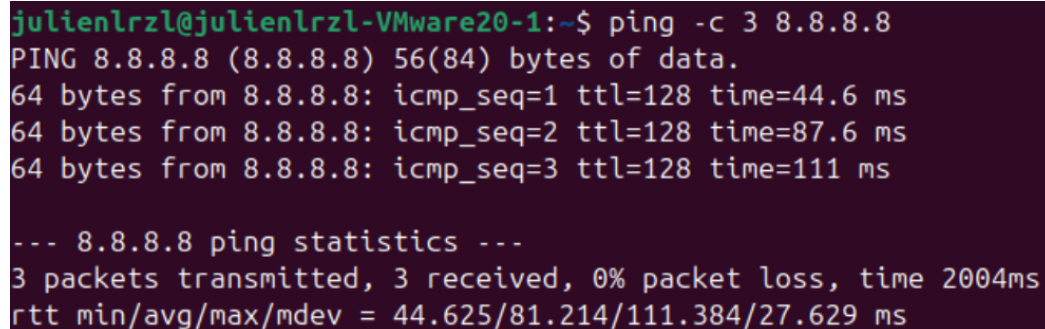
```
julienlrzl@julienlrzl-VMware20-1:~$ sudo pkill -9 suricata 2>/dev/null || true  
sudo rm -f /var/run/suricata.pid  
sudo suricata -c /etc/suricata/suricata.yaml -i ens160 -D  
i: suricata: This is Suricata version 7.0.8 RELEASE running in SYSTEM mode
```

**FIGURE 2.4 : Relance de Suricata en mode démon sur l'interface ens160.**

### 2.1.5 GÉNÉRATION DE TRAFIC ICMP

Un simple ping vers l'adresse publique de Google (8.8.8.8) a été utilisé pour générer du trafic ICMP :

```
$ ping -c 3 8.8.8.8
```



```
julienlrzl@julienlrzl-VMware20-1:~$ ping -c 3 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=44.6 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=87.6 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=111 ms  
  
--- 8.8.8.8 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2004ms  
rtt min/avg/max/mdev = 44.625/81.214/111.384/27.629 ms
```

**FIGURE 2.5 : Génération de trafic ICMP avec la commande ping.**

## 2.1.6 VÉRIFICATION DES ALERTES GÉNÉRÉES

L'analyse du fichier `/var/log/suricata/fast.log` montre bien que la règle locale a déclenché une alerte pour les paquets ICMP observés :

09/25/2025-19:16:40.523467 **[\*\*]** [1:1000001:1] ICMP test detected **[\*\*]** {ICMP}

↪ 172.16.150.130:8 -> 8.8.8.8:0

```
julienlrzi@julienlrzi-Vmware20-1: $ tail -f /var/log/suricata/fast.log
09/25/2025-19:58:25.005565 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (UDP) 172.16.150.130:59815 -> 3.161.2
13.64:443
09/25/2025-19:58:25.005582 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (UDP) 172.16.150.130:59815 -> 3.161.2
13.64:443
09/25/2025-19:58:25.005577 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (UDP) 172.16.150.130:59815 -> 3.161.2
13.64:443
09/25/2025-19:58:25.005590 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (UDP) 172.16.150.130:59815 -> 3.161.2
13.64:443
09/25/2025-19:59:19.690675 [**] [1:1000001:1] ICMP test detected [**] [Classification: (null)] [Priority: 3] (ICMP) 172.16.150.130:8 -> 8.8.8.8:0
09/25/2025-19:59:19.690684 [**] [1:1000001:1] ICMP test detected [**] [Classification: (null)] [Priority: 3] (ICMP) 172.16.150.130:8 -> 8.8.8.8:0
09/25/2025-19:59:19.690691 [**] [1:1000001:1] ICMP test detected [**] [Classification: (null)] [Priority: 3] (ICMP) 172.16.150.130:8 -> 8.8.8.8:0
09/25/2025-19:59:19.735235 [**] [1:1000001:1] ICMP test detected [**] [Classification: (null)] [Priority: 3] (ICMP) 8.8.8.8:0 -> 172.16.150.130:0
09/25/2025-19:59:19.735235 [**] [1:1000001:1] ICMP test detected [**] [Classification: (null)] [Priority: 3] (ICMP) 8.8.8.8:0 -> 172.16.150.130:0
09/25/2025-19:59:19.735235 [**] [1:1000001:1] ICMP test detected [**] [Classification: (null)] [Priority: 3] (ICMP) 8.8.8.8:0 -> 172.16.150.130:0
```

FIGURE 2.6 : Alerte générée dans `fast.log` suite au ping ICMP.

En conclusion, Suricata est correctement configuré pour charger les règles locales et détecter du trafic ICMP simple, démontrant sa capacité à identifier des anomalies sur le réseau.

## 2.2 CONFIGURATION DE SYSLOG-NG

Le rôle de syslog-ng dans notre projet est de récupérer les événements JSON générés par Suricata, d'en conserver une copie locale à des fins de vérification, et de les transmettre automatiquement vers Elasticsearch pour qu'ils puissent être analysés et visualisés dans Kibana.

## 2.2.1 FICHER DE CONFIGURATION

Pour réaliser cette intégration, j'ai ajouté un fichier de configuration dédié dans `/etc/syslog-ng/conf`. Celui-ci définit une source pointant vers le fichier `eve.json` de Suricata, une destination locale où les événements sont recopiés, ainsi qu'une destination HTTP vers Elasticsearch.

```
julienlrzl@julienlrzl-VMware20-1:~$ sudo sed -n '1,200p' /etc/syslog-ng/conf.d/30-suricata-es.conf
[sudo] Mot de passe de julienlrzl :
# Suricata -> Elasticsearch (pass-through JSON, options compatibles)
source s_suricata {
    file("/var/log/suricata/eve.json" follow-freq(1) flags(no-parse));
};

destination d_elastic_http {
    http(
        url("http://127.0.0.1:9200/suricata-${YEAR}.${MONTH}.${DAY}/_doc")
        method("POST")
        headers("Content-Type: application/json")
        body("$MSG")
    );
};

log { source(s_suricata); destination(d_elastic_http); };
```

FIGURE 2.7 : Fichier de configuration syslog-ng pour l'intégration des événements Suricata.

## 2.2.2 DÉMARRAGE ET VÉRIFICATION DU SERVICE

Une fois le fichier ajouté, j'ai redémarré le service `syslog-ng` pour appliquer la configuration. Le statut confirme que le service est bien actif et fonctionne correctement.

```
julienlrzl@julienlrzl-VMware20-1:~$ sudo systemctl status syslog-ng --no-pager -l
● syslog-ng.service - System Logger Daemon
   Loaded: loaded (/usr/lib/systemd/system/syslog-ng.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-09-25 16:48:19 EDT; 3h 57min ago
     Invocation: 8de6aa4373e4d3aabb9404ce006f2d
       Docs: man:syslog-ng(8)
    Main PID: 15026 (syslog-ng)
      Tasks: 5 (limit: 8743)
    Memory: 58.8M (peak: 60.6M, swap: 20.9M, swap peak: 20.9M)
       CPU: 16.564s
    CGroup: /system.slice/syslog-ng.service
            └─15026 /usr/sbin/syslog-ng -F

sept. 25 16:48:19 julienlrzl-VMware20-1 systemd[1]: Starting syslog-ng.service - System Logger Daemon...
sept. 25 16:48:19 julienlrzl-VMware20-1 (yslog-ng)[15026]: syslog-ng.service: Referenced but unset environment variable evaluates to an empty string: SYSLOGNG_OPTS
sept. 25 16:48:19 julienlrzl-VMware20-1 systemd[1]: Started syslog-ng.service - System Logger Daemon.
```

FIGURE 2.8 : Vérification du bon fonctionnement du service `syslog-ng`.

## 2.2.3 COPIE LOCALE DES ÉVÉNEMENTS

Afin de faciliter le débogage, une copie locale des événements est conservée dans le fichier `/var/log/ids/suricata-collected.json`. On peut y observer directement les lignes JSON produites par Suricata et collectées par syslog-ng.

```
julienlrzi@julienlrzi-Vmware20-1:~$ tail -n 10 /var/log/ids/suricata-collected.json
Sep 25 16:20:15 julienlrzi-Vmware20-1 {"timestamp":"2025-09-25T16:20:14.824311-0400","event_type":"stats","stats":{"uptime":1376,"capture":{"kernel_packets":7941,"kernel_drops":0,"errors":0,"af_packet":{"busy_loop_avg":0,"polls":56090,"poll_signal":0,"poll_timeout":54385,"poll_data":1785,"poll_errors":0,"send_errors":0}},"decoder":{"pkts":7941,"bytes":8561583,"in_valid":0,"ipv4":7887,"ipv6":26,"ethernet":7941,"arp":28,"unknown_ethertype":0,"chdic":0,"raw":0,"null":0,"sll":0,"tcp":7671,"udp":241,"sctp":0,"icmpv4":0,"icmpv6":1,"ppp":0,"poppe":0,"geneve":0,"gre":0,"vlan":0,"vlan_qinq":0,"vlan_qinqinq":0,"vxlan":0,"vntag":0,"leee8021ah":0,"teredo":0,"ipv4_in_ipv6":0,"ipv6_in_ipv6":0,"mpls":0,"avg_pkt_size":1078,"max_pkt_size":1514,"max_mac_addr_src":0,"max_mac_addr_dst":0,"erspan":0,"nsh":0,"event":{"ipv4":{"pkt_too_small":0,"hlen_too_small":0,"iplen_smaller_than_hlen":0,"trunc_pkt":0,"opt_invalid":0,"opt_invalid_len":0,"opt_malformed":0,"opt_pad_required":0,"opt_eol_required":0,"opt_duplicate":0,"opt_unknown":0,"wrong_ip_version":0,"lcnpv6":0,"frag_pkt_too_large":0,"frag_overlap":0,"frag_ignored":0},"lcnpv4":{"pkt_too_small":0,"unknown_type":0,"ipv4_trunc_pkt":0,"ipv4_unknown_ver":0},"lcnpv6":{"unknown_type":0,"unknown_code":0,"pkt_too_small":0,"ipv6_unknown_version":0,"ipv6_trunc_pkt":0,"mid_message_with_invalid_hl":0,"unassigned_type":0,"experimentation_type":0,"ipv6":{"pkt_too_small":0,"trunc_pkt":0,"trunc_exthdr":0,"exthdr_dupl_fh":0,"exthdr_useless_fh":0,"exthdr_dupl_bh":0,"exthdr_dupl_hh":0,"exthdr_dupl_ch":0,"exthdr_dupl_ah":0,"exthdr_dupl_eh":0,"exthdr_invalid_optlen":0,"wrong_ip_version":0,"exthdr_ah_res_not_null":0,"hopopts_unknown_opt":0,"hopopts_only_padding":0,"dstopts_only_padding":0,"rh_type":0,"zero_len_pdn":0,"fh_non_zero_reserved_field":0,"data_after_nonce_header":0,"unknown_next_header":0},"lcnpv4":0,"frag_pkt_too_large":0,"frag_overlap":0,"frag_invalid_length":0,"frag_ignored":0,"ipv4_in_ipv6_too_small":0}}
```

FIGURE 2.9 : Copie locale des événements Suricata dans `suricata-collected.json`.

## 2.2.4 VÉRIFICATION CÔTÉ ELASTICSEARCH ET KIBANA

Enfin, les événements collectés sont bien transmis vers Elasticsearch. Dans Kibana, on retrouve un index `suricata-*` contenant les données issues de Suricata. La capture ci-dessous illustre un exemple d'événements reçus, confirmant le bon acheminement de bout en bout.

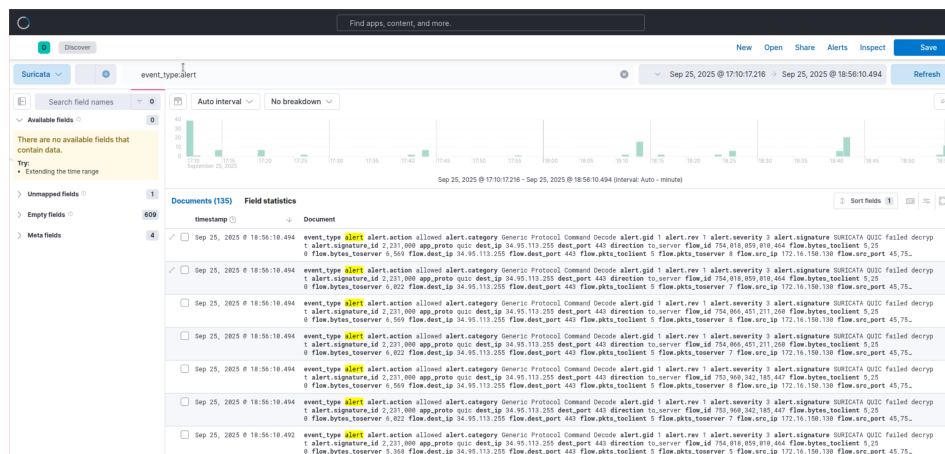


FIGURE 2.10 : Événements Suricata visibles dans Kibana après intégration avec syslog-ng et Elasticsearch.

En résumé, syslog-ng assure avec succès son rôle de passerelle entre Suricata et Elastic-search, permettant d'automatiser la collecte et l'envoi des événements de sécurité vers notre SIEM.

**CHAPITRE III**  
**SCÉNARIOS D'INTRUSION (5 CAS)**

## **CHAPITRE IV**

### **VISUALISATION ET ALERTES**



**CHAPITRE V**  
**ANALYSE ET DISCUSSION**

## CONCLUSION

**APPENDICE A**  
**PREMIÈRE ANNEXE**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.