Linux DFIR - Rapid Audit Log Ingestion with Elasticsearch

Taz Wake | Halkyn Consulting Ltd | Originally published 2024-08-30

During incident response, we are often faced with *suboptimal* situations and incredible time pressures. This means that adaptability is pretty important for incident responders and we just have to find ways to solve our problems, analyse the data and report to our stakeholders.

One common issue I've faced with Linux DFIR is processing the Auditd logs.

Now, to be clear, simply having Auditd logs is a definite improvement and something which is sadly missing from about 75% of organisations I've seen. However, the audit.log format is not a nice thing to read manually.

Example audit.log data viewed with the cat command, super easy to read...

Each event captured by Auditd is likely to result in multiple lines of data and it clearly isn't designed to be read by a human. Yes, you can use **ausearch**, **aureport,** etc., to read it, but that isn't always what we want/need during an investigation. The best approach is to get this data into a SIEM platform.

But we can't always do that.

If you are working a case, you discover a potentially compromised Linux system and (not surprising) it turns out the logs aren't being ingested into the SIEM, you need a solution. Ideally, one that works for 1 system or 100,000 and takes minimal effort from you.

This is where Elasticsearch and Docker can help you out!

In this article, I'll look at how you can quickly spin up an OpenSearch docker container and rapidly ingest auditlogs. This won't work for every situation, but it shows one way to solve the problem! (And you can use this for pretty much any other file format supported by a Beats tool, including Windows EVTX files).

Important caveat - I've only tested this on Ubuntu 22.04.

Also important caveat - this may work on OpenSearch as well, but getting filebeat working will be a bit of a challenge.

Part 1 - Preparation is Everything...

You need to ensure docker and filebeat are installed on your analytical machine. This may require you to remove "unofficial" packages to make sure things are working well. Docker provides guidance about this at https://docs.docker.com/engine/install/ubuntu/

Install Docker

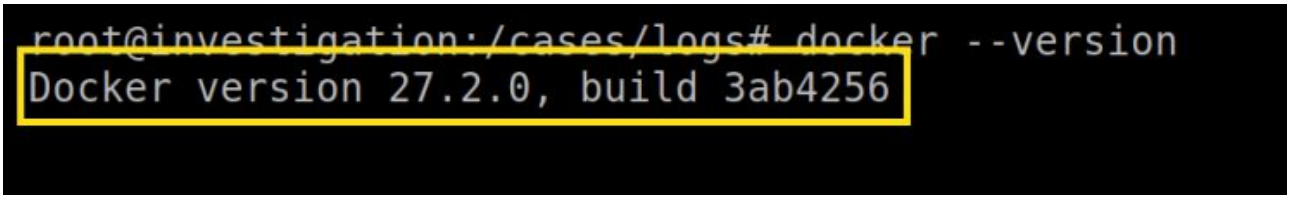
However, if you want to dive right in, run:

Then add the repository keys and sources

Finally, you can install docker!

You can test this by running

and you should see something like this (your version number may vary)



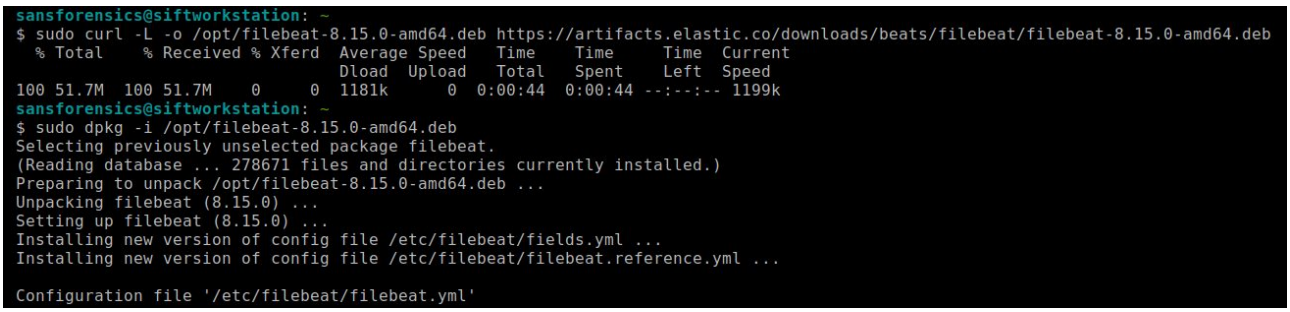
Example output from "docker --version"

Install Filebeat.

There are a few ways to do this but we will keep it simple here and manually install the package.

First, download the package from Elastic.

If all goes well, it should look something like this.



Downloading and installing Filebeat - Note: This was on a system which had a previous install so it may appear differently on your system.

You can validate the install with

And the output should look something like this.

Filebeat version output

Part 2 - Dockering things.

You will need to ensure you have the correct containers - this article will assume you are using Elasticsearch 8.15.0 and Kibana 8.15.0 containers. If you have a different version, adjust the syntax appropriately.

Note: If you need to download these containers, it can take some time as they are approximately 1.1GB each.

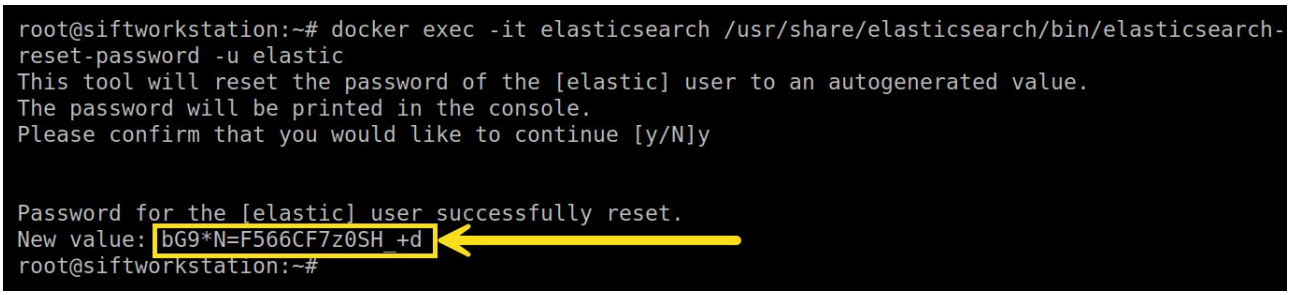
Set up the network - to make life easier down the line

Elasticsearch

Start the Elasticsearch container

Next, generate the Elastic account password (to use with Filebeat and to log in with Kibana) and an enrollment token for kibana:

This should prompt you to see if you are sure you want it to display the password to the console and, if you agree, provide a new password.



Resetting the password for the elastic account

Copy this password and enter it into the **/etc/filebeat/filebeat.yml** file. You will also need this to authenticate against the Kibana login page.

For the Kibana token run

The output should look something like this

```
root@siftworkstation:~# docker exec -it elasticsearch /usr/share/elasticsearch/bin/elasticsearch-
create-enrollment-token -s kibana
eyJ2ZXIiOiI4LjExLjAiLCJhZHIiOlsiMTcyLjE4LjAuMjo5MjAwIl0sImZnciI6ImQ0NTU2M2MyNDhhM2RhMzJiYjc1ZjY3Z
GZjNmRmMGVlNTk0YzJkNjA2N2Q5ZWZmMDljODljN2Y1ZTlkYjNhZGEiLCJrZXkiOiJyYWNXbkpFQjh4bVFaSnliRi00dDo2eW
VZbm1wUFFybVFIakpTN0xOamRnIn0=
```

Generating a Kibana enrollment token.

If you ever need to regenerate the token or reset the password just repeat the steps above.

Some additional steps, which will make life easier later on include exporting the Elastic password to an environment variable and exporting the CA Cert from the Elastic instance.

This should leave you with a file called http_ca.crt in /opt so you can reference this later on..

You can validate the Elastic session with

And the response should look something like this:

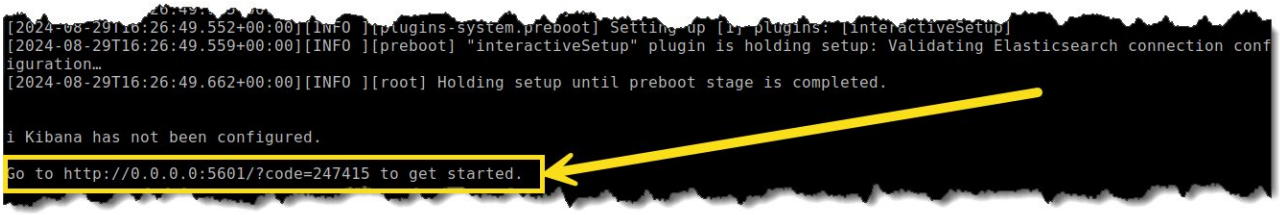
```
{
  "name" : "cdbff1663944",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "TYopolheT82IU1ZIDHnc8Q",
  "version" : {
    "number" : "8.15.0",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "1a77947f34deddb41af25e6f0ddb8e830159c179",
    "build_date" : "2024-08-05T10:05:34.233336849Z",
    "build_snapshot" : false,
    "lucene_version" : "9.11.1",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Validating the Elasticsearch container

Kibana

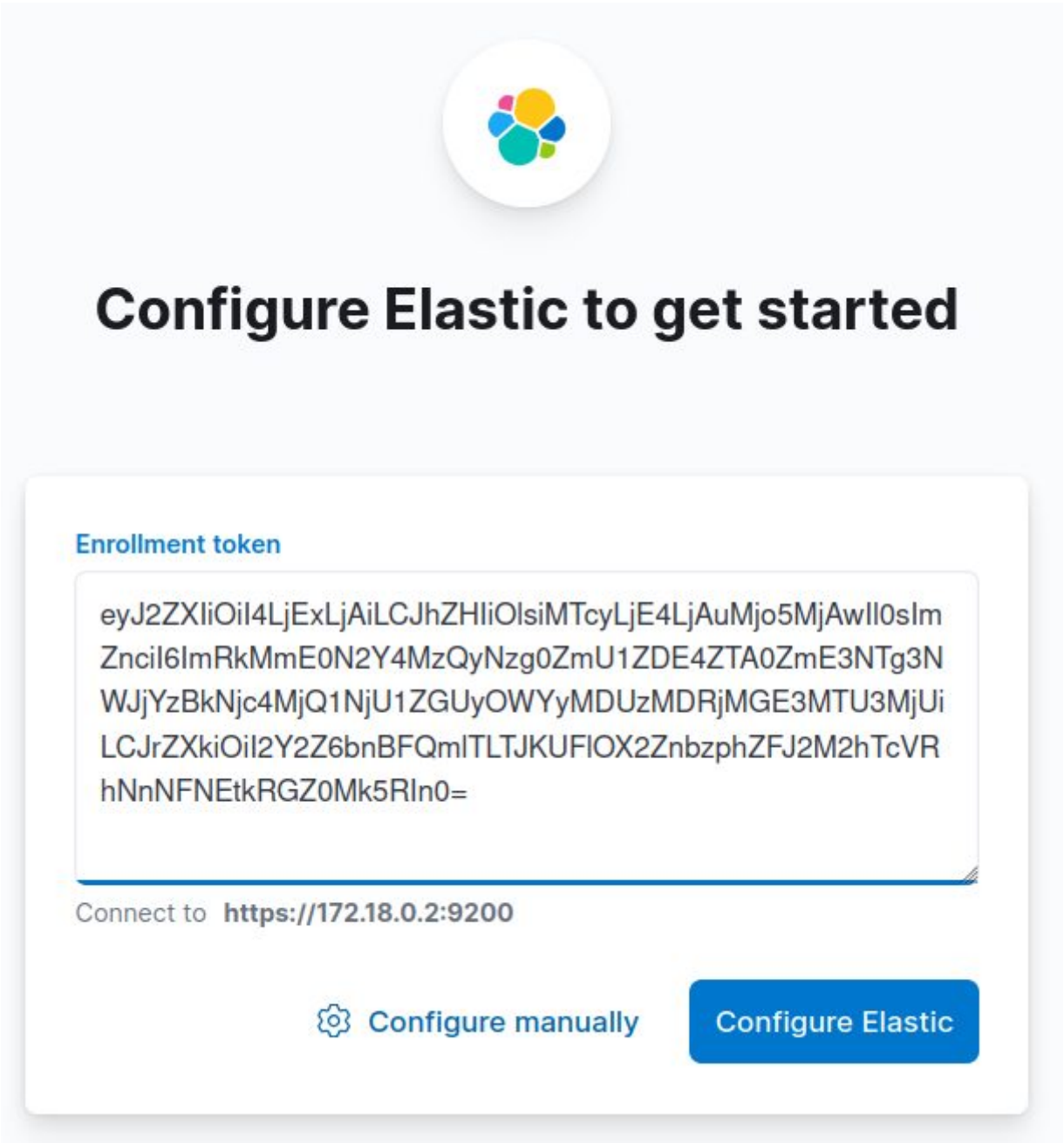
Next, start the Kibana instance. **Note:** *this will run in the terminal you are using, so it is advised to open a second terminal for this.*

When the container starts, you should be presented with a link to start configuring Kibana.



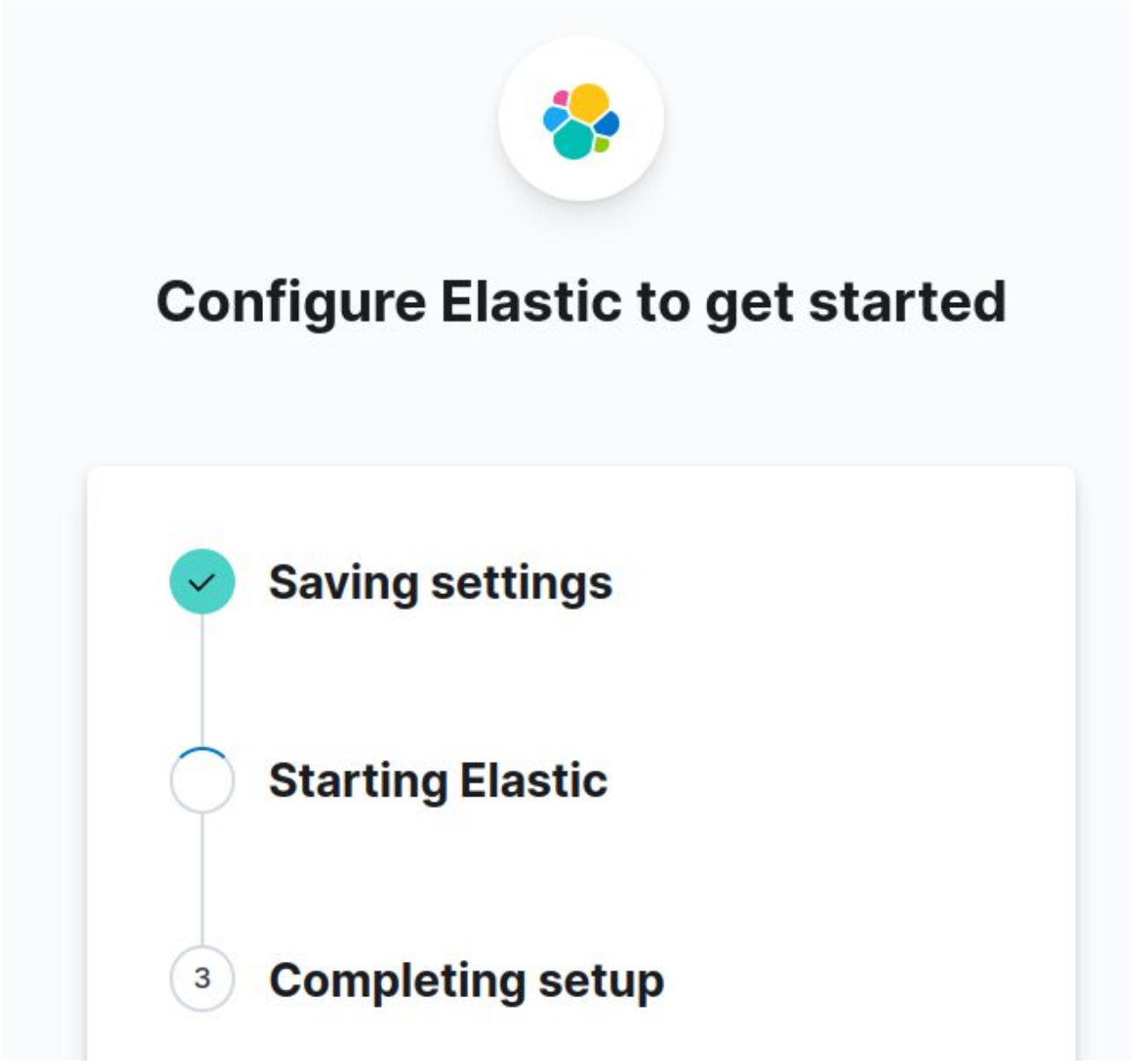
Kibana container showing configuration link

When you open the link, you will be presented with the option to enter your current Kibana enrollment token:



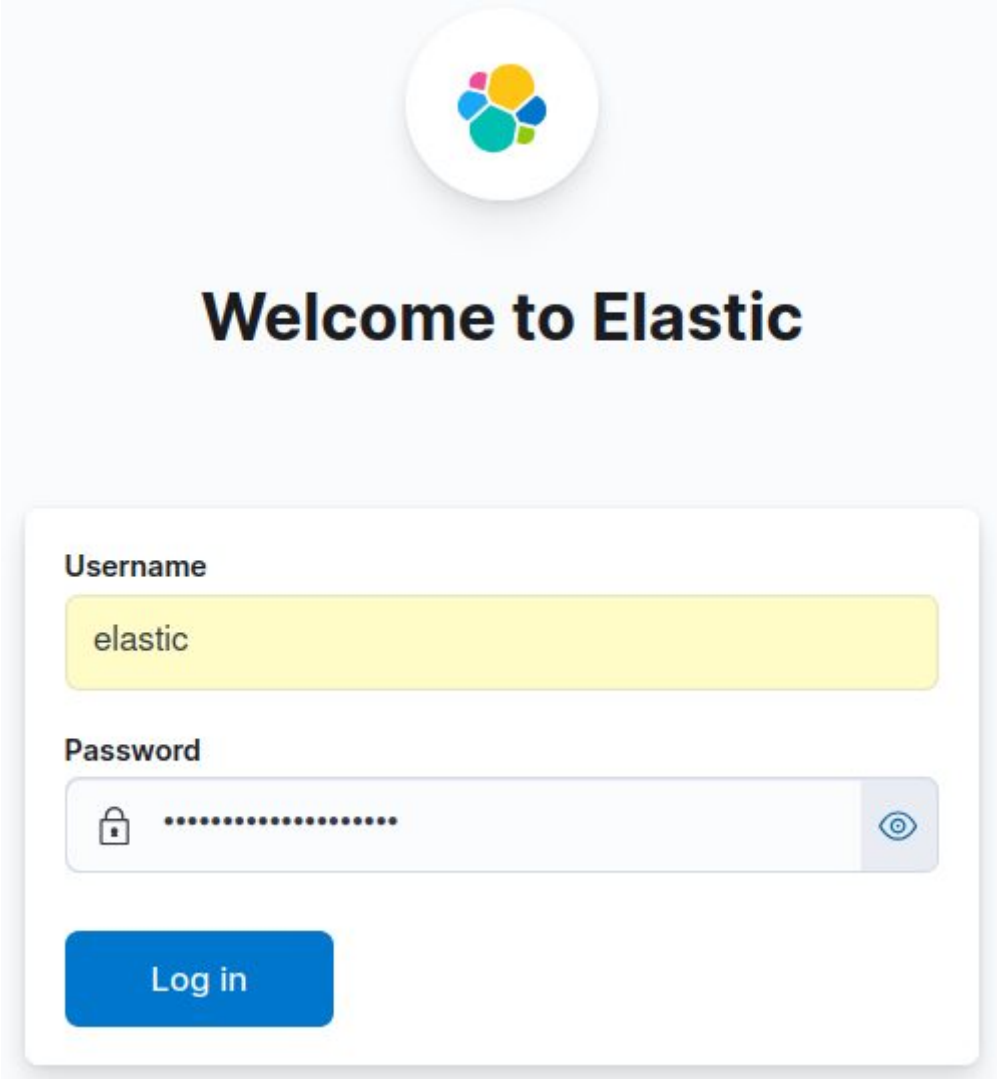
Enter configuration token.

When you click "configure elastic", it will start the process of setting up the environment and the connection to the Elasticsearch container.



Configuring the connection to Elasticsearch

Finally enter the password generated previously and you will be able to log into Kibana.



Kibana Login Page - use the previously generated Elastic password

Now, you haven't uploaded any data so there wont be anything to analyse yet!

But you do have a working Elasticsearch and Kibana configuration, waiting for data.

Part 3 - Filebeating it into shape.

Now we know the Elasticsearch and Kibana containers are working, we can use Filebeat to ingest data. In this example, we are looking at bringing in Auditd logs, but you have a range of options and if you want to analyse Windows Event Logs, you can get even better dashboards with Winlogbeat.

Configure filebeat

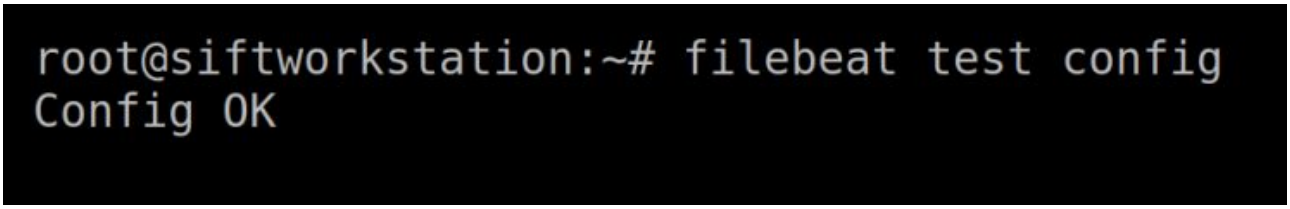
The main step is making sure the filebeat.yml file fits your needs. The example below assumes that the Elasticearch and Kibana containers will be using default ports and available on localhost. It also assumes your audit logs are stored in a folder at /cases/logs/audit. If any of this is different for your environment, change it appropriately.

Open the file at **/etc/filebeat/filebeat.yml** and configure it with the following data. You can add more entries if you wish.

The **ssl.verification_mode: none** reference is to deal with the fact we will be using a self-signed SSL certificate. You may need to adjust paths to reflect your specific environment.

Validate your file with

Hopefully, you will see output that looks like this:



Filebeat config OK

If there are any errors you will need to do some troubleshooting to resolve them.

Next check that the connection between Filebeat, Elasticsearch and Kibana, works with

If all goes well, you should get something that looks like this. If not you will need to do some troubleshooting.

Filebeat test output

Again the WARN notice is because we disabled SSL certificate verification, to allow us to use a self-signed certificate. Do **NOT** use this in a production or internet-facing environment.

Enable the modules

In this example, we are going to analyse logs and possibly some syslog data, so we will enable those modules. If you want to analyse other files, choose the appropriate modules.

Now the individual modules need configuration. They should be in **/etc/filebeat/modules.d/** and named auditd.yml and system.yml

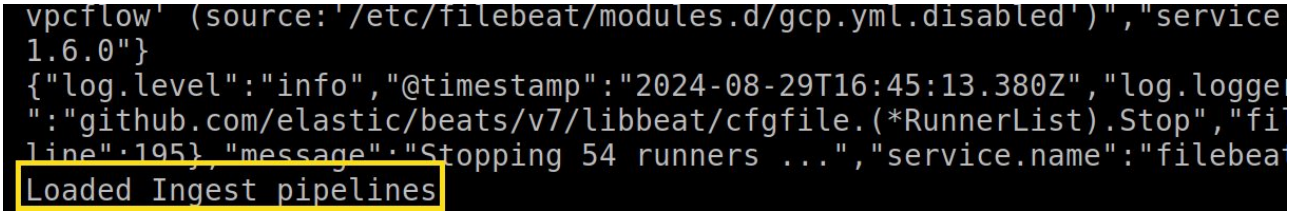
auditd.yml should look something similar to this:

and system.yml should look something like this:

Build dashboards.

Make sure you have updated your filebeat.yml file with the correct password, and then you can use it to configure the Kibana dashboards:

This will run for a minute or two, and generate lots of information on the screen. You should get something similar to this at the end:

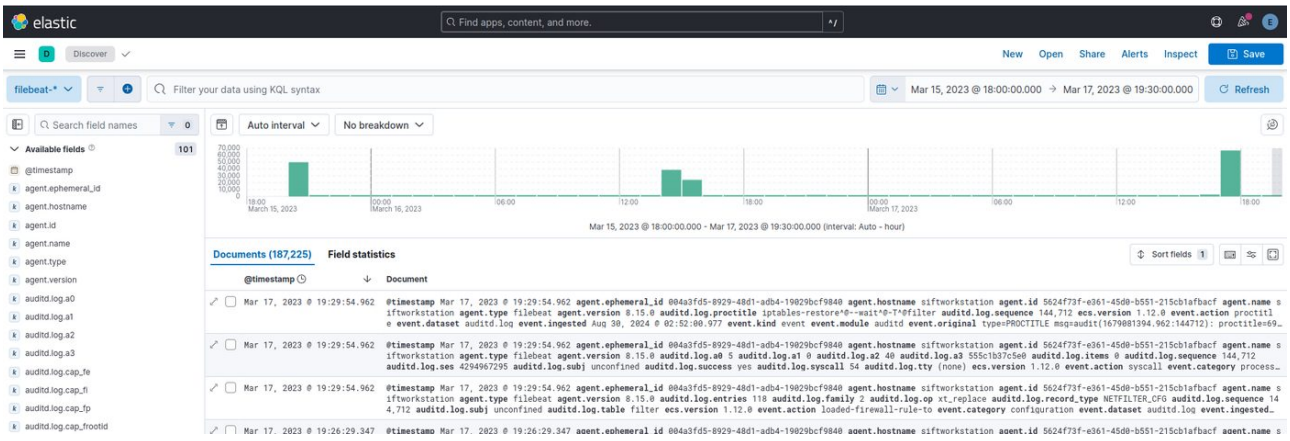


Filebeat setup -e

Part 4 - let it rip.

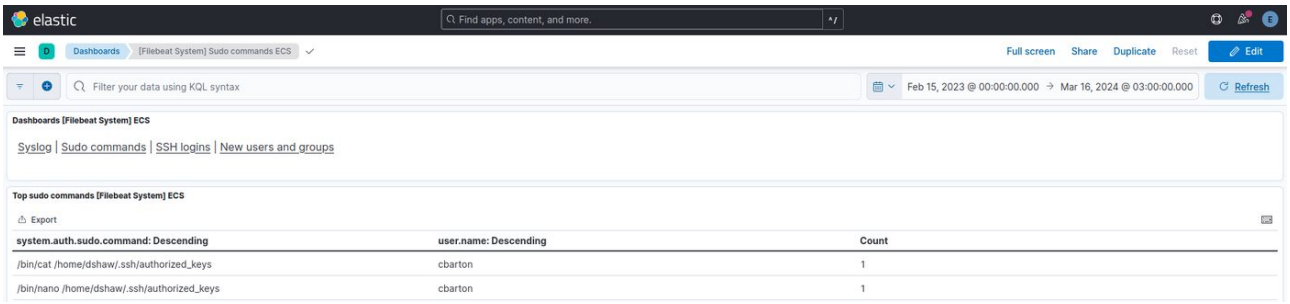
If all has gone well, you can just run:

Wait a few minutes and then the data should be ingested into Elasticsearch and visible in Kibana.

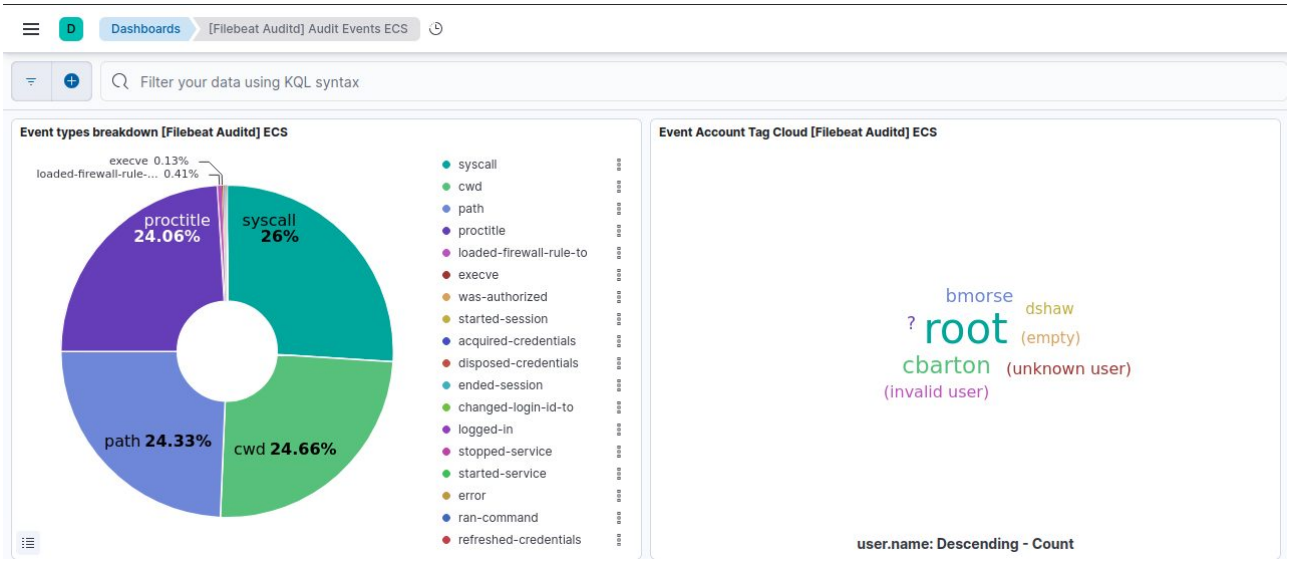


Analysing Auditd Logs in Kibana

The advantage of this approach is that you can use Kibana dashboards to quickly analyse the available evidence.



Dashboards might provide a rapid way to visualise the data.



Visualisations can help, but depend on the data quality.

Considerations

It is important to be aware of the following points:

- This isn't the most efficient way and if you are doing this regularly then it is better to either create a build script or configure a longer-term solution. This is simply a

demonstration of how you can do it in steps that will help you learn more about the process.

- If you are thinking of fully automating this, or creating a portable solution, then you could use a filebeat docker container as well.
- This makes a lot of assumptions. For example, if you have multiple files (such as syslog.1, syslog.2 etc) then you will need to update the paths appropriately.
- You might have to troubleshoot it a **lot**, depending on your local set up. This can be a bit annoying.
- This is a resource hog. The more compute power you can give it, the easier you will find it and the more responsive it will be.
- While it might be possible to port this to OpenSearch, in my experience it is a lot more complex to get working and the support for Filebeat automatic dashboards is much more limited.

Common issues

- Timestamps. If the configuration isn't recognised by Elasticsearch/Filebeat, it is likely to ingest all the data as a flat file with the timestamp set to when you ingest it. This is still something you can analyse, it is just going to be a lot harder.
- Linux logs can be weird. It still requires a fair amount of expertise and experience to analyse this data well.
- If you have complicated, or bespoke logs - including Sysmon for Linux data - this can significantly disrupt how Kibana renders the output.

Summary

This article looks to give some ideas on how you can, with minimal effort, spin up docker containers and analyse large amounts of log data. It won't replace a skilled incident responder, but it might allow that skilled incident responder to work faster.

Hopefully this has given you some ideas on how to implement a similar workflow into your IR, and if so, please let me know.

About Halkyn Consulting

Halkyn Consulting Ltd is a UK-based security and risk management consultancy providing 24/7 incident response, digital forensics, threat hunting, virtual security leadership (vCISO) and security compliance support to organisations worldwide. We deliver clear, practical, vendor-independent advice based on real-world threats — no hype, no upsell.

More free security resources are available at www.halkynconsulting.co.uk/security-resources.

Web: www.halkynconsulting.co.uk | **Email:** info@halkynconsulting.co.uk | **Tel:** +44 1244 940858