TP Splunk

Membre du groupe (Nom et Prénom) :

# **Objectif**

L’objectif de ce TP est de vous apprendre à installer, configurer et manipuler Splunk. Afin d’être capable de manipuler les données détectées par Suricata afin de générer des Matrices de Flux.

# **Splunk**

Splunk est un logiciel de recherche, suivi et d’analyse de données via une interface web. On va utiliser Splunk pour afficher les données récoltées par Suricata et le présenter sous forme de Dashboards, graphes ou moteur de recherches…

## Installation

Pour installer Splunk Entreprise, récupérer le fichier « Splunk\*\*\*.deb » sur le Git.

Créer un dossier dans le /opt/ :

mkdir /opt/splunk

Rentrer dans ce nouveau dossier

cd /opt/splunk/

Déposer le fichier dans ce dossier puis installer Splunk :

dpkg –i splunk\*\*\*.deb

Si jamais ces erreurs «  ldconfig » introuvable … et « start-stop-deamon » introuvable … apparaissent, il est nécessaire de taper ces commandes :

export PATH=$PATH:/usr/local/sbin

export PATH=$PATH:/usr/sbin

export PATH=$PATH:/sbin

Puis tenter de nouveau d’installer Splunk.

Rentrer dans le fichier qui vient d’être crée :

cd /opt/splunk/bin

On peut taper à présent plusieurs commandes :

./splunk start : Démarre Splunk et son interface web

./splunk stop : Arrête Splunk

./splunk restart : Redémarre Splunk

./splunk help : Fourni des exemples de commandes Splunk et des explications à son sujet.

Lorsque Splunk sera lancé pour la première fois, ce dernier va demander un nom d’utilisateur et un mot de passe afin d’interagir avec son interface web. (Il se peut qu’il faudra également accepter la License Splunk lors du premier démarrage).

Une fois Splunk démarré, il fournira une adresse http privé pour accéder à son interface. (Il est recommandé d’utiliser un navigateur web reconnu tel que Firefox ou Edge pour y accéder.)

La commande suivante :

./splunk enable boot-start

Permet à Splunk de démarrer automatiquement lors du démarrage de votre machine.

Une fois Splunk démarré, l’interface web est disponible à l’adresse suivante : <http://localhost:8000>

Dans un premier temps, on souhaite monitorer les fichiers eve.json et fast.log de Suricata afin de pouvoir analyser et utiliser les données récoltés dans ces fichiers en temps réel.

## Configuration

Avant d’accéder à l’interface web, il faut modifier une variable d’un fichier de configuration de splunk pour qu’il puisse tourner sur les VMs de l’IUT.

cd /opt/splunk/etc/system/default/

nano server.conf

Copier la partie (#disk usage)

Coller dans le ficher */opt/splunk/etc/system/local/server.conf* à la fin du fichier

Modifier la ligne minFreeSpace = 50

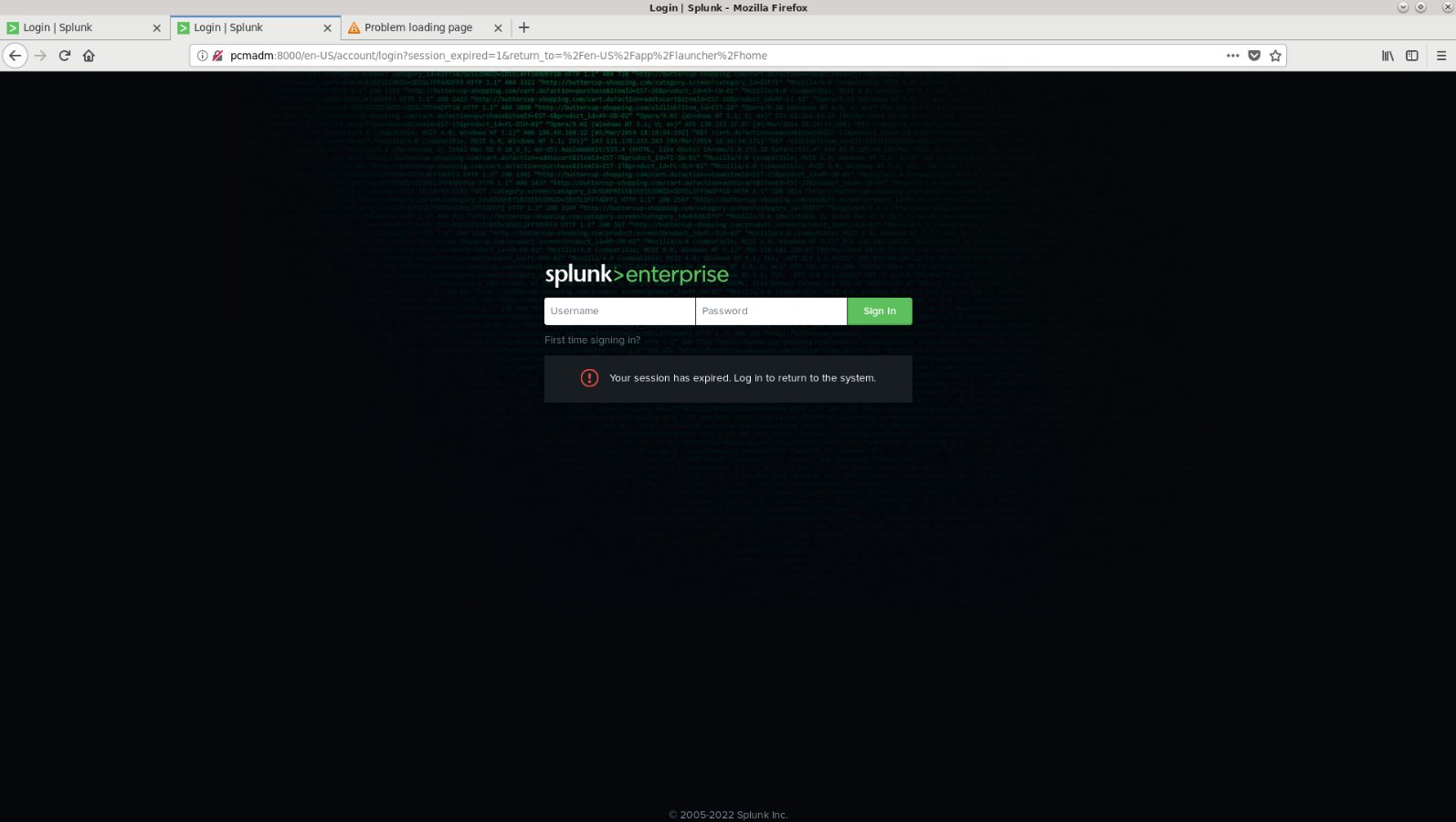
Puis redémarrer splunk

cd /opt/splunk/bin/

./splunk restart

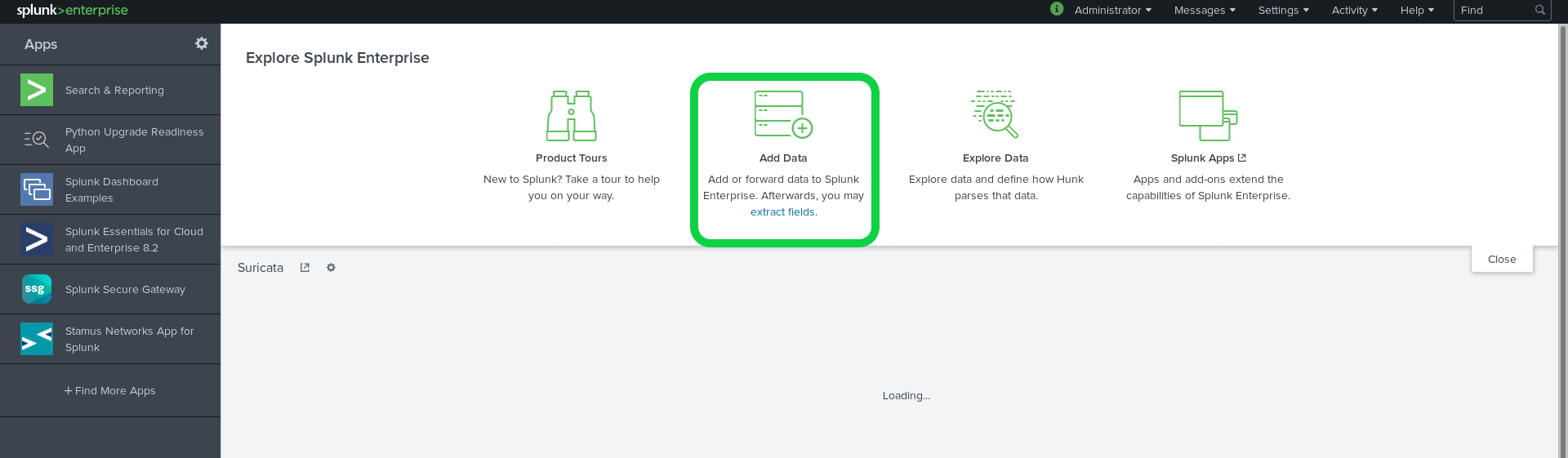
Lorsqu’on accède à l’interface web, rentrez le nom d’utilisateur et le mot de passe choisis lors de l’installation.

Voici la page de l’interface web :

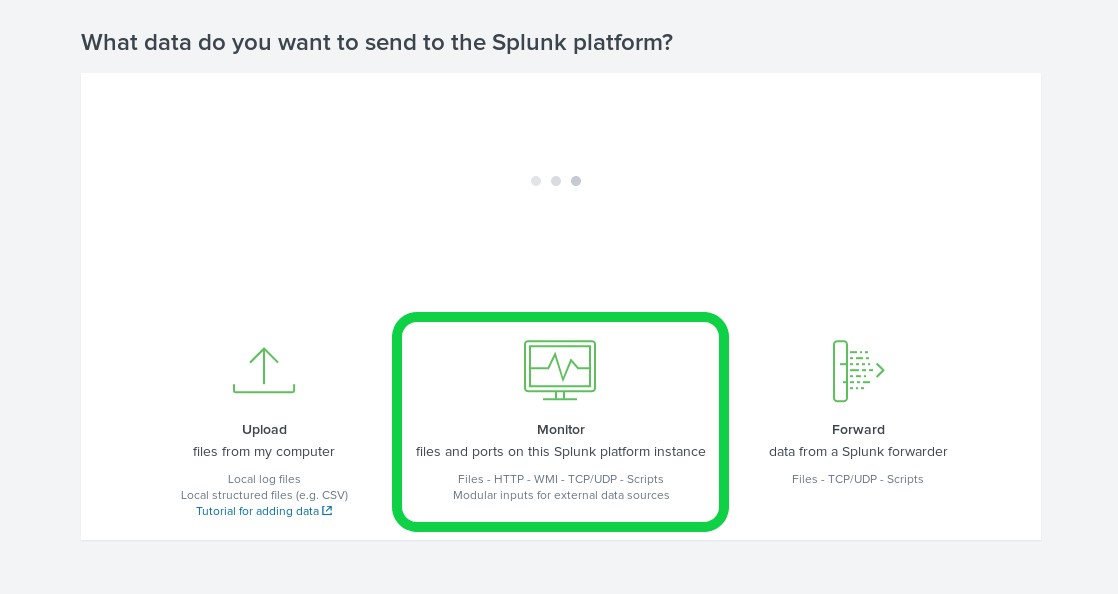


Dans un premier temps, on souhaite monitorer les fichiers eve.json et fast.log de Suricata afin de pouvoir analyser et utiliser les données récoltés dans ces fichiers en temps réel.

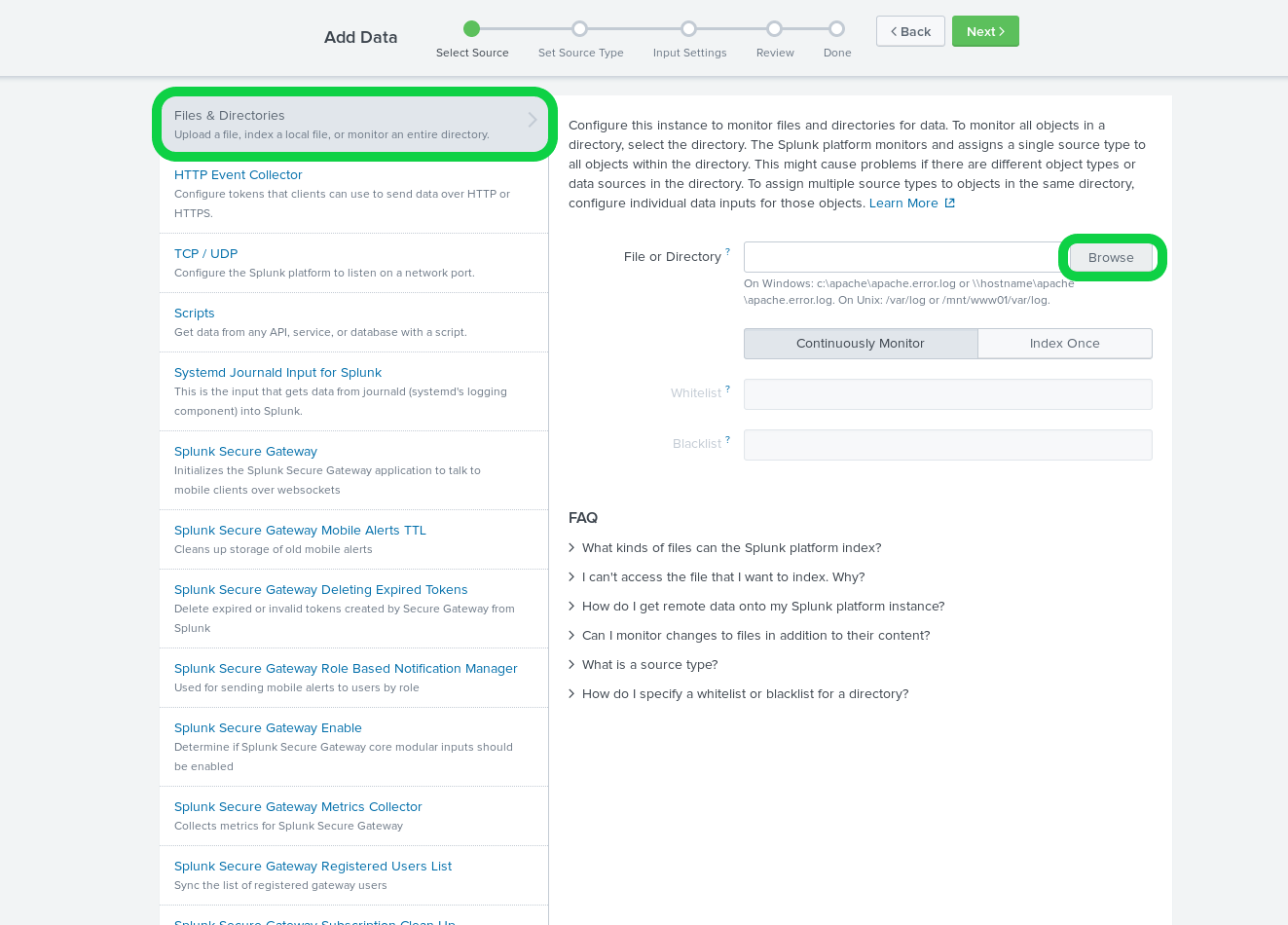
Pour cela cliquer sur « Add Data » :



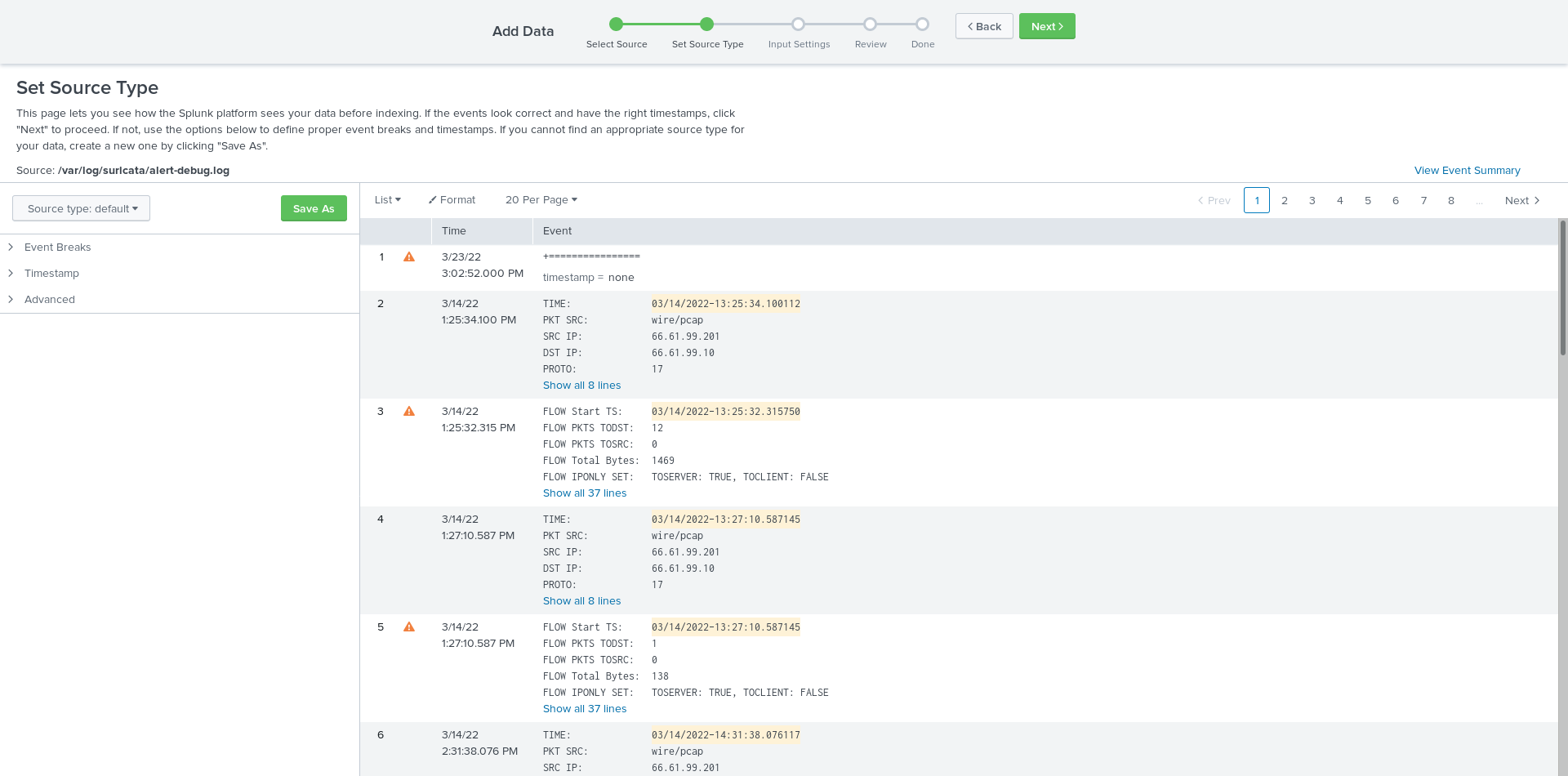
Cliquer sur « Monitor » :



Sélectionnez « Files & Directories » puis il faut cliquer sur « Browse » afin de sélectionner les fichiers souhaités (eve.json par exemple)



Cliquez ensuite sur « next ».

Voici un exemple avec fichier .log :

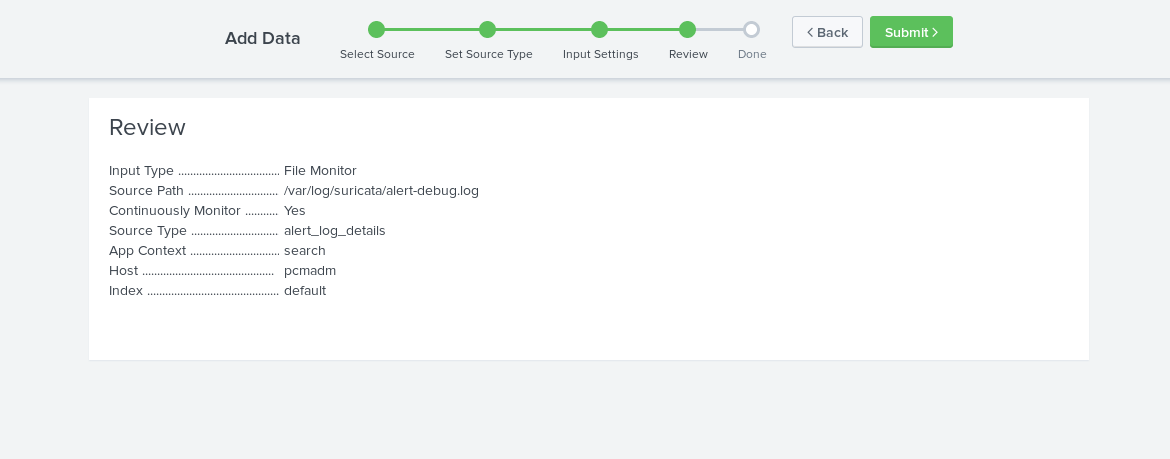
A côté de format, on peut décider de voir le fichier en tant que tableau, en brut ou en liste. Pour le format json, il est recommandé de mettre le fichier sous format de liste.

Splunk connait l’architecture du json, il suffit de cliquer sur « next ».

Pour fast.log par exemple, il faut créer le source type. A droite de l’écran nous avons accès à plusieurs paramètres afin de configurer comment Splunk va lire le fichier monitoré.

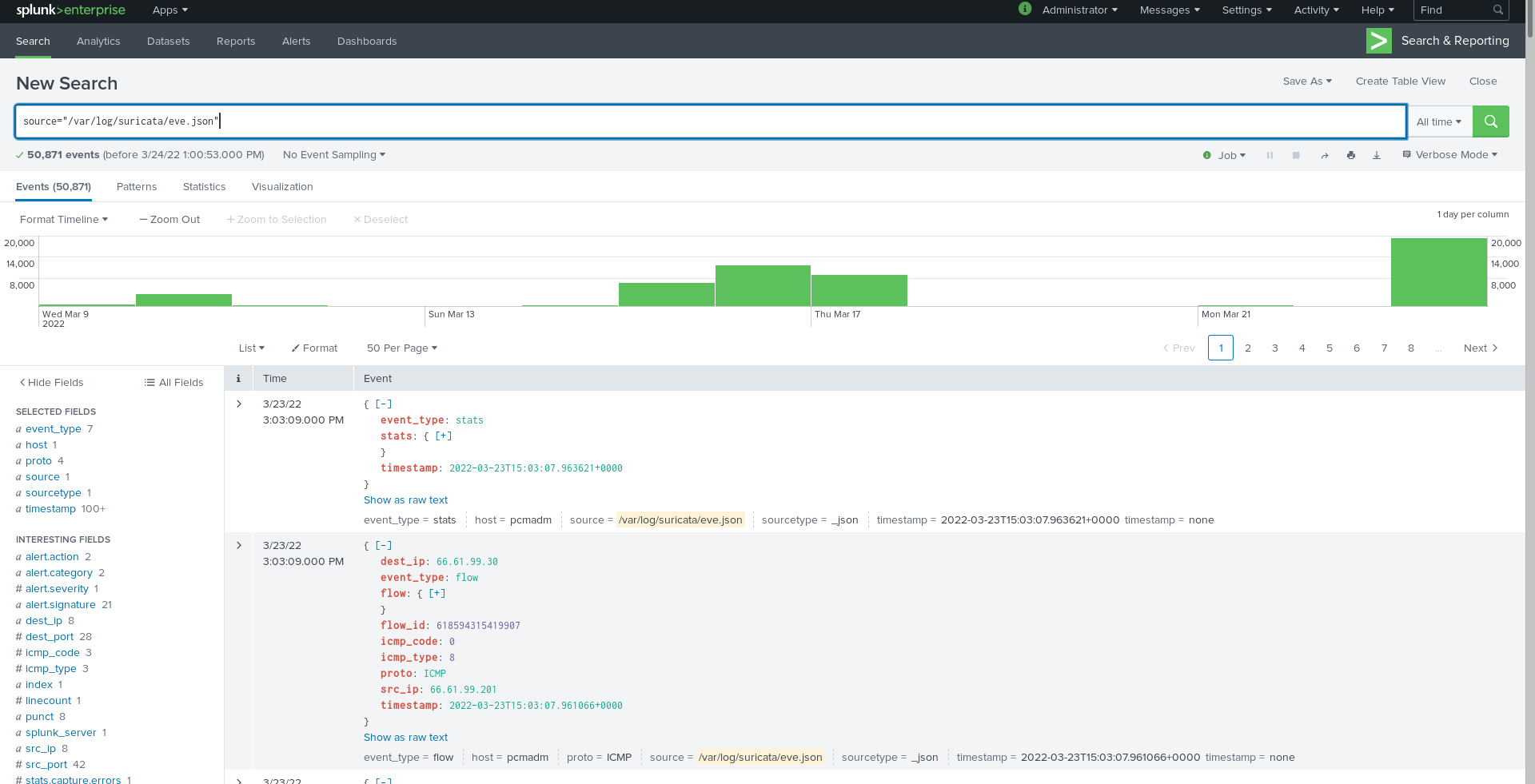
Dans la catégorie « Input Settings » cliquez sur next directement. Cette partie permet de configurer si le fichier doit être monitoré constamment par Splunk ou bien à intervalle régulier.

Enfin dernière étape consiste à énumérer les paramètres sélectionnés :



A présent il suffit de cliquer sur « Submit » pour finaliser le processus

## Recherches

A présent on peut commencer une recherche dans le fichier que vous venez de monitorer :

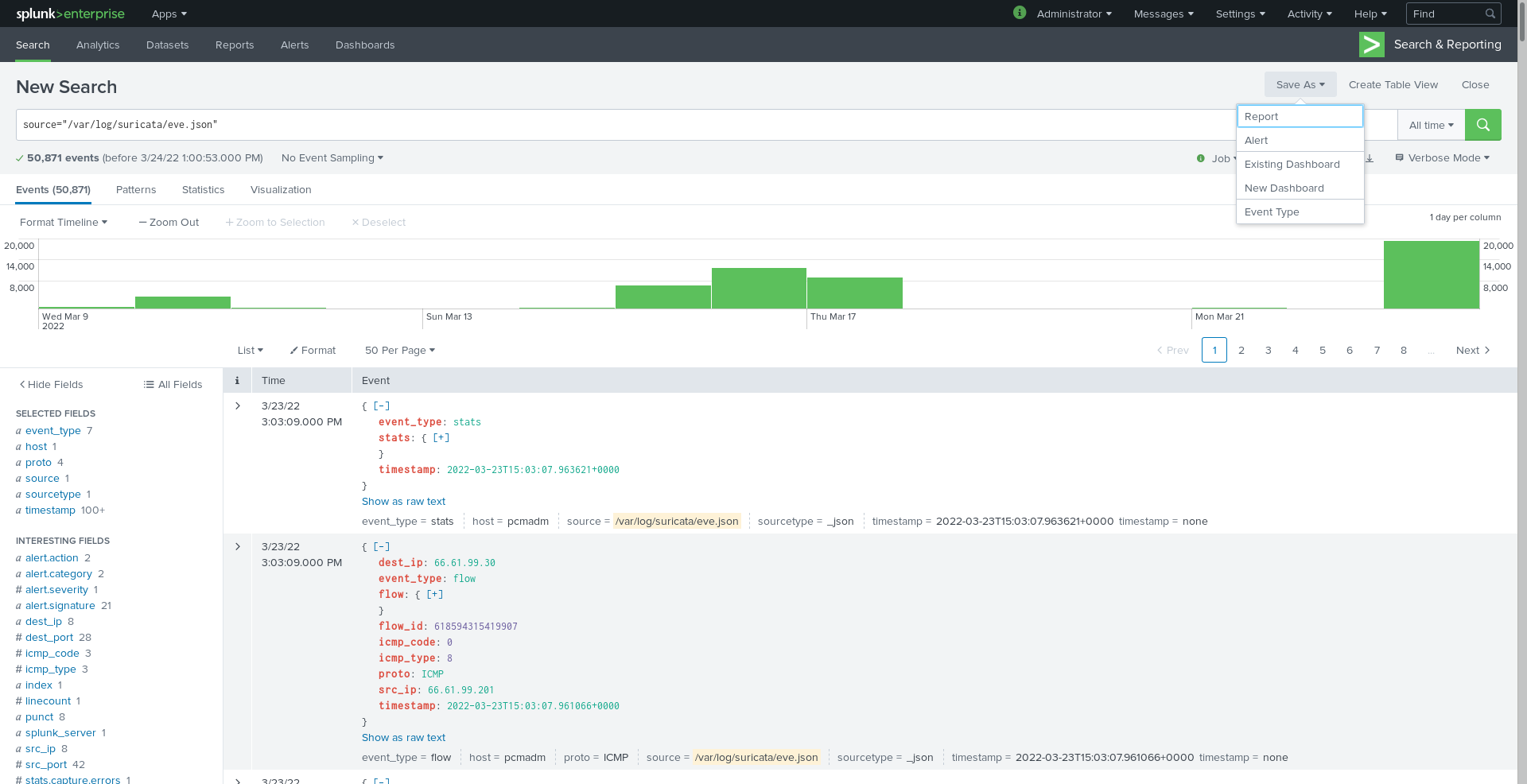
Le : **source= « /var/log/suricata/eve.json »** indique sur quel fichier se base la recherche.

Le graphique montre le nombre de résultats par rapport au temps. La colonne du milieu montre les lignes du fichier.

Et à gauche ce sont les champs qu’a reconnu Splunk à travers le fichier pour aider à la classification de certaines données (la date, le type d’évènement, l’ip source, etc.)

Cet onglet « Search » permet de catégoriser et de trier les différentes données des fichiers.

Voici la documentation officielle de slunk sur les recherches : [search - Splunk Documentation](https://docs.splunk.com/Documentation/Splunk/8.2.5/SearchReference/Search).

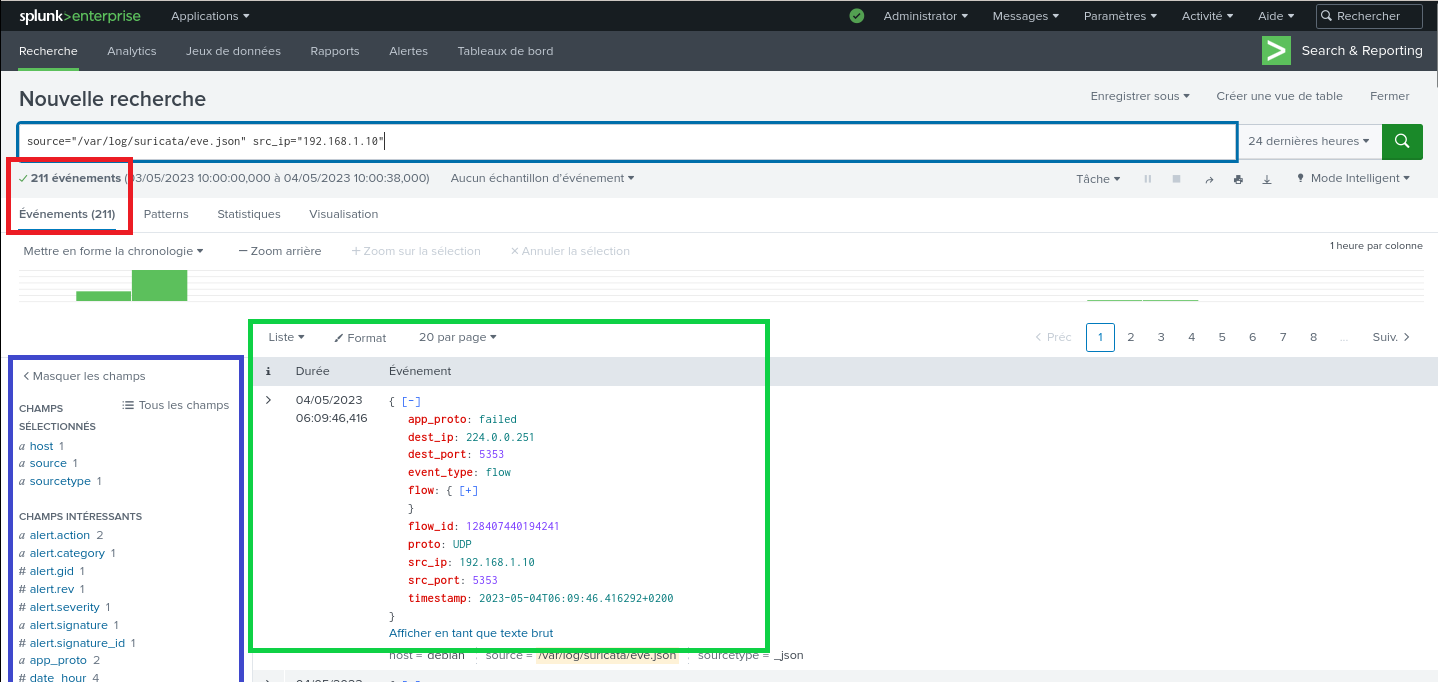
Le résultat des recherches peut être sauvegardé sous plusieurs formes : En rapport, l’enregistrer sur un dashboard ou même créer une alerte.

### Exemple de recherche :

La recherche : **source= « /var/log/suricata/eve.json » src\_ip =«192.168.1.10»**

Affiche la liste de tous les paquets détectés ayant pour ip source « 192.168.1.10 » cette semaine.

Le « src\_ip » est un champ du fichier json détecté par Splunk, on peut donc l’utiliser pour tirer facilement nos recherches.

On lui renseigne l’ip que l’on souhaite apparaître. Il n’y a pas de limite de paramètres dans une recherche Splunk.

En rouge nous avons le nombre d’évènements correspondant à la recherche.

En bleue la liste des champs du fichier json détectés par Suricata et donc exploitable.

En vert les informations de chaque paquet séparé par les champs provenant du fichier json.

La recherche : **source= « /var/log/suricata/eve.json » proto = «UDP» | stats count by proto**

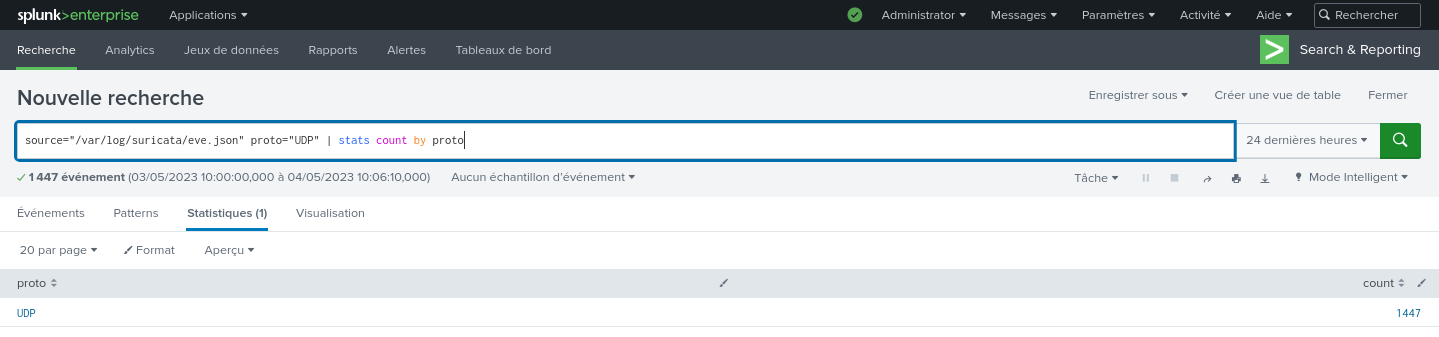
Affiche le nombre de paquets détectés par Suricata avec le protocole UDP.

« proto » est le champ provenant du fichier json correspondant aux protocoles des paquets détectés.

Le « | stats » permet à la recherche Splunk d’ajouter un élément de statistique.

Le « count » permet à Splunk de compter le nombre de résultats que provoque la recherche.

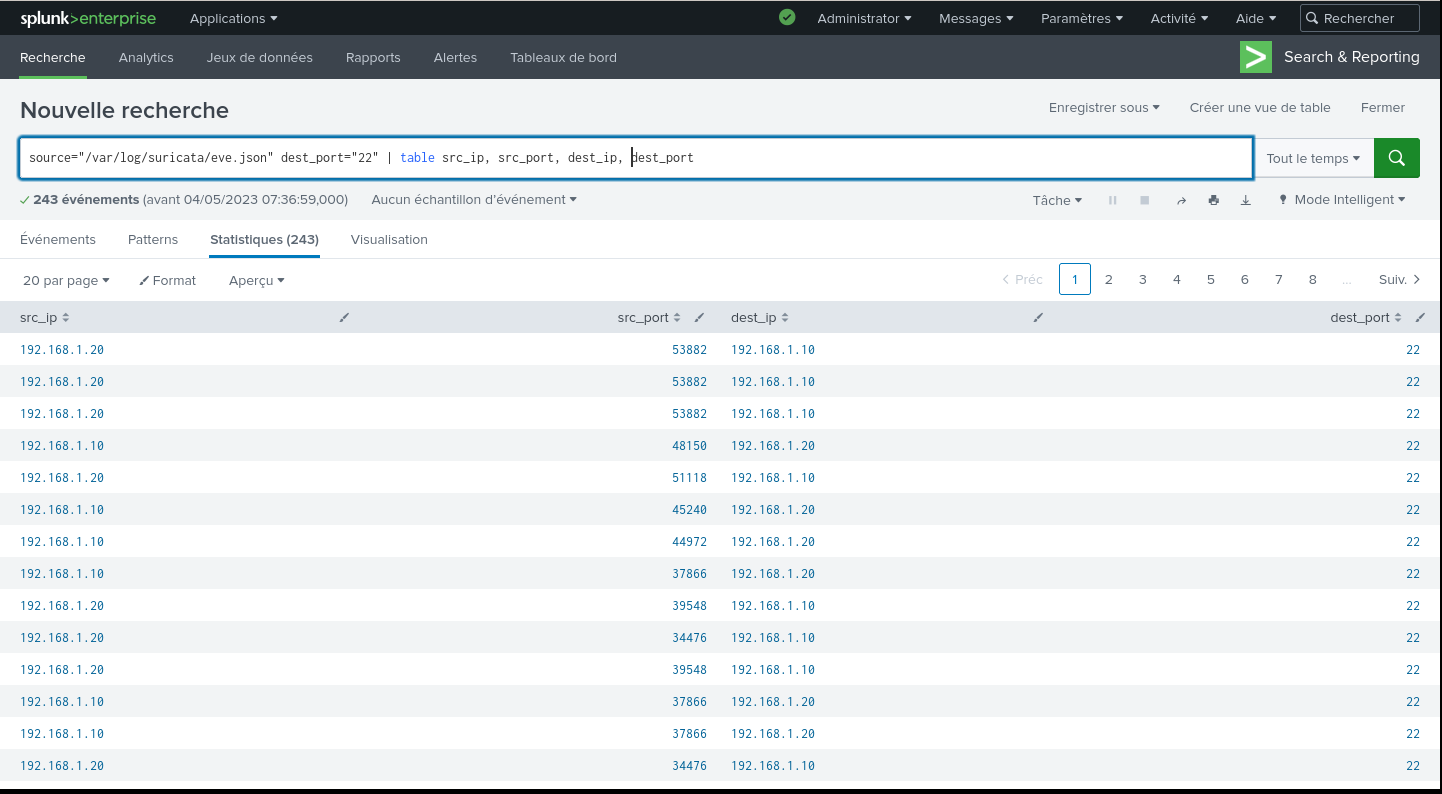
Le « by proto » ajoute un élément de référence. Dans cet exemple les protocoles sont les éléments de références. **(Il faut cependant que ce soit un champ du fichier)**



La recherche : **source= « /var/log/suricata/eve.json » dest\_port= «22» | table src\_ip, src\_port, dest\_ip, dest\_port**

Affiche un tableau récapitulant toutes les paquets ayant le port 22 comme port de destination. Le tableau est classé par ip et port en source et ip et port en destination.

Le « | table » permet de former la recherche sous forme de tableau. Les champs qui suivent défini l’ordre et les champs qui composent le tableau de recherche.

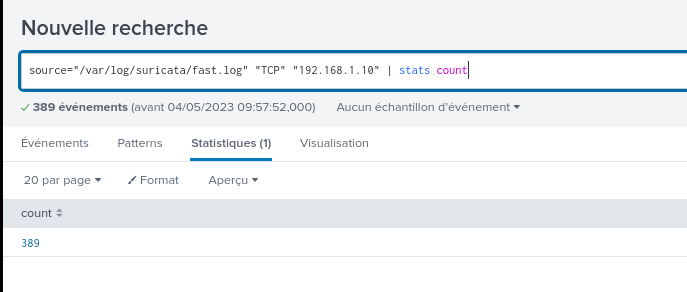


La recherche : **source= « /var/log/suricata/fast.log » «TCP» « 192.168.1.10 » | stats count**

Affiche le nombre d’alerte générée par Suricata ayant le protocole TCP et ayant comme IP source ou destination 192.168.1.10.

Étant donné que le fichier *fast.log* est un fichier de log et non un fichier json, il ne possède pas de champ que Splunk peut récupérer. Pour manipuler ce dossier, il faut donc faire des recherches par string.

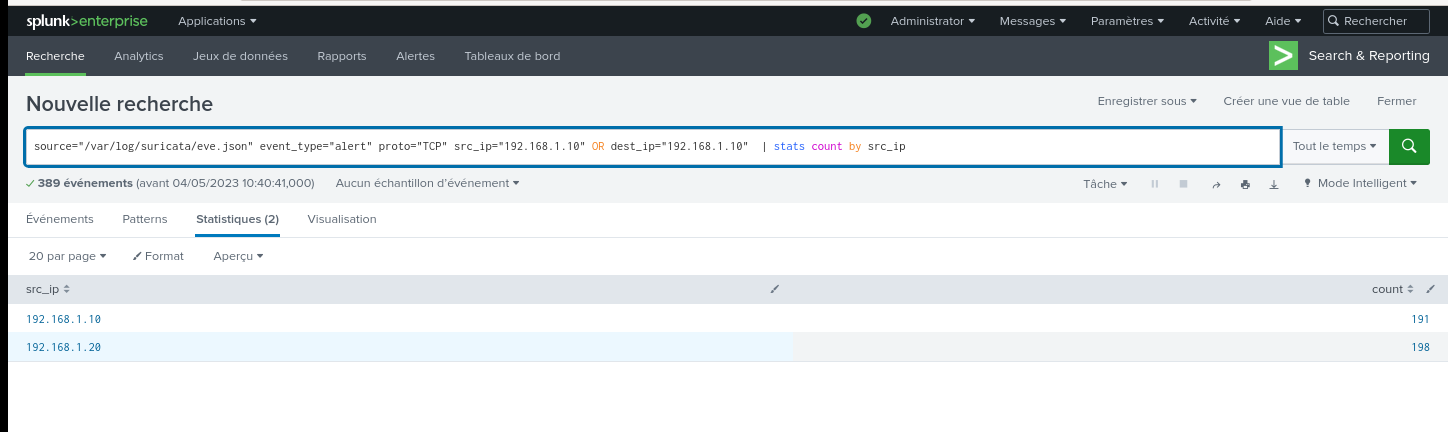
Comparé à la recherche « **source= « /var/log/suricata/eve.json » proto = «UDP» | stats count by proto** » on peut voir qu’on le tableau n’est pas structuré correctement car étant donné qu’il n’y a pas de champ, impossible de préciser avec un « by proto » avec le « count ».



Ce qui rend la manipulation du fichier *fast.log* compliqué. Cependant on peut contourner le problème grâce au fichier *eve.json.*

En effet ce fichier récupère tous les paquets détectés par Suricata, dont les alertes. Si on reprends la recherche ci-dessus mais en la modifiant pour qu’elle fonctionne avec les alertes détectés stockées dans le fichier *eve.json* :

**source= « /var/log/suricata/eve.json » event\_type = «alert» proto=«TCP» src\_ip=« 192.168.1.10 » OR dest\_ip=« 192.168.1.10 »| stats count by src\_ip**



On retrouve un tableau plus lisible et le même nombre d’évènements que la précédente recherche.

Le « event\_type » correspond au champ du fichier json qui est attribué aux alertes de Suricata.

Le « OR » permet d’ajouter l’opérateur OU dans notre recherche entre deux critères.

On peut s’apercevoir que Splunk est très puissant et permet de très nombreuses manipulation des données récoltées.

Toutes les recherches ci-dessus était pour les données datant des dernières 24h. On peut modifier facilement le période de temps que l’on souhaite. Il suffit de cliquer à droite de la barre de rechercher et sélectionner la période désirée. Par exemple pour prendre en compte les données datant d’une semaine :

### **Partie Pratique :**

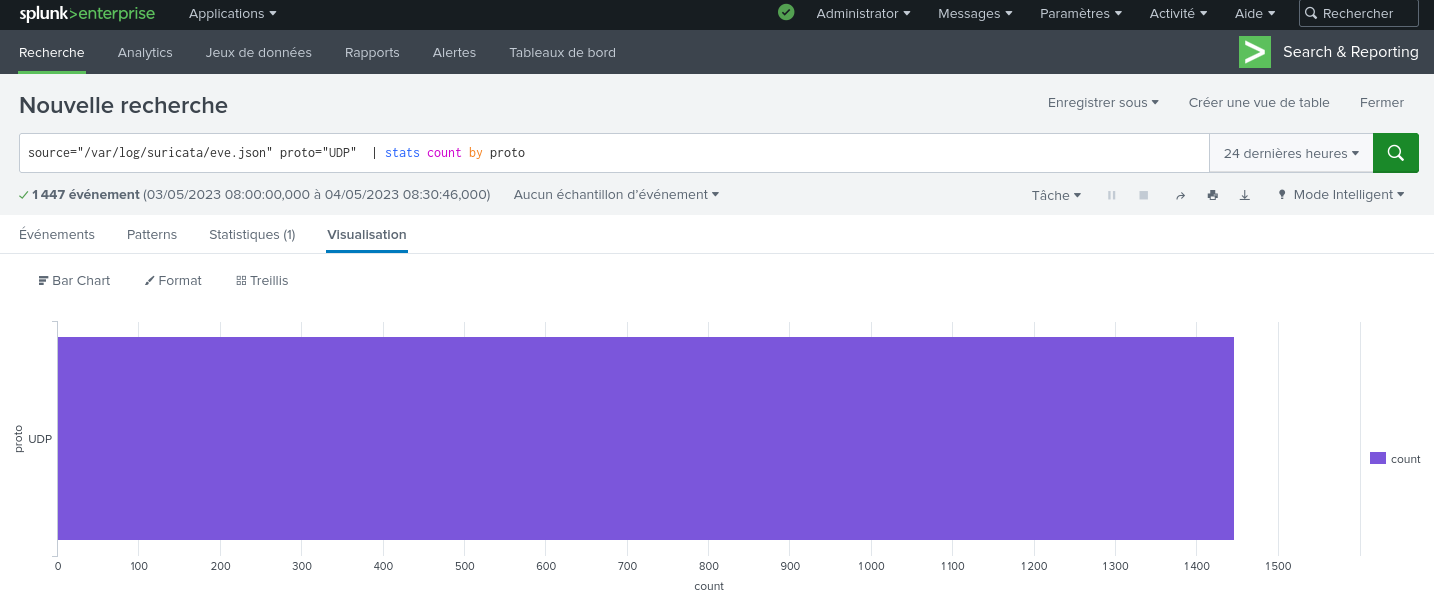
Rédigez ci-dessous la règle Splunk permettant d’afficher le nombre **d’alerte** généré par Suricata avec le protocole ICMP :

Rédigez ci-dessous la règle Splunk permettant d’afficher tous les paquets détectés par Suricata généré par « PC Utilisateur » vers « Suricata » (précisez les IPs de vos machines.)

## Dashboards

Splunk permet de créer et de modifier facilement et rapidement des dashboards. Pour cela on va transformer les recherches réalisées en graphiques.

Pour générer un graphique, reprenons la recherche vu plus tôt : **source= « /var/log/suricata/eve.json » proto = «UDP» | stats count by proto**



A gauche du résultat de la recherche, on peut voir un onglet nous indiquant le format de la recherche (Evenements, Patterns, Statistiques, Visualisation).

Il suffit de cliquer sur « Visualisation » pour transformer la recherche en graphique de notre choix.

Une multitude de forme est possible (colonne, ligne, camembert, etc.)

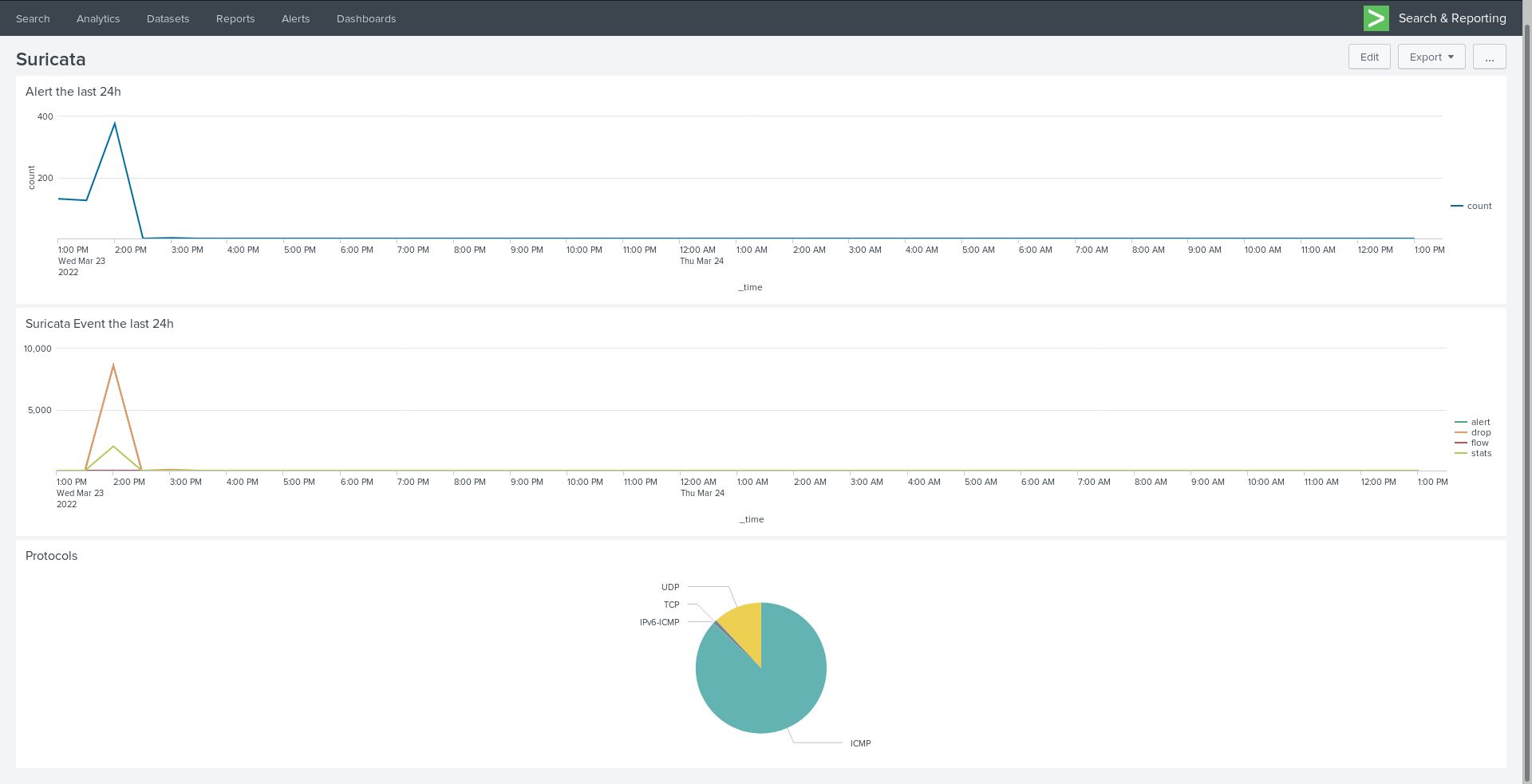
Il est nécessaire que la recherche soit en mode « stats » afin d’afficher un graphique.

Les Dashboards permets de regrouper une panoplie de recherches et d’accéder facilement aux données qui sont stockées dans les fichiers monitorés.

On peut modifier les dashboards selon vos besoins, pour cela il suffit de sélectionner « Edit ».

Depuis cet onglet on peut disposer nos différentes recherches sur nos dashboards comme on le souhaitez, dans l’ordre que l’on désire.

Les dashboards fonctionnent en HTML et on peut les modifier directement en ligne de code. Pour cela, il suffit de cliquer sur le bouton « Source ».

Exemple de Dashboard :

Ce dashboard regroupe 3 recherches différentes :

* Les alertes détectées par Suricata ces dernières 24h (Un simple count du nombre de ligne dans le fichier fast.log)
* Les différents évènements détectés par Suricata ces dernières 24h (A travers le fichier *eve.json* on va compter chaque type de paquet ; les alertes, les drops, les fluw et les paquets de statistiques).
* Et enfin un graphique représentant la répartition des différents protocoles détectés par Suricata depuis son installation. (On fait un count de tous les paquets en catégorisant les protocoles.

### **Partie Pratique :**

Faites un dashboard regroupant trois recherches différentes sous format graphique (vous pouvez choisir la forme des graphiques que vous souhaitez, tant que cela est lisible) :

* La première indiquant le nombre d’alerte de Suricata ce dernier mois.
* La seconde indiquant le nombre et le type d’évènements identifié par Suricata ces dernières 2 dernières semaines.
* Enfin la dernière représentant la répartition des différentes IPs détectés par Suricata (sans paramètre de temps)

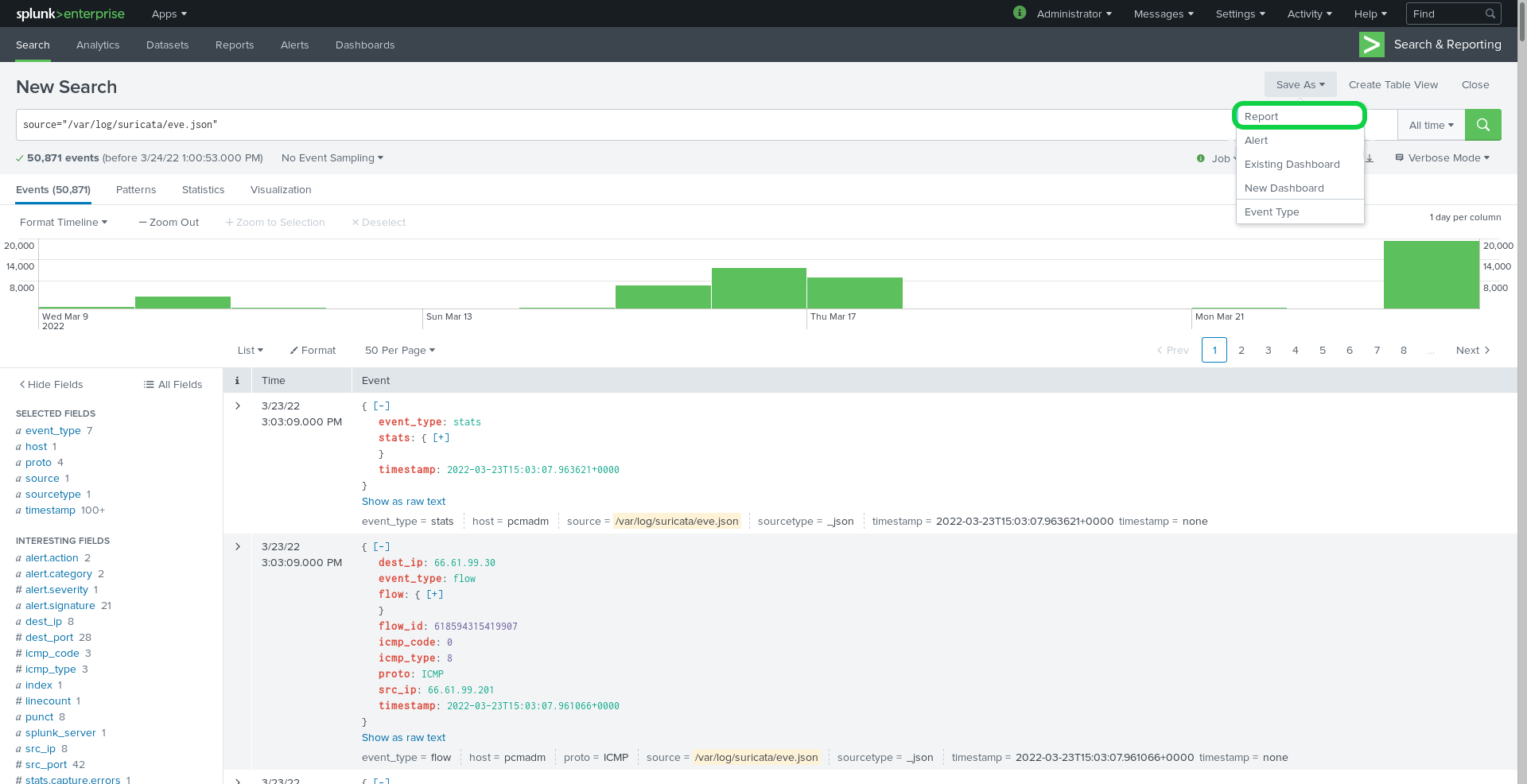
(Si vous n’avez pas suffisamment de données sur les 2 dernières semaines, relancer Suricata en CLI sur l’interface réseau de la VM connectée au réseau de l’IUT. Puis patientez quelques minutes que le fichier récupère des données).

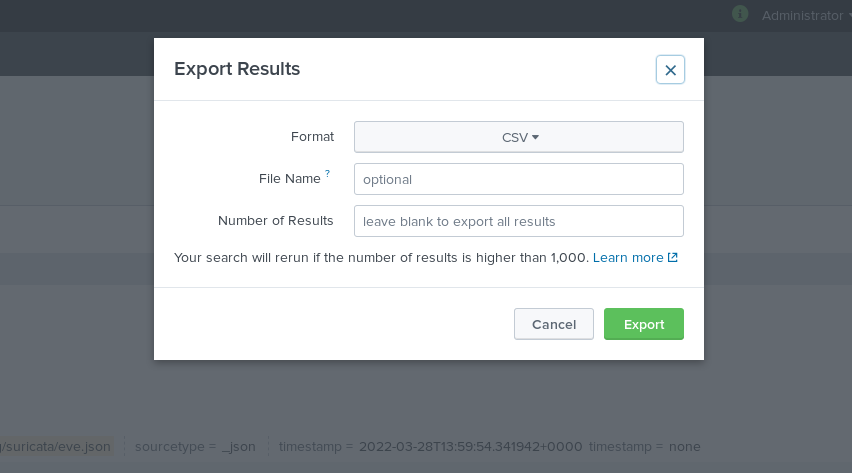
*Mettez un screen de votre dashboard ici*

*Ecrivez les trois règles utilisés dans votre dashboard*

## Reports

Enfin on peut également sauvegarder nos recherches sous formes de rapports, de façon à les sauvegarder mais également de les exporter en format PDF ou CSV.





Permettant ainsi à Splunk de manipuler les données selon notre besoin puis d’exporter ce nouveau jeu de données dans un format utilisable via un script, un logiciel etc.

### **Partie Pratique :**

Générer des rapports représentants :

* Une courbe montrant les alertes Suricata en fonction des semaines.
* Un tableau avec les IPs les plus présentes dans l’envoi de paquet (IP source) par ordre croissant.
* Le protocole ayant généré le plus d’alerte depuis le début de Suricata (Le nombre d’alerte et le protocole).
* Un graphique représentant les ports les plus visés par les paquets reçus.

Enfin générer une matrice des flux en format Excel (avec des champs pouvant être trié) ayant pour colonnes : Src\_IP, Src\_Port, Dest\_IP, Dest\_Port, Protocol. **Et sans doublons.**

*Envoyez les rapports en pièce jointe avec ce TP complété dans un zip. De plus n’oubliez pas d’écrire également les recherches splunk générant les rapports.*

# **Conclusion**

Dans ce TP, nous avons vus comment installer Splunk, comment le configurer et y ajouter les dossiers de Suricata.

Nous avons pu ensuite étudier le fonctionnement de recherche que propose Splunk.

La génération de ses recherches en graphiques et la création de dashboards regroupant plusieurs recherches.

Enfin nous avons également étudiés l’export des résultats de recherche de Splunk sous plusieurs formats pour une exploitation facile.

Vous avez à présent les bases pour générer des Matrices des Flux à l’aide de Suricata et Splunk.

Mail : [quentin.lehoux@outlook.com](mailto:quentin.lehoux@outlook.com)

Objet : COURS\_IUT GROUPE (Nom de famille) TP\_SPLUNK