

TRABALLO TUTELADO AII
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN ENXEÑARÍA DE COMPUTADORES

Guía de instalación de Elastiflow

Estudiante 1: Mauro de los Santos Nodar

Estudiante 2: Juan González Iglesias

A Coruña, abril de 2020.

Índice Xeral

1	Contextualización	1
2	Pasos previos	3
3	Instalación e Configuración	5
3.1	ElasticSearch	5
3.2	Logstash	6
3.2.1	JVM Heap	6
3.2.2	Instalar e Actualizar os plugins de Logstash	7
3.2.3	Copiar os arquivos do Git de Elastiflow ao path de configuración de Logstash	7
3.2.4	Configuración das variables de entorno	7
3.2.5	Engadir o pipeline de Elastiflow	7
3.2.6	Configurando a entrada (<i>inputs</i>) a Logstash	8
3.2.7	Configurando a saída (<i>output</i>) de Logstash	8
3.3	Kibana	9
3.4	Últimos pasos	10
4	Configuración dos Exportadores	13
4.1	Instalación	13
4.2	Tshark	13
4.3	Softflowd	14
4.4	Yaf	14
5	Visualización dos datos	15
6	Conclusionés	19
A	Posibles erros	23

B	Próximos pasos	25
	Bibliografía	27

Índice de Figuras

3.1	Comprobación da execución correcta de ElasticSearch	6
3.2	Configuración de parámetros en Kibana (1)	9
3.3	Configuración de parámetros en Kibana (2)	10
3.4	Comprobación dos portos de Elastiflow (1)	10
3.5	Comprobación dos portos de Elastiflow (2)	10
3.6	Comprobación dos portos de Elastiflow (3)	11
5.1	Discover Kibana	15
5.2	Packet example Kibana	16
5.3	Dashboard flows	17
5.4	Dashboard overview	17
5.5	Dashboard traffic	17
5.6	Dashboard geoip	18
5.7	Dashboard Flow Records	18
B.1	Configuración DNS	25
B.2	Configuración da Application Identification	26

Contextualización

Expónse no seguinte documento unha guía completa para a instalación e configuración de **Elastiflow**, ferramenta que proporciona recopilación e visualización de datos de fluxo de rede (admitindo *Netflow*, *sFlow* e *IPFIX*) utilizando o coñecido **Elastic Stack**. **ELK Stack** incorpora as ferramentas *Elasticsearch* o cal é un motor de búsqueda, *Logstash* que é un pipeline encargado do procesamento dos datos e *Kibana* que permite aos usuarios visualizar e analizar os datos obtidos.

Toda a guía foi probada nunha máquina virtual sobre VirtualBox 6.0.12, cun **Kali Linux 2019.3**, con 4 GB de RAM, 2 procesadores e 50 GB de disco.

Ao non dispor dun *firewall* físico empregamos un **exportador** para xerar os fluxos de tráfico que recibirá o noso colector a partir dun arquivo *.pcap*. Desta maneira simulamos o comportamento dun *firewall/router*.

Pasos previos

Os pasos previos antes de comezar coa instalación e configuración de *Elastiflow* consisten principalmente en:

- Comprobación dos requisitos de instalación, que podemos ver nas seguintes *URLs*:
 - Compatibilidade con *Elastic Stack* en [1]
 - Compatibilidade cos diferentes **Sistemas Operativos** en [2]
 - Compatibilidade co **JDK** en [3]
- Clonado do repositorio **Git** do proxecto de *Elastiflow* en [1]

Instalación e Configuración

Primeiro de todo, instalaremos e configuraremos adecuadamente as ferramentas usadas por *Elastiflow*, as cales son *ElasticSearch*, *Logstash* e *Kibana*, é dicir, o chamado entorno **ELK**.

3.1 ElasticSearch

Executamos os seguintes comandos para a descarga e instalación, no noso caso, da versión **7.6.1**:

```
1 #curl -L -O https://artifacts.elastic.co/downloads/elasticsearch/  
2 elasticsearch-X.Y.Z-amd64.deb  
3  
4 #dpkg -i elasticsearch-X.Y.Z-amd64.deb
```

E o seguinte para a posta en marcha:

```
1 #systemctl start elasticsearch
```

Despois desto, para configuralo correctamente engadiremos dous parámetros ao final do ficheiro */etc/elasticsearch/elasticsearch.yml*:

```
1 indices.query.bool.max_clause_count: 8192  
2 search.max_buckets: 100000
```

Por último, comprobaremos que a ferramenta se está executando da forma correcta facendo unha petición a *localhost* ao porto no que está correndo, neste caso, o 9200:

```
1 #curl http://127.0.0.1:9200
```

Se a saída é da forma seguinte, quere dicir que *ElasticSearch* está operativo da forma correcta:

```

root@kali:~# curl 127.0.0.1:9200
{
  "name" : "kali",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "KRxJtzbsKS4iEz2zUkxL6wg",
  "version" : {
    "number" : "7.4.2",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "2f90bbf7b93631e52bafb59b3b049cb44ec25e96",
    "build_date" : "2019-10-28T20:40:44.881551Z",
    "build_snapshot" : false,
    "lucene_version" : "8.2.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}

```

Figura 3.1: Comprobación da execución correcta de Elasticsearch

3.2 Logstash

Executamos os seguintes comandos para a descarga e instalación, no noso caso, da versión

7.6.1:

```

1 #curl -L -O https://artifacts.elastic.co/downloads/logstash/
2 logstash-X.Y.Z.deb
3
4 #dpkg -i logstash-X.Y.Z.deb

```

Despois disto, imos configuralo da forma correcta para *Elastiflow*.

3.2.1 JVM Heap

No ficheiro `/etc/logstash/jvm.options` cambiaremos o primeiro dos parámetros, cuxo valor predeterminado adoita ser `-Xms2g`, `-Xmx2g` por:

```

1 -Xms4g
2 -Xmx4g

```

3.2.2 Instalar e Actualizar os plugins de Logstash

Para esto, dende o directorio `/usr/share/logstash/bin`, executamos:

```
1 #./logstash-plugin install logstash-codec-sflow
2 #./logstash-plugin update logstash-codec-netflow
3 #./logstash-plugin update logstash-input-udp
4 #./logstash-plugin update logstash-input-tcp
5 #./logstash-plugin update logstash-filter-dns
6 #./logstash-plugin update logstash-filter-geoip
7 #./logstash-plugin update logstash-filter-translate
```

3.2.3 Copiar os arquivos do Git de Elastiflow ao path de configuración de Logstash

Para esto, copiaremos a carpeta `elastiflow/logstash/elastiflow` do Git ao directorio `/etc/-logstash/`:

```
1 #cp -r elastiflow/logstash/elastiflow /etc/logstash/
```

3.2.4 Configuración das variables de entorno

Xa que toda a configuración de **Elastiflow** funciona con variables de entorno, imos copiar as que *Git* nos trae por defecto na carpeta **profile.d**, no ficheiro **elastiflow.sh** ao directorio `/etc/systemd/system/logstash.service.d/` como **elastflow.conf**:

```
1 #mkdir /etc/systemd/system/logstash.service.d
2 #cp profile.d/elastiflow.sh
   /etc/systemd/system/logstash.service.d/elastiflow.conf
```

3.2.5 Engadir o pipeline de Elastiflow

Por defecto, no ficheiro `/etc/logstash/pipelines.yml`, aparécenos un só *pipeline*, chamado *main*, referenciando aos ficheiros de configuración por defecto. Xa que non o imos usar, comentamos as liñas que aparecen e engadimos a continuación as seguintes:

```
1 #- pipeline.id: main
2 # path.config: "/etc/logstash/conf.d/*.conf"
3
4 - pipeline.id: elastiflow
5   path.config: "/etc/logstash/elastiflow/conf.d/*.conf"
```

Á parte, no ficheiro `/etc/logstash/logstash.yml`, cambiaremos o campo `pipeline.id` por `elastiflow`, ou ben polo nome que lle decidiramos poñer ao `pipeline` engadido anteriormente, á vez que comentamos o `id` de `main`, que previamente tamén comentamos no ficheiro `pipelines.yml`.

```
1 #pipeline.id: main
2 pipeline.id: elastiflow
```

3.2.6 Configurando a entrada (*inputs*) a Logstash

Simplemente deixaremos todo por defecto, a menos que queiramos activar a recepción de tráfico referente a **IPv6**, que teremos que comentar os arquivos `.disabled` do directorio `/etc/logstash/elastiflow/conf.d`

```
1 #mv 10_input_ipfix_ipv6.logstash.conf.disabled
   10_input_ipfix_ipv6.logstash.conf
2 #mv 10_input_netflow_ipv6.logstash.conf.disabled
   10_input_netflow_ipv6.logstash.conf
3 #mv 10_input_sflow_ipv6.logstash.conf.disabled
   10_input_sflow_ipv6.logstash.conf
```

Como extra, para mellorar o rendemento, cambiaremos as variables de entorno tanto de `workers` como de `queue_size`, incrementándoas respectivamente de 2 a 4 e de 2000 a 4096. Isto facémolo editando o ficheiro `/etc/systemd/system/logstash.service.d/elastiflow.conf` ou individualmente cada ficheiro de configuración en `/etc/logstash/elastiflow/conf.d`, tanto o de *Netflow*, como o de *sFlow* e *IPFIX*.

3.2.7 Configurando a saída (*output*) de Logstash

Deixaremos os valores por defecto, xa que son a dirección e o porto onde está correndo **Elasticsearch**. Engadir, que se queremos usar un *cluster* de nodos de *Elasticsearch* en vez de un só, deberíamos cambiarlle o nome aos seguintes arquivos no directorio `/etc/logstash/elastiflow/conf.d`:

```
1 #mv 30_output_10_single.logstash.conf
   30_output_10_single.logstash.conf.disabled
2 #mv 30_output_20_multi.logstash.conf
   30_output_20_multi.logstash.conf.disabled
```

Esto é recomendable cando nos movamos en cifras maiores a **2500 flows/sec**.

Por último, poremos en marcha *Logstash*, executando no directorio `/usr/share/logstash/bin` o seguinte comando:

```
1 #./logstash --path.settings /etc/logstash --config.reload.automatic
```

Indicámoslle o *path* onde se atopan todos os ficheiros de configuración editados previamente, así como a opción de que se recargue automaticamente cando a configuración sexa modificada, para evitar ter que reiniciar o servico.

3.3 Kibana

Para a súa instalación executaremos os seguintes comandos, usaremos a versión **7.6.1**:

```
1 #curl -L -O https://artifacts.elastic.co/downloads/kibana/  
2 kibana-X.Y.Z-linux-x86_64.tar.gz  
3  
4 #tar xzvf kibana-X.Y.Z-linux-x86_64.tar.gz
```

Para activar Kibana, bastará con executar o seguinte comando dende o directorio **kibana-7.6.1-linux-x86_64**:

```
1 #./bin/kibana
```

Como punto extra comentar que se estamos executando este servico dende *root*, debere-mos indicarlle o *flag* **-allow-root**.

En canto á configuración de *Kibana*, levarémola a cabo dende a súa interfaz gráfica. Polo que despois de executar os comandos anteriores, accederemos á dirección ***http://localhost:5601*** dende un navegador (*Firefox*, no noso caso) e procederemos ao seguinte:

- **Importación das plantillas:** Dende *Management*, *Saved Objects*, *clickaremos* en *Import*, e importaremos o arquivo do *Git* ***elastiflow.kibana.<version>.json***, contido na carpeta *kibana*.

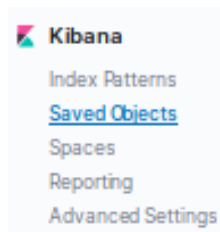


Figura 3.2: Configuración de parámetros en Kibana (1)

- **Configuración de parámetros:** Dende *Management, Advanced Settings*, procederemos a verificar que os seguintes tres parámetros teñen os valores indicados:

Advanced Setting	Value	Why make the change?
doc_table:highlight	false	There is a pretty big query performance penalty that comes with using the highlighting feature. As it isn't very useful for this use-case, it is better to just turn it off.
filters:pinnedByDefault	true	Pinning a filter will allow it to persist when you are changing dashboards. This is very useful when drill-down into something of interest and you want to change dashboards for a different perspective of the same data. This is the first setting I change whenever I am working with Kibana.
state:storeInSessionStorage	true	Kibana URLs can get pretty large. Especially when working with Vega visualizations. This will likely result in error messages for users of Internet Explorer. Using in-session storage will fix this issue for these users.

Figura 3.3: Configuración de parámetros en Kibana (2)

3.4 Últimos pasos

Unha vez feito esto, temos o entorno **ELK** instalado, ben configurado e operativo. **Logstash** está correndo e escoitando nos portos 2055, 6343 e 4739 (de **Netflow**, **sFlow** e **IPFIX** respectivamente). Cando reciba os datos de entrada (*inputs*), reenviarallos a **ElasticSearch** (é dicir, éste será o *output* de **Logstash**), o cal está escoitando na dirección *localhost:9200*. Por último, **ElasticSearch** enviará os datos procesados e listos para ser visualizados de forma correcta a **Kibana**, o cal está correndo e escoitando na dirección *localhost:5601*.

É boa práctica antes de seguir con esta guía, comprobar co comando **lsof -ni** que temos todos os portos mencionados anteriormente escoitando como podemos ver nas seguintes imaxes:

```
java      28611      root    95u  IPv4  967809      0t0  UDP *:2055
java      28611      root    96u  IPv4  967179      0t0  UDP *:4739
java      28611      root    97u  IPv4  967180      0t0  UDP *:6343
```

Figura 3.4: Comprobación dos portos de Elastiflow (1)

```
node      29467      root    27u  IPv4  913237      0t0  TCP 127.0.0.1:5601 (LISTEN)
```

Figura 3.5: Comprobación dos portos de Elastiflow (2)


```
java      8180 elasticsearch  220u  IPv6  62849      0t0  TCP 127.0.0.1:9200 (LISTEN)
```

Figura 3.6: Comprobación dos portos de Elastiflow (3)

Agora ben, o seguinte será enviar os paquetes de datos a través dun exportador.

Configuración dos Exportadores

Neste capítulo procederase á configuración e execución do exportador así como a obtención dun arquivo *.pcap* con distinto tipo de tráfico xerado.

Eliximos dúas ferramentas como exportadores, a primeira é **softflowd** para *NetFlow* e a segunda é **yaf** para *IPFIX*.

Por outra banda para capturar paquetes da rede utilizamos a ferramenta **tshark**.

4.1 Instalación

Para instalar o exportador **softflowd** e o *sniffer* **tshark** bastará con acudir aos repositorios no noso *Kali* da seguinte maneira:

```
1 # apt-get install softflowd
2 # apt-get install tshark
```

Para instalar **yaf** bastará con descargar o ficheiro *.tar.gz* da páxina web oficial [11] da versión que nos elixamos, no noso caso a **2.10.0**. Una vez descargado, descomprimímolos e executamos o instalador.

4.2 Tshark

Eleximos capturar a nosa propia rede durante un longo período de tempo co fin de obter un elevado número de paquetes de distintas tipoloxías, para iso só falta executar o seguinte comando e gardar os resultados nun arquivo específico:

```
1 # tshark -i eth0 -w arquivo.pcap
```

4.3 Softflowd

A utilización deste analizador e exportador de tráfico de rede é moi sinxela. Para a posta en marcha executamos o seguinte comando, onde lle estamos dicindo que envíe á *IP* e ao porto indicados tras o *flag -n*, o tráfico capturado no arquivo tras o *flag -r*:

```
1 # softflowd -n 127.0.0.1:2055 -r arquivo.pcap
```

Tamén temos a opción de exportar directamente a nosa tarxeta de rede, para iso executariamos, onde tras o *flag -i* indicariamos a interfaz de rede a enviar:

```
1 # softflowd -i eth0 -n 127.0.0.1:2055
```

4.4 Yaf

Para empezar a exportar paquetes dun ficheiro *.pcap* executamos o seguinte comando:

```
1 # yaf --uniflow --in arquivo.pcap --out 127.0.0.1 --ipfix-port 2055  
  --ipfix udp
```

Visualización dos datos

Para visualizar os datos obtidos vamos a empregar a interfaz gráfica **kibana** que nos vai permitir recoller, organizar e preparar os distintos datos para fins analíticos.

Dentro da súa interfaz gráfica podemos encontrar o apartado **Discover** o cal vains permitir examinar,filtrar e buscar rexistros nun intervalo específico. Vains amosar una pantalla semellante á seguinte [fig:5.1].

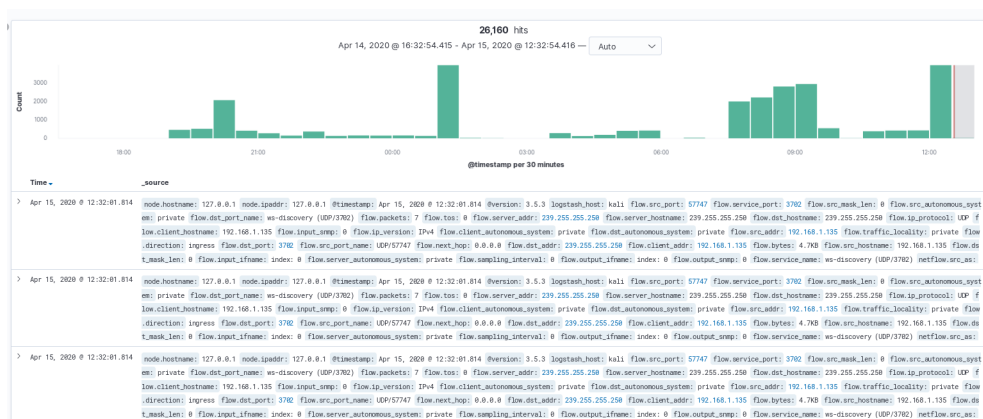


Figura 5.1: Discover Kibana

Como podemos ver aparecenos unha *Search Bar* para buscar campos específicos ou mensaxes completos, tamén un *Time Filter* para filtrar rexistros en base a intervalos de tempo e un *Date Histogram* que é un gráfico de barras que por defecto mostra o recuento de todos os rexistros en función do tempo.

Se facemos *click* nun paquete amosaranos a información asociada a este en forma de táboa ou en formato *JSON* (fig:5.2).

The screenshot shows the Kibana interface for packet analysis. At the top, a list of packets is shown with a selected one. The selected packet's source information is displayed in a JSON-like format:

```

flow.server_geo_location: 37.751,-97.822 flow.tos: 0 flow.sampling_interval: 0
flow.direction: unspecified flow.ip_version: IPv4 flow.src_port: 11889
flow.autonomous_system: Google LLC (15169) flow.src_port_name: UDP/11889
flow.country_code: US flow.service_name: dns (UDP/53) flow.dst_country_code: US
flow.server_country: United States flow.vlan: 0 flow.server_asn: 15169

```

Below this, there are links to 'Expanded document', 'View surrounding documents', and 'View single document'. The 'Expanded document' section shows a table of fields and their values for the selected packet:

Field	Value
@timestamp	Apr 15, 2020 @ 11:30:54.000
@version	3.5.3
_id	xR4yfxEBhtk90MJGS1
_index	elastiflow-3.5.3-2020.04.15
_score	-
_type	_doc
event.host	127.0.0.1
event.type	ipfix
flow.autonomous_system	Google LLC (15169)
flow.client_addr	192.168.89.2
flow.client_autonomous_system	private
flow.client_hostname	192.168.89.2
flow.country	United States

Figura 5.2: Packet example Kibana

A seguinte sección que imos ver é a de **Dashboard**, na cal vamos a poder visualizar distintos tipos de gráficas e cuadros de mando. Temos a opción de utilizar as propias de **ElastiFlow** ou crear as nosas personalizadas. Dentro das primeiras podemos atopar múltiples opcións, algunhas delas son as seguintes:

- O *dashboard Flows* amósanos o intercambio de información entre os distintos clientes e os respectivos servidores [fig:5.3].

- Grazas ao *dashboard Geolp* podemos ver a localización xeográfica de todos os clientes e servidores [fig:5.6].

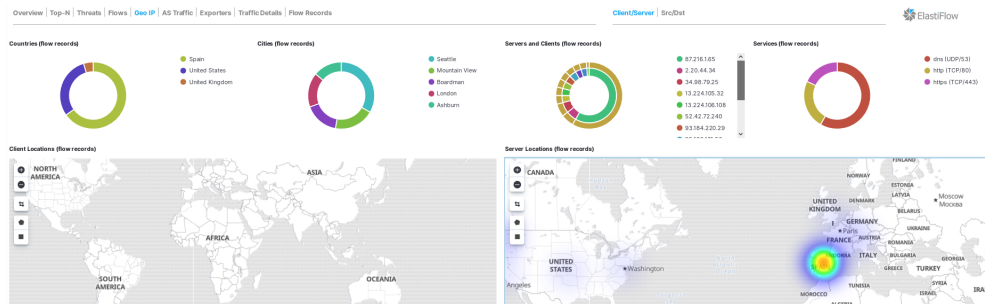


Figura 5.6: Dashboard geolp

- Por último podemos ver un exemplo do *dashboard Flow Records* o cal se asemella a sección *Discover* [fig:5.7].

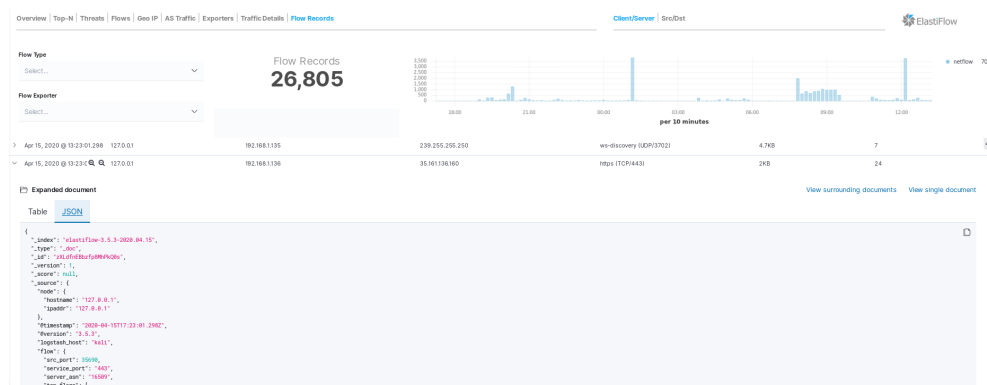


Figura 5.7: Dashboard Flow Records

- A parte destas plantillas, quizais as máis xerais ou importantes, temos máis *dashboards* onde afondar e ver máis información da rede recolectada, así como tamén temos a opción de crear as nosas propias gráficas e táboas na sección *Visualize*.

Conclusións

A instalación e posta en marcha de *Elastic Stack* e *Elastiflow*, aínda que longa, é bastante sinxela e fácil de realizar, e máis con esta guía. Unha vez levantada a infraestrutura, esta dános a posibilidade de analizar todo o tráfico da nosa rede e sacar conclusións e información acerca desta, a un custo moi baixo e cun detalle moi grande, polo que pode ser unha solución totalmente viable para o mundo real, o que sumado á súa enorme escalabilidade (que fai que poida ser adaptada a todo tipo de infraestrutura ou necesidade), fai deste *framework* unha solución para monitorización de rede totalmente recomendable.

Cabe destacar a falta nesta práctica dun *firewall* ou *router* para poder parecerse máis ao funcionamento que tería na vida real, pero a falta de este equipo físico levounos a necesidade de buscar outras alternativas, neste caso o uso de *.pcaps* e *exportadores*.

Apéndices

Apéndice A

Posibles erros

Expóñense a continuación, erros obtidos durante a aplicación desta guía e as súas correspondentes solucións:

1. Erro nos *paths*: Dependendo da *distro* ou do sistema operativo que usemos, certos directorios poden cambiar, polo que se se atopan problemas para a execución de certos comandos, no ficheiro de configuración de **Logstash**, **startup.options**, veñen indicados os *paths* aos que referencia a ferramenta.
2. Erro do estilo: *Error: Request Timeout* debido a intentar correr **Kibana** antes de que **Logstash** e **Elasticsearch** estén plenamente operativos: Deberáanse respetar as ordes de instalación e execución propostas na guía, sendo estas sempre da forma que **Logstash** e **Elasticsearch** son previos a **Kibana**, non iniciando nunca un servizo antes de que o anterior esté 100% operativo.
3. Erro do estilo: *Logstash could not be started because there is already another instance*: Isto ocorre, como ben di o erro, cando varias instancias de *Logstash* están correndo de forma simultánea. Adoita pasar cando executamos de forma manual *Logstash* (forma explicada nesta guía, dende */usr/share/logstash/bin*) cando está correndo o servizo. Para solucionar isto, bastará con facer un **stop** e un **disable** do servizo **logstash.service** antes de executar *Logstash* de forma manual.
4. Tempo de espera: A posta en marcha do entorno é lenta, polo que hai que ter paciencia e deixar que as diferentes ferramentas carguen e comecen a funcionar. Este aspecto acentúase en *Logstash* onde poden chegar a facer falta **10 minutos** e un bo número de *warnings* verbosos até que comeza a executarse con normalidade.

-
5. Erro de versión: Durante a realización deste manual tivemos que cambiar de versión do entorno de **ELK** pois dábanos erros no proceso de escoita dos portos. Empezamos coa versión **7.4.1** pero tivemos que actualizar a versión á **7.6.1**, probando tamén a **7.3.1** e vendo que funcionaba en ambas correctamente.
 6. Proceso de **Logstash killed** repentinamente. Isto adoita deberse aos altos requisitos de memoria necesarios para o despliegue. Con menos de 4 GB de RAM e 2 procesadores, pode ser que non sexa suficiente e ocorran este tipo de erros. A solución é sinxela, dotar de máis capacidades á máquina onde desplegamos *Elastiflow*.

Próximos pasos

Propóñense a continuación unha serie de pasos **opcionais** a realizar unha vez teñamos **Elastiflow** operativo, co fin de mellorar as súas funcionalidades. As seguintes medidas foron abordadas dunha forma eminentemente teórica, sen chegar a corroborar o seu funcionamento dun xeito práctico.

1. Activar o **DNS**: Aínda que en versións anteriores era recomendable ter desactivada esta opción por cuestións de rendemento, a partir da versión **3.0.8** do filtro *DNS*, mellórase o mecanismo de caché, tanto de peticións erróneas como correctas habilitando así a utilización de *DNS* sen ter un impacto notable no rendemento da ferramenta.

O importante nesta opción é ter unha configuración correcta da caché, aumentando o volume desta para as peticións fallidas (xa que a maioría serán *DNS timeouts*).

Para levar a cabo isto, bastará con modificar unha serie de variables de entorno como son a activación do *DNS* (**IP2HOST**), cuxos valores poden ser *false*, *true*, *exporters* ou *endpoints*, o **servidor** *DNS* ao que facerlle as consultas e os tempos de **caching** e **TTLs**.

No ficheiro `/etc/systemd/system/logstash.service.d/elastiflow.conf` podemos modificalas:

```
# Name resolution option
Environment="ELASTIFLOW_RESOLVE_IP2HOST=true"
Environment="ELASTIFLOW_NAMESERVER=8.8.8.8"
Environment="ELASTIFLOW_DNS_HIT_CACHE_SIZE=25000"
Environment="ELASTIFLOW_DNS_HIT_CACHE_TTL=900"
Environment="ELASTIFLOW_DNS_FAILED_CACHE_SIZE=75000"
Environment="ELASTIFLOW_DNS_FAILED_CACHE_TTL=3600"
```

Figura B.1: Configuración DNS

-
2. Configurar a **application identification**: Para enriquecer a información, *Netflow* permítenos indicar os dispositivos específicos onde é recollida: Neste momento, *Elastiflow* só ten soporte para dispositivos **fortinet** ou **cisco**, polo que xunto á dirección do mesmo indicáremosllo no ficheiro `/etc/logstash/elastiflow/dictionaries/app_id.srctype.yml`:

```
# DO NOT DELETE these two entries unless you have added your own entries.
"192.0.2.1": "cisco_nbar2"
"192.0.2.2": "fortinet"
"192.168.3.1": "fortinet"
```

Figura B.2: Configuración da Application Identification

3. **Securizar ElasticSearch**: Se usamos a versión **OpenSource** de *ElasticSearch* (como é o caso), ignorarase o usuario e contrasinal por defecto, polo que se poderán deixar sen modificar xa que non serán usados. Se activamos as variables de entorno **ELASTIFLOW_ES_SSL_ENABLE** e **ELASTIFLOW_ES_SSL_VERIFY**, necesitaremos descomentar a opción **cacert** no *output* de *Elasticsearch* (`/etc/logstash/elastiflow/conf.d/30_output_*`) e engadir o *path* ao certificado que usemos.
4. Poderíase incluír o uso dun exportador para o tráfico **sFlow**, xa que foi o protocolo de fluxo de rede que quedou por *testear*.

Bibliografía

- [1] "MANUAL DE INSTALACIÓN DE ELASTIFLOW", <https://github.com/robcowart/elastiflow/blob/master/INSTALL.md/>.
- [2] "REQUISITOS DE SISTEMA OPERATIVO", <https://www.elastic.co/es/support/matrix>.
- [3] "REQUISITOS DE JDK", https://www.elastic.co/es/support/matrix#matrix_jvm.
- [4] "REPOSITORIO OFICIAL ELASTIFLOW", <https://github.com/robcowart/elastiflow/>.
- [5] "INSTALACIÓN DO ENTORNO ELK", <https://www.elastic.co/guide/en/elastic-stack-get-started/7.4/get-started-elastic-stack.html#install-kibana>.
- [6] "CONFIGURACIÓN DE SFLOW", <https://kb.fortinet.com/kb/documentLink.do?externalID=FD32024>.
- [7] "CONFIGURACIÓN DE SFLOW (2)", <https://support.auvik.com/hc/en-us/articles/211530826-How-to-enable-flow-on-your-Fortinet-Fortigate-fire>
- [8] "CONFIGURACIÓN DE NETFLOW", <https://kb.fortinet.com/kb/documentLink.do?externalID=FD36460>.
- [9] "PARSEANDO LOGS CON LOGSTASH", <https://www.elastic.co/guide/en/logstash/current/advanced-pipeline.html>.
- [10] "MANUAL DE SOFTFLOWD", <http://manpages.ubuntu.com/manpages/bionic/man8/softflowd.8.html>.
- [11] "REPOSITORIO YAF", <https://tools.netsa.cert.org/yaf/download.html>.