Comment installer et configurer Suricata IDS avec Elastic Stack sur Debian 12

Suricata est un outil de surveillance de réseau qui examine et traite chaque paquet de trafic Internet transitant par votre serveur. Il peut générer des événements de journal, déclencher des alertes et supprimer du trafic lors de la détection d'une activité suspecte.

Vous pouvez soit installer Suricata sur une seule machine pour surveiller son trafic, soit le déployer sur un hôte de passerelle pour analyser tout le trafic entrant et sortant des autres serveurs qui y sont connectés. Tu peux combinez Suricata avec Elasticsearch, Kibana et Filebeat pour créer un outil de gestion des informations et des événements de sécurité (SIEM).

Dans ce didacticiel, vous allez installer Suricata IDS avec ElasticStack sur un serveur Debian 12. Les différents composants de la pile sont :

- Elasticsearch pour stocker, indexer, corréler et rechercher les événements de sécurité du serveur.
- Kibana pour afficher les journaux stockés dans Elasticsearch.
- Filebeat pour analyser le fichier journal de Suricata `eve. json` et envoyer chaque événement à Elasticsearch pour traitement.
- Suricata pour analyser le trafic réseau à la recherche d'événements suspects et supprimer les paquets non valides.

Le didacticiel est divisé en deux parties, la première partie traitera de l'installation et de la configuration de Suricata, et la deuxième partie traitera de l'installation et de la configuration d'Elastic Stack.

Nous allons installer Suricata et la stack Elastic sur différents serveurs pour notre tutoriel.

Conditions préalables

- Les serveurs hébergeant la Suite Elastic et Suricata doivent disposer d'un minimum de 4 Go de RAM et de 2 cœurs de processeur.
- Les serveurs doivent pouvoir communiquer entre eux en utilisant des adresses IP privées.
- Les serveurs doivent exécuter Debian 12 avec un utilisateur sudo non root.
- Si vous souhaitez accéder aux tableaux de bord Kibana de partout, configurez un domaine (`kibana.exemple.com`) pointant vers le serveur sur lequel Elasticsearch sera installé.
- Installez les packages essentiels sur les deux serveurs. Certains d'entre eux sont peut-être déjà installés.

```
$ sudo apt install wget curl nano ufw software-properties-common dirmngr apt-transport-https gnupg2 ca-certificates lsb-release debian-archive-keyring unzip -y
```

- Assurez-vous que tout est mis à jour sur les deux serveurs.

```
$ sudo apt update
```

PARTIE 1

Étape 1 - Installer Suricata

Suricata est disponible dans les dépôts officiels Debian. Installez-le à l'aide de la commande suivante.

```
$ sudo apt install suricata
```

Le service Suricata est automatiquement activé et démarré. Avant de continuer, arrêtez le service Suricata car nous devons d'abord le configurer.

```
$ sudo systemctl stop suricata
```

Étape 2 - Configurer Suricata

Suricata stocke sa configuration dans le fichier `/etc/suricata/suricata.yaml` . Le Suricata est le mode IDS (Intrusion Detection System), dans lequel le trafic est uniquement enregistré et non arrêté. Si vous êtes nouveau sur Suricata, vous devez laisser le mode inchangé. Une fois que vous l'avez configuré et en avez appris davantage, vous pouvez activer le mode IPS (Intrusion Prevention System).

Activer l'ID de communauté

Le champ Community ID facilite la corrélation des données entre les enregistrements générés par différents outils de surveillance. Puisque nous utiliserons Suricata avec Elasticsearch, l'activation de l'ID communautaire peut être utile.

Ouvrir le fichier `/etc/suricata/suricata.yaml` pour l'édition.

```
$ sudo nano /etc/suricata/suricata.yaml
```

Locate the line # Community Flow ID and set the value of the variable community-id to true.

```
# Community Flow ID
# Adds a 'community_id' field to EVE records. These are meant to give
# records a predictable flow ID that can be used to match records to
# output of other tools such as Zeek (Bro).
#
# Takes a 'seed' that needs to be same across sensors and tools
# to make the id less predictable.

# enable/disable the community id feature.
community-id: true
```

Enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité.

Désormais, vos événements porteront un identifiant comme `1:S+3BA2UmrHK0Pk+u3XH78GAFTtQ=` que vous pouvez utiliser pour faire correspondre des ensembles de données entre différents outils de surveillance.

Sélectionnez l'interface réseau

Sélectionnez l'interface réseau

Le fichier de configuration Suricata par défaut inspecte le trafic sur l'interface eth0 device/network . Si votre serveur utilise une interface réseau différente, vous devrez la mettre à jour dans la configuration. Vérifiez le nom de périphérique de votre interface réseau à l'aide de la commande suivante.

```
$ ip -p -j route show default
```

Vous recevrez une sortie comme celle-ci.

```
[ {
        "dst": "default",
        "gateway": "159.223.208.1",
        "dev": "eth0",
        "protocol": "static",
        "flags": [ ]
    } ]
```

La variable dev fait référence au périphérique réseau. Dans notre sortie, il montre eth0 comme périphérique réseau. Votre sortie peut être différente selon votre système. Maintenant que vous connaissez le nom de votre appareil, ouvrez le fichier de configuration.

```
$ sudo nano /etc/suricata/suricata.yaml
```

Trouver la ligne `af-packet:` autour du numéro de ligne 580. En dessous, définissez la valeur de la variable `interface` au nom de l'appareil de votre système.

```
# Linux high speed capture support
af-packet:
  - interface: eth0
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
. . .
```

Si vous souhaitez ajouter des interfaces supplémentaires, vous pouvez le faire en les ajoutant en bas de la page af-paquet section vers la ligne 650.

Pour ajouter une nouvelle interface, insérez-la juste au-dessus du - interface: default comme indiqué ci-dessous.

```
    #  For eBPF and XDP setup including bypass, filter and load balancing, please
    #  see doc/userguide/capture-hardware/ebpf-xdp.rst for more info.

  - interface: enp0s1
    cluster-id: 98
...
  - interface: default
    #threads: auto
    #use-mmap: no
    #tpacket-v3: yes
```

Nous avons ajouté une nouvelle interface enp0s1 et une valeur unique pour le cluster-id variable dans notre exemple. Vous devez inclure un ID de cluster unique avec chaque interface que vous ajoutez.

Trouver la ligne pcap : et en dessous, définissez la valeur de la variable interface au nom de l'appareil de votre système.

```
# Cross platform libpcap capture support
pcap:
  - interface: eth0
    # On Linux, pcap will try to use mmap'ed capture and will use "buffer-size"
    # as total memory used by the ring. So set this to something bigger
    # than 1% of your bandwidth.
```

Pour ajouter une nouvelle interface comme avant, insérez-la juste au-dessus du - interface : default comme indiqué ci-dessous.

```
- interface: enp0s1
# Put default values here
  - interface: default
    #checksum-checks: auto
```

Une fois que vous avez terminé, enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité.

Étape 3 - Configurer les règles Suricata

Suricata, par défaut, utilise uniquement un ensemble limité de règles pour détecter le trafic réseau. Vous pouvez ajouter d'autres ensembles de règles provenant de fournisseurs externes à l'aide d'un outil appelé suricata-mise à jour . Exécutez la commande suivante pour inclure des règles supplémentaires.

```
$ sudo suricata-update -o /etc/suricata/rules
4/10/2023 -- 14:12:05 - <Info> -- Using data-directory /var/lib/suricata.
4/10/2023 -- 14:12:05 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
4/10/2023 -- 14:12:05 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
.....
4/10/2023 -- 14:12:05 - <Info> -- No sources configured, will use Emerging Threats Open
4/10/2023 -- 14:12:05 - <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-6.0.10/emerging.rules.tar.gz.
 100% - 4073339/4073339
.....
4/10/2023 -- 14:12:09 - <Info> -- Writing rules to /etc/suricata/rules/suricata.rules: total: 45058; enabled: 35175; added: 45058; removed 0; modified: 0
4/10/2023 -- 14:12:10 - <Info> -- Writing /etc/suricata/rules/classification.config
4/10/2023 -- 14:12:10 - <Info> -- Testing with suricata -T.
4/10/2023 -- 14:12:33 - <Info> -- Done.
```

-o /etc/suricata/rules Une partie de la commande demande à l'outil de mise à jour d'enregistrer les règles dans le /etc/suricata/rules annuaire. Ce paramètre est important sinon vous obtiendrez ce qui suit L'erreur lors de la validation.

```
<Warning> - [ERRCODE: SC_ERR_NO_RULES(42)] - No rule files match the pattern /etc/suricata/rules/suricata.rules
```

Ajouter des fournisseurs d'ensembles de règles

Vous pouvez étendre les règles de Suricata en ajoutant plus de fournisseurs. Il peut récupérer des règles auprès de divers fournisseurs gratuits et commerciaux.

Vous pouvez répertorier la liste des fournisseurs par défaut à l'aide de la commande suivante.

```
$ sudo suricata-update list-sources
```

Parexemple, si vous souhaitez inclure le green/hunting ensemble de règles, vous pouvez l'activer avec la commande suivante.

```
$ sudo suricata-update enable-source tgreen/hunting
4/10/2023 -- 14:24:26 - <Info> -- Using data-directory /var/lib/suricata.
4/10/2023 -- 14:24:26 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml 4/10/2023 -- 14:24:26 - <Info> -- Using /etc/suricata/rules for Suricata provided rules. 4/10/2023 -- 14:24:26 - <Info> -- Found Suricata version 6.0.10 at /usr/bin/suricata.
4/10/2023 -- 14:24:26 - <Info> -- Creating directory /var/lib/suricata/update/sources
4/10/2023 -- 14:24:26 - <Info> -- Enabling default source et/open
4/10/2023 -- 14:24:26 - <Info> -- Source tgreen/hunting enabled
```

Exécutez le suricata-mise à jour commandez à nouveau pour télécharger et mettre à jour les nouvelles règles. Suricata, par défaut, peut traiter toute modification de règle sans redémarrer.

Étape 4 - Valider la configuration de Suricata

Suricata est livré avec un outil de validation pour vérifier le fichier de configuration et les règles pour détecter les erreurs. Exécutez la commande suivante pour exécuter l'outil de validation.

```
$ sudo suricata -T -c /etc/suricata/suricata.yaml -v
4/10/2023 -- 14:24:43 - <Info> - Running suricata under test mode
4/10/2023 -- 14:24:43 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
4/10/2023 -- 14:24:43 - <Info> - CPUs/cores online: 2
4/10/2023 -- 14:24:43 - <Info> - fast output device (regular) initialized: fast.log
4/10/2023 -- 14:24:43 - <Info> - eve-log output device (regular) initialized: eve.json
4/10/2023 -- 14:24:43 - <Info> - stats output device (regular) initialized: stats.log
4/10/2023 -- 14:24:53 - <Info> - 1 rule files processed. 35175 rules successfully loaded, 0 rules failed
4/10/2023 -- 14:24:53 - <Info> - Threshold config parsed: 0 rule(s) found
4/10/2023 -- 14:24:54 - <Info> - 35178 signatures processed. 1255 are IP-only rules, 5282 are inspecting packet payload, 28436 inspect application layer, 108 are decoder event only 4/10/2023 -- 14:25:07 - <Notice> - Configuration provided was successfully loaded. Exiting.
4/10/2023 -- 14:25:07 - <Info> - cleaning up signature grouping structure... complete
```

L'indicateur -T indique à Suricata de s'exécuter en mode test, l'indicateur -c configure l'emplacement du fichier de configuration et l'indicateur -v imprime la sortie détaillée de la commande. En fonction de la configuration de votre système et du nombre de règles ajoutées, la commande peut prendre quelques minutes.

Étape 5 - Exécuter Suricata

Maintenant que Suricata est configuré et installé, il est temps d'exécuter l'application.

```
$ sudo systemctl start suricata
```

Vérifiez l'état du processus.

```
$ sudo systemctl status suricata
```

Vous devriez voir le résultat suivant si tout fonctionne correctement.

```
? suricata.service - Suricata IDS/IDP daemon
     Loaded: loaded (/lib/systemd/system/suricata.service; enabled; preset: enabled)
     Active: active (running) since Wed 2023-10-04 14:25:49 UTC; 6s ago
       Docs: man:suricata(8)
             man:suricatasc(8)
             https://suricata-ids.org/docs/
    Process: 1283 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid (code=exited, status=0/SUCCESS)
   Main PID: 1284 (Suricata-Main)
      Tasks: 1 (limit: 4652)
     Memory: 211.7M
        CPU: 6.132s
     CGroup: /system.slice/suricata.service
             ??1284 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

Oct 04 14:25:49 suricata systemd[1]: Starting suricata.service - Suricata IDS/IDP daemon...
Oct 04 14:25:49 suricata suricata[1283]: 4/10/2023 -- 14:25:49 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
Oct 04 14:25:49 suricata systemd[1]: Started suricata.service - Suricata IDS/IDP daemon.
```

Le processus peut prendre quelques minutes pour terminer l'analyse de toutes les règles. Par conséquent, la vérification de l'état ci-dessus ne constitue pas une indication complète quant à savoir si Suricata est opérationnel et prêt. Vous pouvez surveiller le fichier journal pour cela à l'aide de la commande suivante.

```
$ sudo tail -f /var/log/suricata/suricata.log
```

Si vous voyez la ligne suivante dans le fichier journal, cela signifie que Suricata est en cours d'exécution et prêt à surveiller le trafic réseau. Quittez la commande "tail" en appuyant sur les touches CTRL+C

```
4/10/2023 -- 14:26:12 - <Info> - All AFP capture threads are running.
```

Étape 6 - Test des règles Suricata

Nous vérifierons si Suricata détecte un trafic suspect. Le guide Suricata recommande de tester la règle ET Open numéro 2100498 à l'aide de la commande suivante.

```
$ curl http://testmynids.org/uid/index.html
```

Vous obtiendrez la réponse suivante.

```
uid=0(root) gid=0(root) groups=0(root)
```

La commande ci-dessus prétend renvoyer la sortie de en utilisant la commande *id* qui peut être exécutée sur un système compromis. Pour tester si Suricata a détecté le trafic, vous devez vérifier le fichier journal numéro de règle spécifié.

```
$ grep 2100498 /var/log/suricata/fast.log
```

Si votre requête utilisait IPv6, vous devriez voir le résultat suivant.

```
10/04/2023-14:26:37.511168  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 2600:9000:23d0:a200:0018:30b3:e400:93a1:80 -> 2a03:b0c0:0002:0 0 Si votre
```

requête utilisait IPv4, vous verriez le résultat suivant.

```
10/04/2023-14:26:37.511168  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 108.158.221.5:80 -> 95.17
```

Suricata enregistre également les événements dans le fichier /var/log/suricata/eve.log au format JSON. Pour lire et interpréter ces règles, vous devez installer jq, ce qui sort du cadre de ce tutoriel.

PARTIE 2

Nous en avons terminé avec la première partie du didacticiel, où nous avons installé Suricata et l'avons testé. La partie suivante consiste à installer la pile ELK et à la configurer pour visualiser Suricata et ses journaux. Deuxième partie de le tutoriel est censé être réalisé sur le deuxième serveur, sauf indication contraire.

Étape 7 - Installer Elasticsearch

La première étape de l'installation d'Elasticsearch consiste à ajouter la clé Elastic GPG à votre serveur.

```
$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg
```

Créez un référentiel pour le package Elasticsearch en créant le fichier /etc/apt/sources.list.d/elastic-7.x.list.

```
$ echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list
```

Mettez à jour la liste des référentiels de votre système.

```
$ sudo apt update
```

Installez Elasticsearch et Kibana.

```
$ sudo apt install elasticsearch
```

Vous obtiendrez le résultat suivant lors de l'installation d'Elasticsearch.

```
-------------------------- Security autoconfiguration information ------------------------------

Authentication and authorization are enabled.
TLS for the transport and HTTP layers is enabled and configured.

The generated password for the elastic built-in superuser is : IuRTjJr+=NqIClxZwKBn

If this node should join an existing cluster, you can reconfigure this with
'/usr/share/elasticsearch/bin/elasticsearch-reconfigure-node --enrollment-token <token-here>'
after creating an enrollment token on your existing cluster.

You can complete the following actions at any time:

Reset the password of the elastic built-in superuser with
'/usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic'.

Generate an enrollment token for Kibana instances with
 '/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana'.

Generate an enrollment token for Elasticsearch nodes with
'/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s node'.

-------------------------------------------------------------------------------------------------
```

Nous utiliserons ces informations plus tard.

Localisez l'adresse IP privée de votre serveur à l'aide de la commande suivante.

```
$ ip -brief address show
lo               UNKNOWN        127.0.0.1/8 ::1/128
eth0             UP             159.223.220.228/20 10.18.0.5/16 2a03:b0c0:2:d0::e0e:c001/64 fe80::841e:feff:fee4:e653/64
eth1             UP             10.133.0.2/16 fe80::d865:d5ff:fe29:b50f/64
```

Notez l'IP privée de votre serveur (10.133.0.2 dans ce cas). Nous l'appellerons your_private_IP. L'adresse IP publique du serveur (159.223.220.228) sera appelée your_public_IP dans le reste du didacticiel. Notez également le nom de l'interface réseau de votre serveur, eth1.

Étape 8 - Configurer Elasticsearch

Elasticsearch stocke sa configuration dans le dépôt /etc/elasticsearch/elasticsearch.yml Ouvrez le fichier pour le modifier.

```
$ sudo nano /etc/elasticsearch/elasticsearch.yml
```

Elasticsearch n'accepte que les connexions locales par défaut. Nous devons le modifier pour que Kibana puisse y accéder via l'adresse IP privée.

Trouver la ligne #network.host: 192.168.0.1 et ajoutez la ligne suivante juste en dessous, comme indiqué ci-dessous.

```
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
#network.host: 192.168.0.1
network.bind_host: ["127.0.0.1", "your_private_IP"]
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
```

Cela garantira qu'Elastic peut toujours accepter les connexions locales tout en étant disponible pour Kibana via l'adresse IP privée.

é et à garantir qu'Elastic est configuré pour s'exécuter sur un seul nœud. Si vous envisagez d'utiliser plusieurs nœuds de recherche Elastic, ignorez les modifications ci-dessous et enregistrez le fichier.

Pour ce faire, ajoutez la ligne suivante à la fin du fichier.

```
. . .
discovery.type: single-node
```

Commentez également la ligne suivante en ajoutant un dièse (#) devant elle.

```
#cluster.initial_master_nodes: ["kibana"]
```

Une fois que vous avez terminé, enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité.

Configurer le pare-feu

Ajoutez les règles de pare-feu appropriées pour Elasticsearch afin qu'il soit accessible via le réseau privé.

```
$ sudo ufw allow in on eth1
$ sudo ufw allow out on eth1
```

Assurez-vous de choisir le nom de l'interface dans la première commande comme celui que vous avez obtenu à l'étape 7.

Démarrer Elasticsearch

Rechargez le démon de service.

```
$ sudo systemctl daemon-reload
```

Activez le service Elasticsearch.

```
$ sudo systemctl enable elasticsearch
```

Maintenant que vous avez configuré Elasticsearch, il est temps de démarrer le service.

```
$ sudo systemctl start elasticsearch
```

Vérifiez l'état du service.

```
$ sudo systemctl status elasticsearch
? elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; preset: enabled)
     Active: active (running) since Wed 2023-10-04 14:30:55 UTC; 8s ago
       Docs: https://www.elastic.co
   Main PID: 1731 (java)
      Tasks: 71 (limit: 4652)
     Memory: 2.3G
        CPU: 44.355s
     CGroup: /system.slice/elasticsearch.service
```

Créer des mots de passe Elasticsearch

Après avoir activé le paramètre de sécurité d'Elasticsearch, l'étape suivante consiste à générer le mot de passe pour le superutilisateur Elasticsearch. Le mot de passe par défaut a été fourni lors de l'installation, ce qui vous pouvez l'utiliser mais il est recommandé de le modifier.

Exécutez la commande suivante pour réinitialiser le mot de passe Elasticsearch. Choisissez un mot de passe fort.

```
$ sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic -i
This tool will reset the password of the [elastic] user.
You will be prompted to enter the password.
Please confirm that you would like to continue [y/N]y


Enter password for [elastic]: <ENTER-PASSWORD>
Re-enter password for [elastic]: <CONFIRM-PASSWORD>
Password for the [elastic] user successfully reset.
```

Maintenant, testons si Elasticsearch répond aux requêtes.

```
$ sudo curl --cacert /etc/elasticsearch/certs/http_ca.crt -u elastic https://localhost:9200
Enter host password for user 'elastic':
{
  "name" : "kibana",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "KGYx4poLSxKhPyOlYrMq1g",
  "version" : {
    "number" : "8.10.2",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "6d20dd8ce62365be9b1aca96427de4622e970e9e",
    "build_date" : "2023-09-19T08:16:24.564900370Z",
    "build_snapshot" : false,
    "lucene_version" : "9.7.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Cela confirme qu'Elasticsearch est entièrement fonctionnel et fonctionne correctement.

Étape 9 - Installer et configurer Kibana

Installez Kibana.

```
$ sudo apt install kibana
```

La première étape de la configuration de Kibana consiste à activer la fonction de sécurité xpack en générant des clés secrètes. Kibana utilise ces clés secrètes pour stocker les données dans Elasticsearch. L'utilitaire permettant de générer des clés secrètes est accessible depuis le répertoire /usr/share/kibana/bin.

```
$ sudo /usr/share/kibana/bin/kibana-encryption-keys generate -q --force                    --forcer
```

Le drapeau -q supprime les instructions de commande et le drapeau --forcer garantit que de nouveaux secrets sont générés. Vous recevrez une sortie comme celle-ci.

```
xpack.encryptedSavedObjects.encryptionKey:248eb61d444215a6e710f6d1d53cd803
xpack.reporting.encryptionKey: aecd17bf4f82953739a9e2a9fcad1891
xpack.security.encryptionKey: 2d733ae5f8ed5f15efd75c6d08373f36
```

Copiez la sortie. Ouvrez le fichier de configuration de Kibana à l'adresse `/etc/kibana/kibana.yml` pour le modifier

```
$ sudo nano /etc/kibana/kibana.yml
```

Collez le code de la commande précédente à la fin du fichier.

```
. . .
# Maximum number of documents loaded by each shard to generate autocomplete suggestions.
# This value must be a whole number greater than zero. Defaults to 100_000
#unifiedSearch.autocomplete.valueSuggestions.terminateAfter: 100000

xpack.encryptedSavedObjects.encryptionKey: 3ff21c6daf52ab73e932576c2e981711
xpack.reporting.encryptionKey: edf9c3863ae339bfbd48c713efebcfe9
xpack.security.encryptionKey: 7841fd0c4097987a16c215d9429daec1
```

Copiez le fichier de certificat CA /etc/elasticsearch/certs/http_ca.crt au /etc/kibana annuaire.

```
$ sudo cp /etc/elasticsearch/certs/http_ca.crt /etc/kibana/
```

Configurer l'hôte Kibana

Kibana doit être configuré pour qu'il soit accessible sur l'adresse IP privée du serveur. Trouver la ligne #serveur.host : "localhost" dans le fichier et ajoutez la ligne suivante juste en dessous, comme indiqué.

```
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
#server.host: "localhost"
server.host: "your_private_IP"
```

Désactiver la télémétrie

Kibana renvoie les données à ses serveurs par défaut. Cela peut affecter les performances et constitue également un risque pour la confidentialité. Par conséquent, vous devez désactiver la télémétrie. Ajoutez le code suivant à la fin du fichier pour désactiver la télémétrie. Le premier paramètre désactive la télémétrie et le deuxième paramètre interdit l'écrasement du premier paramètre de la section Paramètres avancés de Kibana.

```
telemetry.optIn: false
telemetry.allowChangingOptInStatus: false
```

Une fois que vous avez terminé, enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité.

Configurer SSL

Trouver la variable elasticsearch.ssl.certificateAuthorities et décommentez-le et modifiez sa valeur comme indiqué ci-dessous.

```
elasticsearch.ssl.certificateAuthorities: [ "/etc/kibana/http_ca.crt" ]
```

Configurer l'accès à Kibana

L'étape suivante consiste à générer un jeton d'inscription que nous utiliserons plus tard pour nous connecter à l'interface Web de Kibana.

```
$ sudo /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana
eyJ2ZXIiOiI4LjEwLjIiLCJhZHIiOlsiMTU5LjIyMy4yMjAuMjI4OjkyMDAiXSwiZmdyIjoiOGMyYTcyYmUwMDg5NTJlOGMxMWUwNDgzYjE2OTcwOTMxZWZlNzYyMDAwNzhhOGMwNTNmNWU0NGJiY2U4NzcwMSIsImtleSI6IlQ5eE0tNG9CUWZDaGdaakUwbFAzOk9QTU5uWVRnUWppU3lvU0h u
```

Démarrer Kibana

Maintenant que vous avez configuré l'accès sécurisé et la mise en réseau pour Kibana, démarrez et activez le processus.

```
$ sudo systemctl enable kibana --now
```

Vérifiez l'état pour voir s'il est en cours d'exécution.

```
$ sudo systemctl status kibana
? kibana.service - Kibana
     Loaded: loaded (/lib/systemd/system/kibana.service; enabled; preset: enabled)
     Active: active (running) since Wed 2023-10-04 15:27:28 UTC; 9s ago
       Docs: https://www.elastic.co
   Main PID: 2686 (node)
      Tasks: 11 (limit: 4652)
     Memory: 241.5M
        CPU: 9.902s
     CGroup: /system.slice/kibana.service
             ??2686 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist

Oct 04 15:27:28 kibana systemd[1]: Started kibana.service - Kibana.
```

Étape 10 - Installer et configurer Filebeat

Il est important de noter que nous allons installer Filebeat sur le serveur Suricata. Alors revenez-y et ajoutez la clé Elastic GPG pour commencer.

```
$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg
```

Créez le référentiel élastique.

```
$ echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list
```

Mettez à jour la liste des référentiels système.

```
$ sudo apt update
```

Enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité.

Installez Filebeat.

```
$ sudo apt install filebeat
```

Avant de configurer Filebeat, nous devons copier le http_ca.crt fichier du serveur Elasticsearch vers le serveur Filebeat. Exécutez la commande suivante sur le serveur Filebeat.

```
$ scp username@your_public_ip:/etc/elasticsearch/certs/http_ca.crt /etc/filebeat
```

Filebeat stocke sa configuration dans le /etc/filebeat/filebeat.yml déposer. Ouvrez-le pour le modifier.

```
$ sudo nano /etc/filebeat/filebeat.yml
```

La première chose à faire est de le connecter au tableau de bord de Kibana. Trouver la ligne #host: "localhost:5601" dans la section Kibana et ajoutez les lignes suivantes juste en dessous, comme indiqué.

```
. . .
# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  #host: "localhost:5601"
  host: "your_private_IP:5601"
  protocol: "http"
  ssl.enabled: true
  ssl.certificate_authorities: ["/etc/filebeat/http_ca.crt"]

. . .
```

Ensuite, recherchez la section Elasticsearch Output du fichier et modifiez les valeurs du mot osts, protocol, username, et password comme indiqué ci-dessous. Pour le nom d'utilisateur, choisissez elastic comme la valeur, et pour le de passe, utilisez la valeur générée à l'étape 8 de ce didacticiel.

```
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["your_private_IP:9200"]

  # Protocol - either `http` (default) or `https`.
  protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
```

```
  username: "elastic"
  password: "bd1YjfhSa8RC8SMvTIwg"
  ssl.certificate_authorities: ["/etc/filebeat/http_ca.crt"]
  ssl.verification_mode: full

. . .
```

Ajoutez la ligne suivante au bas du fichier.

```
setup.ilm.overwrite: true
```

Une fois que vous avez terminé, enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité. Il faut encore une étape pour s'assurer que Filebeat se connecte à Elasticsearch. Nous devons transmettre les informations SSL d'Elasticsearch à Filebeat pour qu'il puisse se connecter.

Testez la connexion du Filebeat au serveur Elasticsearch. Il vous sera demandé votre mot de passe Elasticsearch.

```
$ curl -v --cacert /etc/filebeat/http_ca.crt https://your_private_ip:9200 -u elastic
```

Vous obtiendrez le résultat suivant.

```
Enter host password for user 'elastic':
*   Trying 10.133.0.2:9200...
* Connected to 10.133.0.2 (10.133.0.2) port 9200 (#0)
* ALPN: offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
*  CAfile: /etc/filebeat/http_ca.crt
*  CApath: /etc/ssl/certs
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384
* ALPN: server did not agree on a protocol. Uses default.
* Server certificate:
*  subject: CN=kibana
*  start date: Oct  4 14:28:33 2023 GMT
*  expire date: Oct  3 14:28:33 2025 GMT
*  subjectAltName: host "10.133.0.2" matched cert's IP address!
*  issuer: CN=Elasticsearch security auto-configuration HTTP CA
*  SSL certificate verify ok.
* using HTTP/1.x
* Server auth using Basic with user 'elastic'
> GET / HTTP/1.1
> Host: 10.133.0.2:9200
> Authorization: Basic ZWxhc3RpYzpsaWZlc3Vja3M2NjIwMDI=
> User-Agent: curl/7.88.1
> Accept: */*
>
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
< HTTP/1.1 200 OK
< X-elastic-product: Elasticsearch
< content-type: application/json
< content-length: 530
<
{
  "name" : "kibana",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "KGYx4poLSxKhPyOlYrMq1g",
  "version" : {
    "number" : "8.10.2",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "6d20dd8ce62365be9b1aca96427de4622e970e9e",
    "build_date" : "2023-09-19T08:16:24.564900370Z",
    "build_snapshot" : false,
    "lucene_version" : "9.7.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
* Connection #0 to host 10.133.0.2 left intact
```

Ensuite, activez le module Suricata intégré de Filebeat.

```
$ sudo filebeat modules enable suricata
```

Ouvrez le /etc/filebeat/modules.d/suricata.yml fichier à éditer.

```
$ sudo nano /etc/filebeat/modules.d/suricata.yml
```

Modifiez le fichier comme indiqué ci-dessous. Vous devez changer la valeur de activé variable à vrai . Décommentez également la variable var.chemins et définissez sa valeur comme indiqué.

```
# Module: suricata
# Docs: https://www.elastic.co/guide/en/beats/filebeat/8.10/filebeat-module-suricata.html

- module: suricata
  # All logs
  eve:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: ["/var/log/suricata/eve.json"]
```

Une fois que vous avez terminé, enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité.

La dernière étape de la configuration de Filebeat consiste à charger les tableaux de bord et les pipelines SIEM dans Elasticsearch à l'aide de l'outil configuration du battement de fichiers commande.

```
$ sudo filebeat setup
```

L'exécution de la commande peut prendre quelques minutes. Une fois terminé, vous devriez recevoir le résultat suivant.

```
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Loaded Ingest pipelines
```

Démarrez le service Filebeat.

```
$ sudo systemctl start filebeat
```

Vérifiez l'état du service.

```
$ sudo systemctl status filebeat
```

Étape 11 - Accéder au tableau de bord Kibana

Étant donné que KIbana est configuré pour accéder à Elasticsearch uniquement via son adresse IP privée, vous disposez de deux options pour y accéder. La première méthode consiste à utiliser un tunnel SSH vers le serveur de recherche Elastic depuis votre PC. Cela transférera le port 5601 de votre PC vers l'adresse IP privée du serveur, et vous pourrez accéder à Kibana depuis votre PC et y accéder depuis n'importe où ailleurs. http://localhost:5601 . Mais cette méthode signifie que vous ne le ferez pas

L'autre option consiste à installer Nginx sur votre serveur Suricata et à l'utiliser comme proxy inverse pour accéder au serveur d'Elasticsearch via son adresse IP privée. Nous discuterons des deux manières. Tu peux choisir de toute façon en fonction de vos besoins.

Utilisation du tunnel local SSH

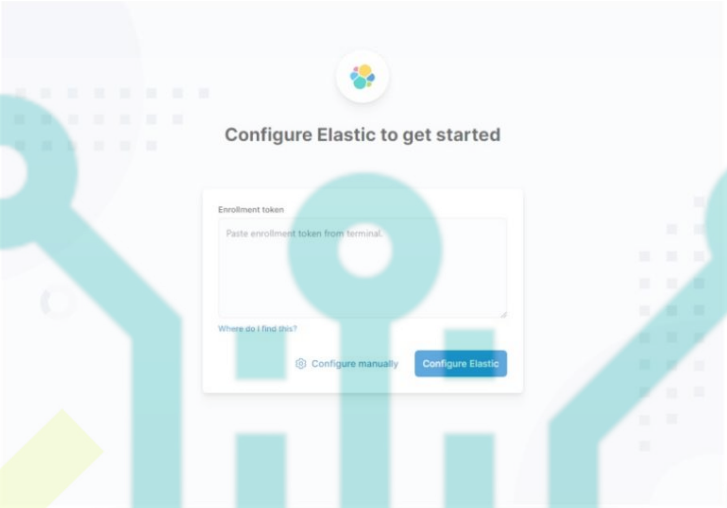
Si vous utilisez Windows 10 ou Windows 11, vous pouvez exécuter SSH LocalTunnel à partir de votre Windows Powershell. Sous Linux ou macOS, vous pouvez utiliser le terminal. Vous devrez probablement configurez l'accès SSH si vous ne l'avez pas déjà fait.

Exécutez la commande suivante dans le terminal de votre ordinateur pour créer le tunnel SSH.

```
$ ssh -L 5601:your_private_IP:5601 navjot@your_public_IP -N
```

- commande `-L` L'indicateur fait référence au tunnel SSH local, qui transfère le trafic du port de votre PC vers le serveur.
- commande `private_IP:5601` est l'adresse IP vers laquelle votre trafic est redirigé sur le serveur. Dans ce cas, remplacez-le par l'adresse IP privée de votre serveur Elasticsearch. est l'adresse IP publique
- commande `your_public_IP` du serveur Elasticsearch, utilisée pour ouvrir une connexion SSH. L'indicateur indique à OpenSSH de n'exécuter
- commande `-N` aucune commande mais de maintenir la connexion active tant que le tunnel est exécuté.

Maintenant que le tunnel est ouvert, vous pouvez accéder à Kibana en ouvrant l'URL `http://localhost:5601` sur le navigateur de votre PC. Vous obtiendrez l'écran suivant.



Vous devrez maintenir la commande en cours d'exécution aussi longtemps que vous aurez besoin d'accéder à Kibana. Appuyez sur Ctrl + C dans votre terminal pour fermer le tunnel.

Utiliser le proxy inverse Nginx

Cette méthode est la mieux adaptée si vous souhaitez accéder au tableau de bord depuis n'importe où dans le monde.

Configurer le pare-feu

Avant de continuer, vous devez ouvrir les ports HTTP et HTTPS dans le pare-feu.

```
$ sudo ufw allow http
$ sudo ufw allow https
```

Installer Nginx

Debian 12 est livré avec une ancienne version de Nginx. Pour installer la dernière version, vous devez télécharger le référentiel officiel Nginx.

Importez la clé de signature de Nginx.

```
$ curl https://nginx.org/keys/nginx_signing.key | gpg --dearmor \ |
 sudo tee /usr/share/keyrings/nginx-archive-keyring.gpg >/dev/null
```

Ajoutez le référentiel pour la version stable de Nginx.

```
$ echo "deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg] \
http://nginx.org/packages/debian `lsb_release -cs` nginx" \
    | sudo tee /etc/apt/sources.list.d/nginx.list
```

Mettez à jour les référentiels système.

```
$ sudo apt update
```

Installez Nginx.

```
$ sudo apt install nginx
```

Vérifiez l'installation. Puisque nous sommes sur Debian, le `sudo` dans le commandement est essentiel.

```
$ sudo nginx -v
nginx version: nginx/1.24.0
```

Démarrez le serveur Nginx.

```
$ sudo systemctl start nginx
```

Installer et configurer SSL

La première étape consiste à installer le certificat SSL Let's Encrypt. Nous devons installer Certbot pour générer le certificat SSL. Vous pouvez soit installer Certbot à l'aide du référentiel Debian, soit récupérer la dernière version version à l'aide de l'outil Snapd. Nous utiliserons la version Snapd.

Debian 12 n'est pas fourni avec Snapd installé. Installez le package Snapd.

```
$ sudo apt install snapd
```

Exécutez les commandes suivantes pour vous assurer que votre version de Snapd est à jour.

```
$ sudo snap install core && sudo snap refresh core
```

Installez Certbot.

```
$ sudo snap install --classic certbot
```

Utilisez la commande suivante pour vous assurer que la commande Certbot peut être exécutée en créant un lien symbolique vers le `/usr/bin` annuaire.

```
$ sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

Confirmez l'installation de Certbot.

```
$ certbot --version
certbot 2.7.0
```

Générer le certificat SSL pour le domaine `kibana.exemple.com` .

```
$ sudo certbot certonly --nginx --agree-tos --no-eff-email --staple-ocsp --preferred-challenges http -m name@example.com -d kibana.example.com
```

La commande ci-dessus téléchargera un certificat sur le `/etc/letsencrypt/live/kibana.example.com` répertoire sur votre serveur.

Générez un certificat de groupe Diffie-Hellman .

```
$ sudo openssl dhparam -dsaparam -out /etc/ssl/certs/dhparam.pem 4096
```

Pour vérifier si le renouvellement SSL fonctionne correctement, effectuez un essai à sec du processus.

```
$ sudo certbot renew --dry-run
```

Si vous ne voyez aucune erreur, vous êtes prêt. Votre certificat se renouvellera automatiquement.

Configurer Nginx

Créez et ouvrez le fichier de configuration Nginx pour Kibana.

```
$ sudo nano /etc/nginx/conf.d/kibana.conf
```

Collez-y le code suivant. Remplacez l'adresse IP par l'adresse IP privée de votre serveur Elasticsearch.

```
server {
        listen 80; listen [::]:80;
        server_name kibana.example.com;
        return 301 https://$host$request_uri;
}

server {
        server_name kibana.example.com;
        charset utf-8;

        listen 443 ssl http2;
        listen [::]:443 ssl http2;

        access_log /var/log/nginx/kibana.access.log;
        error_log /var/log/nginx/kibana.error.log;

        ssl_certificate /etc/letsencrypt/live/kibana.example.com/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/kibana.example.com/privkey.pem;
        ssl_trusted_certificate /etc/letsencrypt/live/kibana.example.com/chain.pem;
        ssl_session_timeout 1d;
        ssl_session_cache shared:MozSSL:10m;
        ssl_session_tickets off;

           ssl_protocols TLSv1.2 TLSv1.3;
           ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256

              resolver 8.8.8.8;

        ssl_stapling on;
        ssl_stapling_verify on;
        ssl_dhparam /etc/ssl/certs/dhparam.pem;

        location / {
                proxy_pass http://your_private_IP:5601;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
        }
}
```

Enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité.

Ouvrir le fichier `/etc/nginx/nginx.conf`

```
$ sudo nano /etc/nginx/nginx.conf
```

Ajoutez la ligne suivante avant la ligne `include /etc/nginx/conf.d/*.conf;`.

```
include /etc/nginx/conf.d/*.conf;.
```

Enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité.

Vérifiez la configuration.

```
$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Redémarrez le service Nginx.

```
$ sudo systemctl restart nginx
```

Votre tableau de bord Kibana doit être accessible via l'URL `https://kibana.example.com` depuis n'importe où vous voulez.

Étape 12 - Gestion des tableaux de bord Kibana

Avant de poursuivre la gestion des tableaux de bord, vous devez ajouter le champ URL de base dans la configuration de Kibana.

Ouvrez le fichier de configuration de Kibana.

```
$ sudo nano /etc/kibana/kibana.yml
```

Trouver la ligne commentée `#server.publicBaseUrl: ""` et modifiez-le comme suit en supprimant le hachage devant lui.

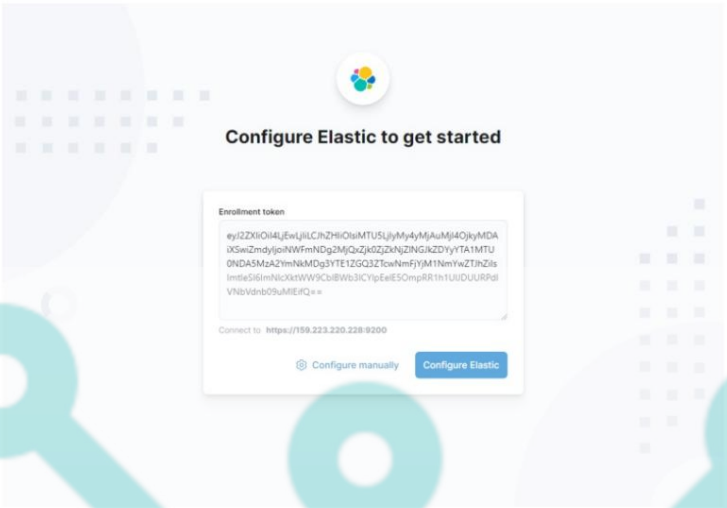
```
server.publicBaseUrl: "https://kibana.example.com"
```

Enregistrez le fichier en appuyant sur Ctrl + X et en entrant Y lorsque vous y êtes invité.

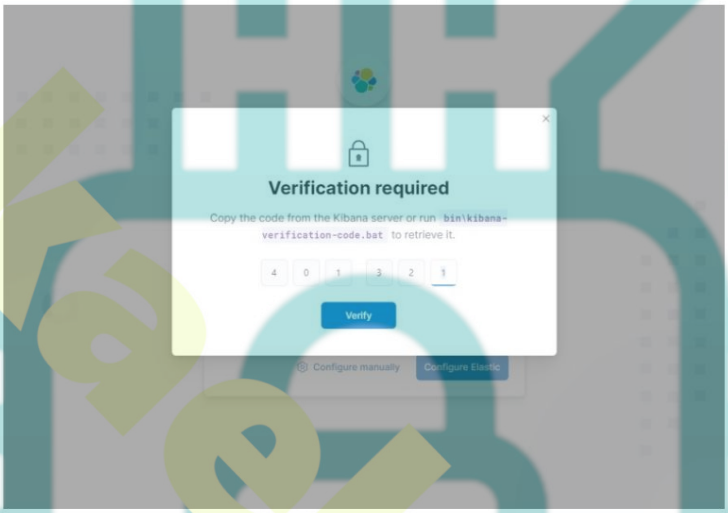
Redémarrez le service Kibana.

```
$ sudo systemctl restart kibana
```

Attendez quelques minutes et chargez l'URL `https://kibana.example.com` dans votre navigateur. Vous obtiendrez le champ du jeton d'inscription. Remplissez le jeton d'inscription que vous avez généré à l'étape 9.



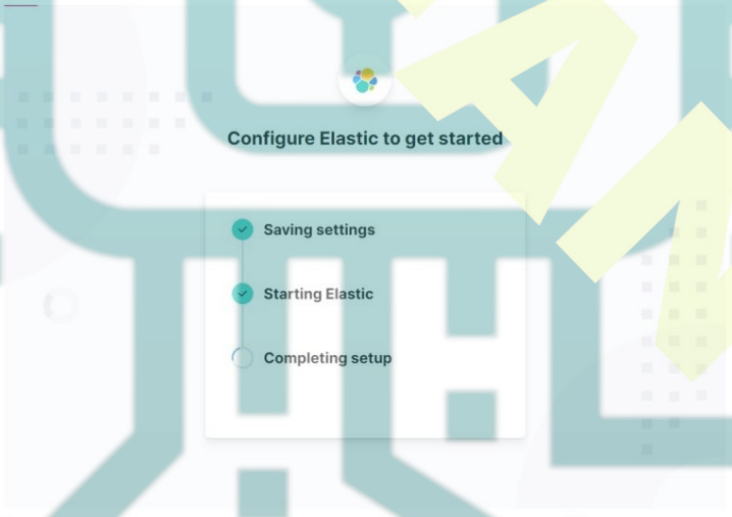
Cliquez sur le bouton Configurer Elastic pour continuer. Ensuite, il vous sera demandé le code de vérification.



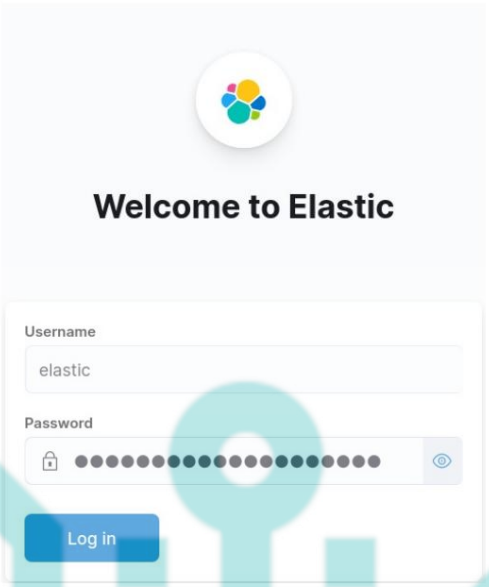
Revenez au terminal Elasticsearch et exécutez la commande suivante pour générer le code. Entrez ce code sur la page et cliquez sur le bouton Vérifier pour continuer.

```
$ sudo /usr/local/share/kibana/bin/kibana-verification-code
```

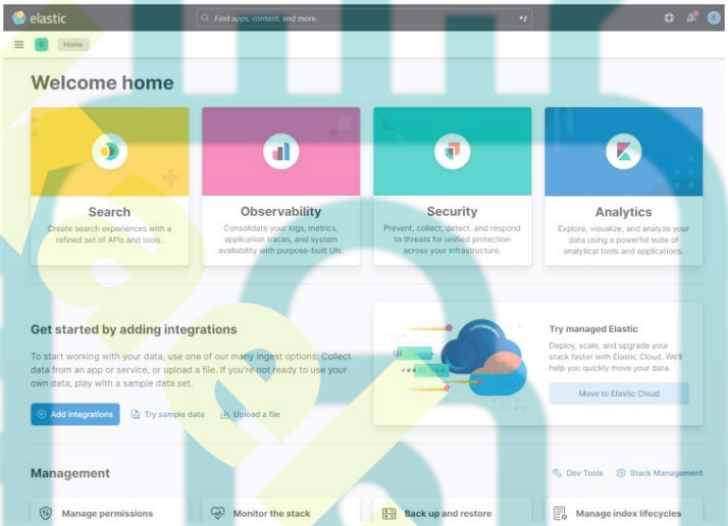
Ensuite, attendez la fin de la configuration d'Elastic. Cela prendra plusieurs minutes.



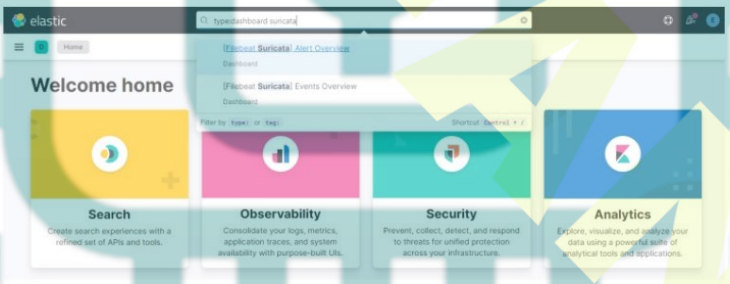
Ensuite, vous serez redirigé vers l'écran de connexion.



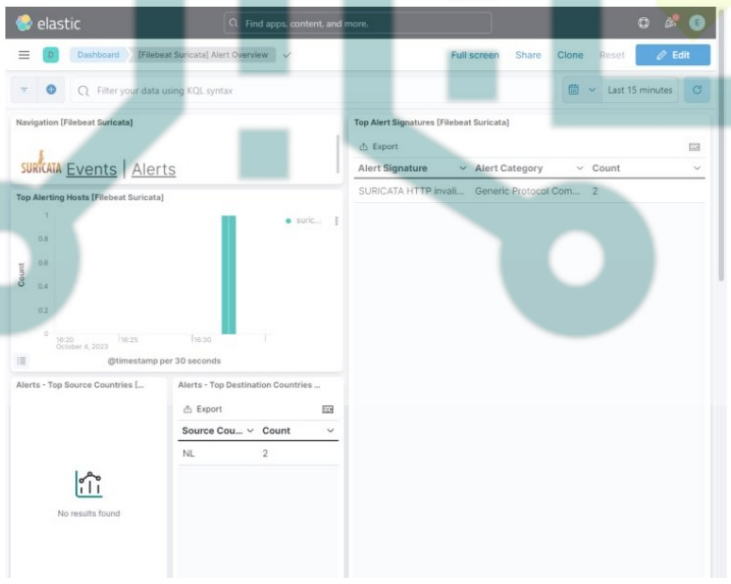
Connectez-vous avec le nom d'utilisateur `elastic` et le mot de passe que vous avez généré auparavant et vous obtiendrez l'écran suivant.



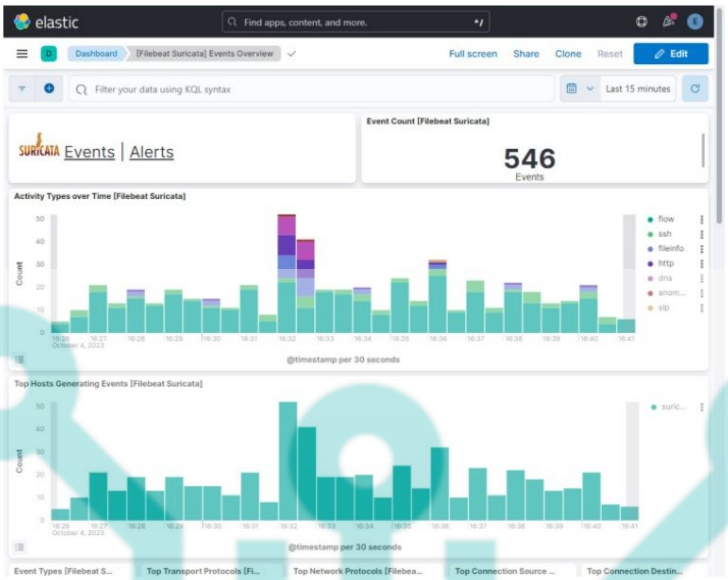
Type `type:data suricata` dans la zone de recherche en haut pour localiser les informations de Suricata.



Cliquez sur le premier résultat **([Filebeat Suricata] Alert Overview)** et vous obtiendrez un écran similaire au suivant. Par défaut, il affiche les entrées des 15 dernières minutes uniquement, mais nous affichons sur une période plus longue pour afficher plus de données pour le didacticiel.

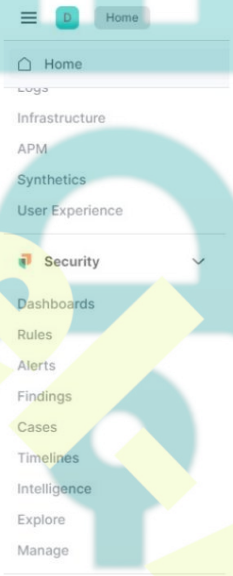


Cliquez sur le bouton Événements pour afficher tous les événements enregistrés.

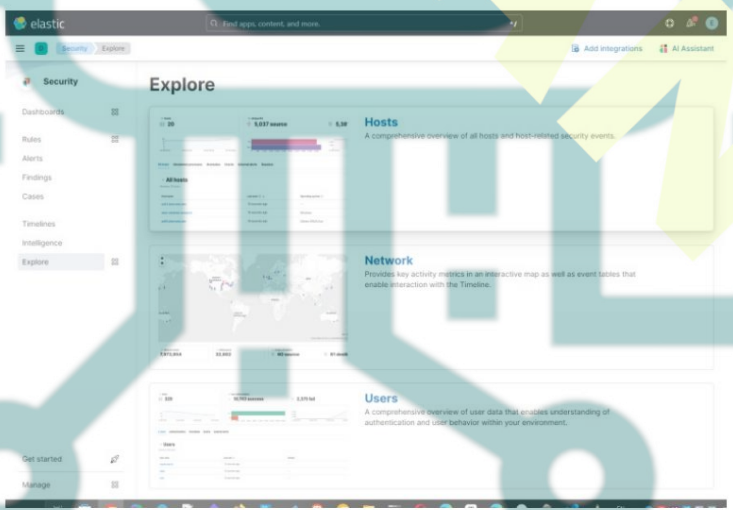


En faisant défiler les pages d'événements et d'alertes, vous pouvez identifier chaque événement et alerte par le type de protocole, les ports source et de destination et l'adresse IP de la source. Vous pouvez également afficher les pays d'où provient le trafic.

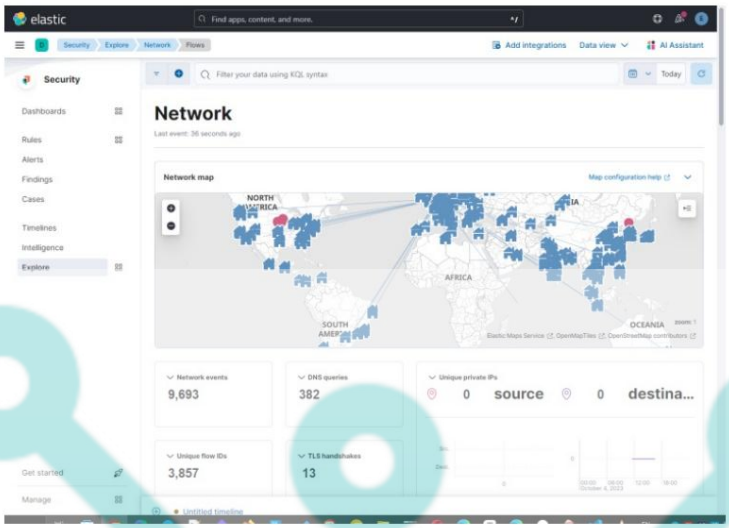
Vous pouvez utiliser Kibana et Filebeat pour accéder et générer d'autres types de tableaux de bord. L'un des tableaux de bord intégrés utiles que vous pouvez utiliser immédiatement est le tableau de bord de sécurité. Cliquez sur le menu Sécurité >> Explorer dans le menu hamburger de gauche.



Sur la page suivante, sélectionnez l' option Réseau pour ouvrir le tableau de bord associé.



En cliquant sur l' option Réseau , vous obtiendrez l'écran suivant.



Vous pouvez ajouter plus de tableaux de bord comme Nginx en activant et en configurant les modules Filebeat intégrés.

Conclusion

Ceci conclut le didacticiel d'installation et de configuration de Suricata IDS avec Elastic Stack sur un serveur Debian 12. Vous avez également configuré Nginx comme proxy inverse pour accéder aux tableaux de bord Kibana en externe. Si vous avez des questions, postez-les dans les commentaires ci-dessous.
