TITANIA NIPPER

elastic

How to aggregate your Nipper Audit Reports in Elasticsearch and Explore the Data in Kibana

Nipper and Elastic Integration

Reducing your mean time to detect misconfigurations and vulnerabilities in firewalls, switches and routers, Titania Nipper accurately audits network devices, prioritizes risks and provides exact technical fixes to help remediate issues.

Nipper's accurate audit data – such as your detailed compliance posture against standards including DISA STIG, DHS CDM/NIST 800-53 and PCI – can now be injected into the Elastic Stack via JSON, where the combined solution provides greater scope to analyze and remediate large numbers of your machines on a daily basis.

The Kibana dashboard then gives you the power to examine your security posture from different angles, filtering by categories of error and drilling down to precise detail about devices/models impacted and how to mitigate risks.

This user guide shows you step-by-step how to aggregate your Nipper audit reports in Elasticsearch and use your Kibana dashboard to explore the data.

Published May 2022

Titania Limited
Security House
Barbourne Road
Worcester
WR1 1RS

Telephone: (+44)1905 888 785
Technical Support: support@titania.com
Licensing: enquiries@titania.com
Nipper Support: https://www.titania.com/support/nipper-support

Table of Contents

Contents 3

Prerequisites for Aggregating Nipper Audit Reports in Elasticsearch 4

Step 1 4
Configuring Nipper to Emit JSON in the Correct Format

Step 2 5
Running an Audit

Step 3 6
Creating the Elastic Index

Step 4 7
Use Logstash to Inject Nipper Output into the Elasticsearch Index

Step 5 8
Creating a Kibana Index Pattern

Step 6 11
Exploring the Data

Conclusion and Further Help 12

Prerequisites for Aggregating Nipper Audit Reports in Elasticsearch

Before you begin, please ensure you have completed the prerequisite technical set up:

- Download the digital version of this guide from the support section of the Titania website for a link to scripts you will need to download (a zip file called Nipper_Elastic_Ingest),
- Nipper (v 2.6.3 or above) is licensed and installed on your local Windows 10 machine,
- WSL is configured and available to run Logstash,
- Elastic and Kibana are installed and running on your local machine*, there is no security on the Elastic Index, and
- Docker Desktop is installed on Windows 10 (a powershell script is provided in the Nipper_Elastic_Ingest zip file to pull and run the containers).

New to Nipper?

You can download the Nipper Beginner's Guide from the Titania website: titania.com

- If you need to install Nipper:
  - Go to the 'Downloading Nipper' section of the Nipper Beginner's Guide
- If you need to install your license:
  - Go to the 'Downloading your license' section of the Nipper Beginner's Guide
- To audit your devices and generate reports:
  - Open Nipper and select 'New Report' on the Nipper homepage. Step-by-step guides to generating each report can also be found on the website: www.titania.com/support

* If Elastic and Kibana are installed remotely, the URLs provided in the digital version of this guide will need to be updated accordingly, and the Logstash conf script adjusted to connect to the instance. An example file 'ls_with_creds.conf' is provided in the Nipper_Elastic_Ingest zip file.

For further information on installing the Elastic stack, please refer to the Elastic website: elastic.co

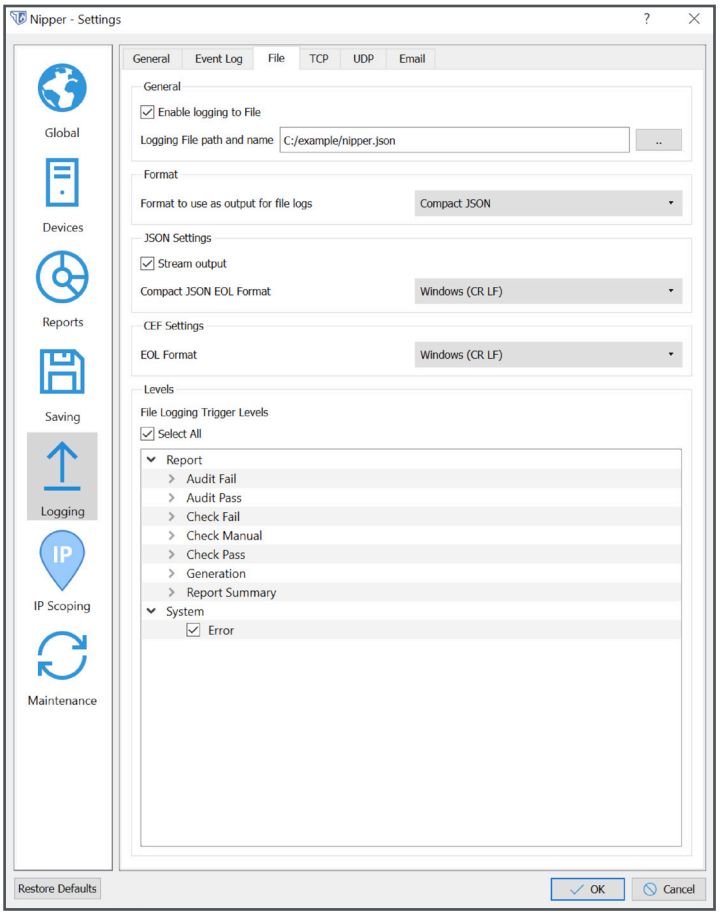
Step 1

Configuring Nipper to Emit JSON in the Correct Format

Logstash expects JSON in NDJSON. This means that each JSON Object appears on a separate line in the file, and not encapsulated in an array.

In order to configure Nipper to emit the JSON in the correct format you need to:

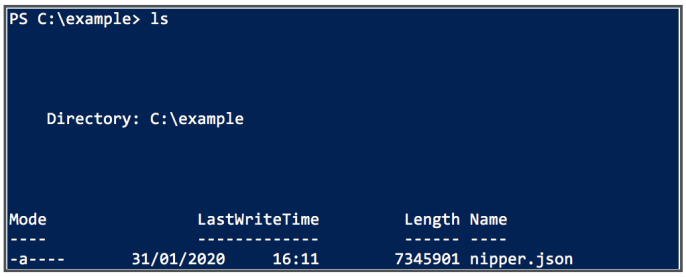
- Open Nipper and click 'Settings'
- Click the 'Logging' icon and open the 'File' tab
- Ensure that:
  - 'Enable logging to File' is checked
  - The file path to the output file is OK
  - 'Compact JSON' is selected from the dropdown
  - 'Stream output' is checked, and
  - 'Select All' Logging Trigger Levels is checked
- Finally, click 'OK' to confirm the settings.



Step 2

Running an Audit

- Now click the ‘Reports’ icon to choose the audit you wish to run
- Follow the onscreen instructions to choose the network device configurations you wish to include in your reports scope
- Click ‘Finish’
- The file will now appear in the specified directory.



If there are lots of devices being audited and/or lots of audit types being conducted, it can take time to write out the file after the audit is complete.

Listing the size of the file a few times until it stops growing in size ensures that the process is complete.

Please note Nipper will append to this file if further audits are performed, so you may wish to move/delete the file before performing a subsequent audit.

The contents of nipper.json should look similar to the fragment below, which is shown as an example:

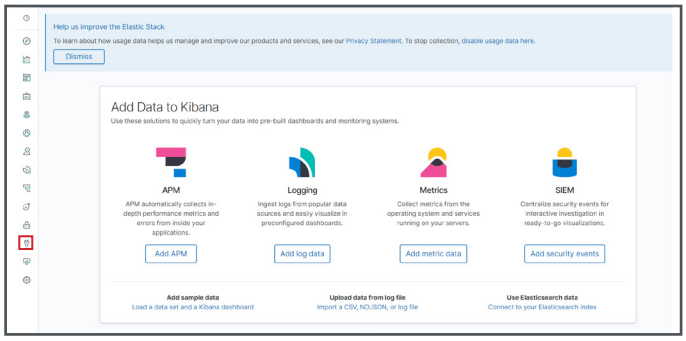

```
{"audit_type":"Security Audit","date_time":"Fri Jan 31 16:05:41
2020","device":{"collection_ip":"","filename":"3com5500
.txt","hostname":"5500-EI","manufacturer":"3COM","model":"5500 Series
Switch","operating_system":{"name":"SS4","version
":"5500-EI"}},"ease":{"description":"Dictionary-based password
guessing attacks have been widely documented on the Inte
rnet and published media, enabling an attacker with very little
knowledge or experience to perform the attack. There ar
e a numb
```


* Note there is no ‘[‘ opening bracket. Just a ‘{‘ opening bracket, and the JSON record is all on one line.

Step 3

Creating the Elastic Index

» Navigate to your Kibana dashboard:
http://localhost:5601/app/kibana#/home?_g=()



» Select the ‘Dev Tools’ icon from the left hand toolbar

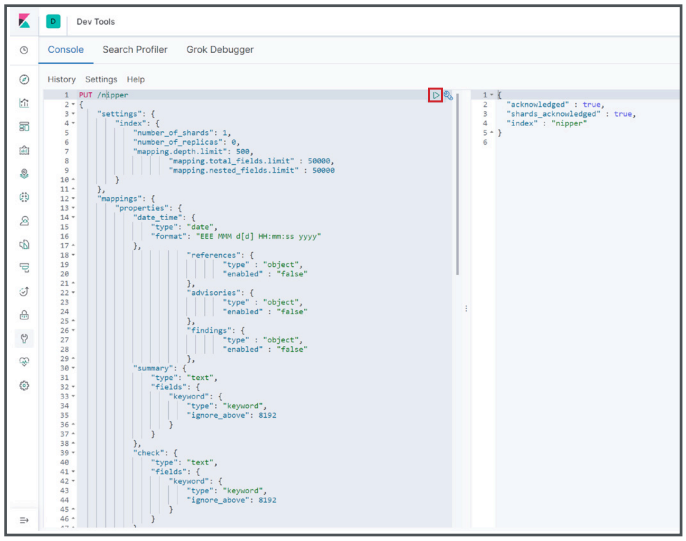
» Now configure the index and apply a mapping.
The mapping extends the index length of some fields, and masks out those not needed.

Locate the .txt file script (shown right) in the Nipper_Elastic_Ingest zip file to copy and paste into the Console panel.

» Once the text has been pasted into the console, click anywhere inside the text, then click the ‘Run’ arrow in the top right hand corner.

This action creates an index called ‘nipper’ with the correct mappings to accept the data from the tool.

If the index already exists, then you will get an error in the right hand pane after clicking ‘Run’.

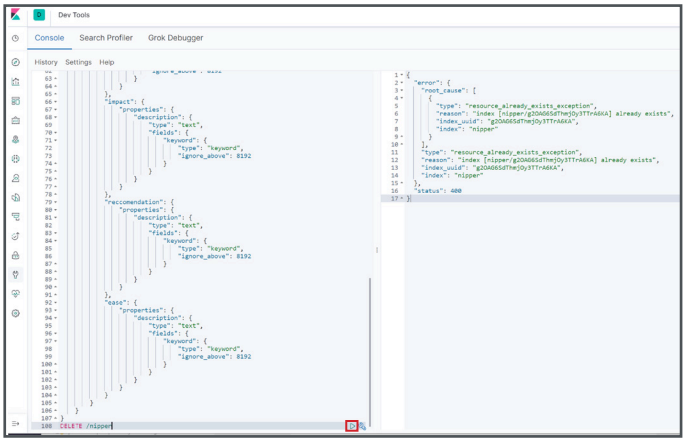



```
PUT /nipper
{
  "settings": {
    "index": {
      "number_of_shards": 1,
      "number_of_replicas": 0,
      "mapping.depth.limit": 500,
            "mapping.total_fields.limit" : 50000,
            "mapping.nested_fields.limit" : 50000
    }
  },
  "mappings": {
    "properties": {
      "date_time": {
        "type": "date",
        "format": "EEE MMM d[d] HH:mm:ss yyyy"
      },
            "references": {
                "type" : "object",
                "enabled" : "false"
            },
            "advisories": {
                "type" : "object",
                "enabled" : "false"
            },
            "findings": {
                "type" : "object",
                "enabled" : "false"
            },
      "summary": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 8192
          }
        }
      },
      "check": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 8192
          }
        }
      },
      "fix": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 8192
          }
        }
      },
      "description": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 8192
          }
        }
      },
      "impact": {
        "properties": {
          "description": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 8192
              }
            }
          }
        }
      },
      "reccomendation": {
        "properties": {
          "description": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 8192
              }
            }
          }
        }
      },
      "ease": {
        "properties": {
          "description": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 8192
              }
            }
          }
        }
      }
    }
  }
}
```

» If you wish to start afresh, issue a ‘DELETE /nipper’ on the Console pane, and then try again.

There is no need to replace the index creation text, just append it in the Console window, click on it, then click ‘Run’. Once the index is deleted, you can return to the creation text, click that, and press ‘Run’ again.

You now have an index with the correct mapping to accept Titania data.



Step 4

Use Logstash to Inject Nipper Output into the Elasticsearch Index

The next step is to get the data into the index. An easy way to do this is using Logstash from the Elastic ELK stack. To do this, Logstash needs a config file.

» Locate the .exe file named ‘l.conf’ (shown right) in the Nipper_Elastic_Ingest zip file.

» Now invoke Logstash:
cat nipper.json | logstash -f l.conf

» The nipper.json data is now in Elastic.

```
input { stdin{} }
filter {
    json {
        source => "message"
    }
}
output {
    elasticsearch {
            hosts => ["localhost:9200"]
            index => "nipper"
        }
}
```

Below it is invoked on a WSL (windows subsystem for Linux) Ubuntu instance. Note the output to the console issues some warnings, but completes successfully:

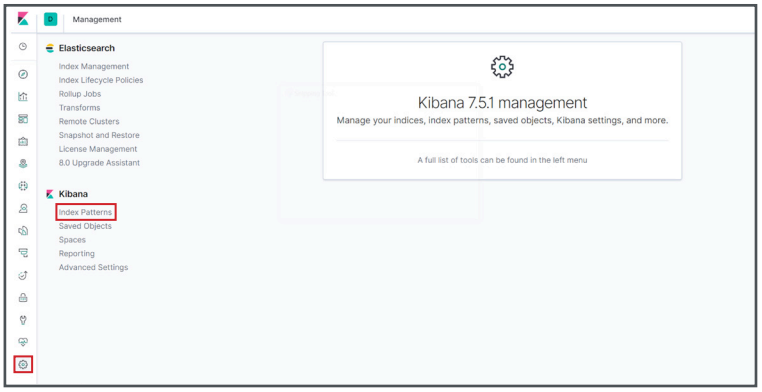
```
cat nipper.json | logstash -f l.conf --path.data . -l .
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.headius.backport9.modules.Modules (file:/usr/share/logstash/logstash-core/lib/jars/jruby-complete-9.2.8.0.jar) to field java.io.FileDescriptor.fd
WARNING: Please consider reporting this to the maintainers of com.headius.backport9.modules.Modules
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Thread.exclusive is deprecated, use Thread::Mutex
WARNING: Could not find logstash.yml which is typically located in $LS_HOME/config or /etc/logstash. You can specify the path using --path.settings. Continuing using the defaults
Could not find log4j2 configuration at path /usr/share/logstash/config/log4j2.properties. Using default config which logs errors to the console
[INFO ] 2020-01-31 18:01:44.570 [main] writabledirectory - Creating directory {:setting=>"path.queue", :path=>"./queue"}
[INFO ] 2020-01-31 18:01:44.593 [main] writabledirectory - Creating directory {:setting=>"path.dead_letter_queue", :path=>"./dead_letter_queue"}
[WARN ] 2020-01-31 18:01:45.367 [LogStash::Runner] multilocal - Ignoring the 'pipelines.yml' file because modules or command line options are specified
[INFO ] 2020-01-31 18:01:45.386 [LogStash::Runner] runner - Starting Logstash {"logstash.version"=>"7.5.2"}
[INFO ] 2020-01-31 18:01:45.428 [LogStash::Runner] agent - No persistent UUID file found. Generating new UUID {:uuid=>"5b1127a5-1139-4949-aec4-c18a3e88fbfa", :path=>"./uuid"}
[INFO ] 2020-01-31 18:01:47.634 [Converge PipelineAction::Create<main>] Reflections - Reflections took 66 ms to scan 1 urls, producing 20 keys and 40 values
[INFO ] 2020-01-31 18:01:49.890 [[main]-pipeline-manager] elasticsearch - Elasticsearch pool URLs updated {:changes=>{:removed=>[], :added=>[http://localhost:9200/]}}
[WARN ] 2020-01-31 18:01:50.199 [[main]-pipeline-manager] elasticsearch - Restored connection to ES instance {:url=>"http://localhost:9200/"}

[INFO ] 2020-01-31 18:01:50.475 [[main]-pipeline-manager] elasticsearch - ES Output version determined {:es_version=>7}

[WARN ] 2020-01-31 18:01:50.482 [[main]-pipeline-manager] elasticsearch - Detected a 6.x and above cluster: the `type` event field won't be used to determine the document _type {:es_version=>7}
[INFO ] 2020-01-31 18:01:50.563 [[main]-pipeline-manager] elasticsearch - New Elasticsearch output {:class=>"LogStash::Outputs::Elasticsearch", :hosts=>["//localhost:9200"]}

[INFO ] 2020-01-31 18:01:50.647 [Ruby-0-Thread-5: :1] elasticsearch - Using default mapping template
[WARN ] 2020-01-31 18:01:50.714 [[main]-pipeline-manager] LazyDelegatingGauge - A gauge metric of an unknown type (org.jruby.specialized.RubyArrayOneObject) has been create for key: cluster_uuids.
This may result in invalid serialization.  It is recommended to log an issue to the responsible developer/development team.
[INFO ] 2020-01-31 18:01:50.726 [[main]-pipeline-manager] javapipeline - Starting pipeline {:pipeline_id=>"main", "pipeline.workers"=>8, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.
max_inflight"=>1000, "pipeline.sources"=>["/c/example/l.conf"], :thread=>"#<Thread:0x6dd7bd2c run>"}

[INFO ] 2020-01-31 18:01:50.768 [Ruby-0-Thread-5: :1] elasticsearch - Attempting to install template {:manage_template=>{"index_patterns"=>"logstash-*", "version"=>60001, "settings"=>{"index.
refresh_interval"=>"5s", "number_of_shards"=>1}, "mappings"=>{"dynamic_templates"=>[{"message_field"=>{"path_match"=>"message", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text",
"norms"=>false}}}, {"string_fields"=>{"match"=>"*", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword", "ignore_above"=>256}}}}],
"properties"=>{"@timestamp"=>{"type"=>"date"}, "@version"=>{"type"=>"keyword"}, "geoip"=>{"dynamic"=>true, "properties"=>{"ip"=>{"type"=>"ip"}, "location"=>{"type"=>"geo_point"}, "latitude"=>{"type"=>"half_
float"}, "longitude"=>{"type"=>"half_float"}}}}}}}
[INFO ] 2020-01-31 18:01:50.965 [[main]-pipeline-manager] javapipeline - Pipeline started {"pipeline.id"=>"main"}
The stdin plugin is now waiting for input:
[INFO ] 2020-01-31 18:01:51.130 [Agent thread] agent - Pipelines running {:count=>1, :running_pipelines=>[:main], :non_running_pipelines=>[]}
[INFO ] 2020-01-31 18:01:51.792 [Api Webserver] agent - Successfully started Logstash API endpoint {:port=>9600}
[INFO ] 2020-01-31 18:01:57.911 [LogStash::Runner] runner - Logstash shut down.
```

Step 5

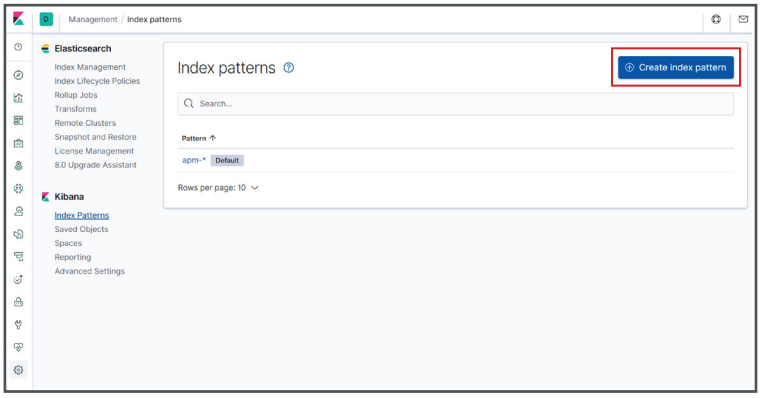
Creating a Kibana Index Pattern

» Firstly, click on the ‘Settings’ icon in the Kibana dashboard

» And click on the ‘Index Patterns’ link



» Click on the blue ‘Create Index Pattern’ button



» Now type the name of the index you created into the index pattern box

You don’t have to type the complete name - you can use wild cards (this helps if you want Kibana to look over multiple Elastic indexes) - but in this case, typing nipper* works.

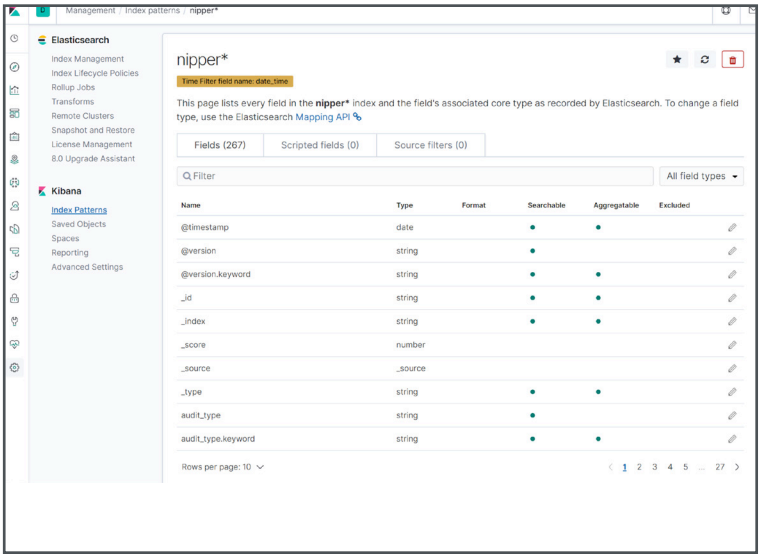
It will tell you Kibana has matched with the Elastic index called nipper*.

» Click the ‘Next Step’ button

» Select date_time from the drop down box, and click the ‘Create Index’ pattern.

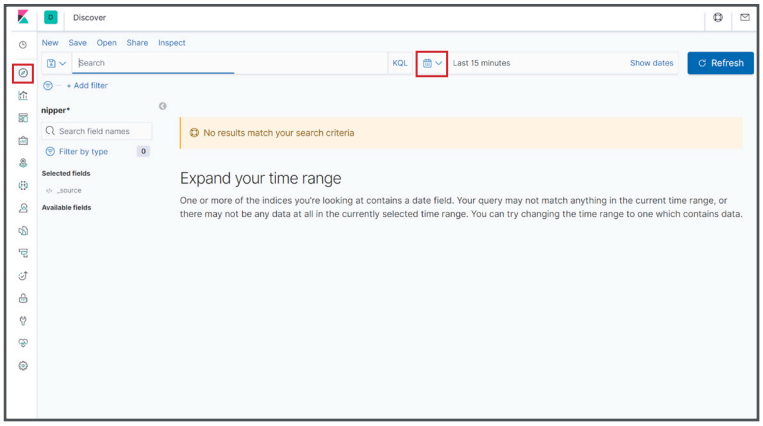
The date_time is the field you mapped to contain the date of the events in the Nipper JSON output.

You will now see that the index has been created.

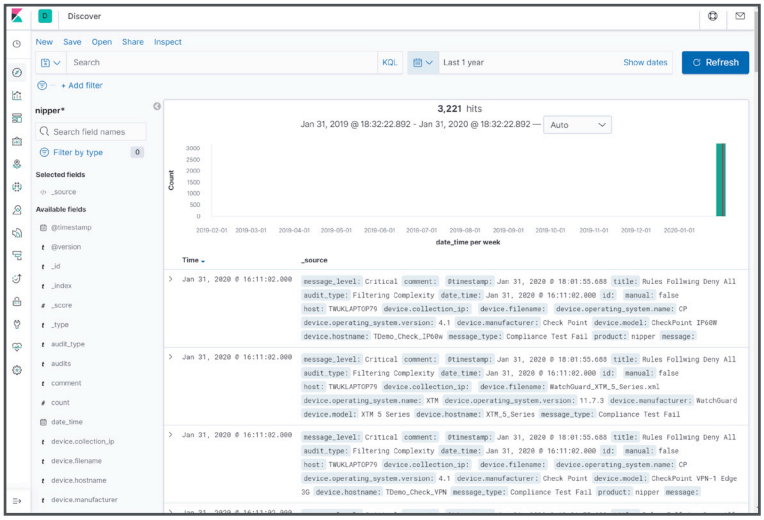


» Next, click on the ‘Discover’ icon on the left toolbar.

If the data you are analysing wasn’t created in the last 15 minutes, it is likely you will need to change the time window with the calendar item to see the data.

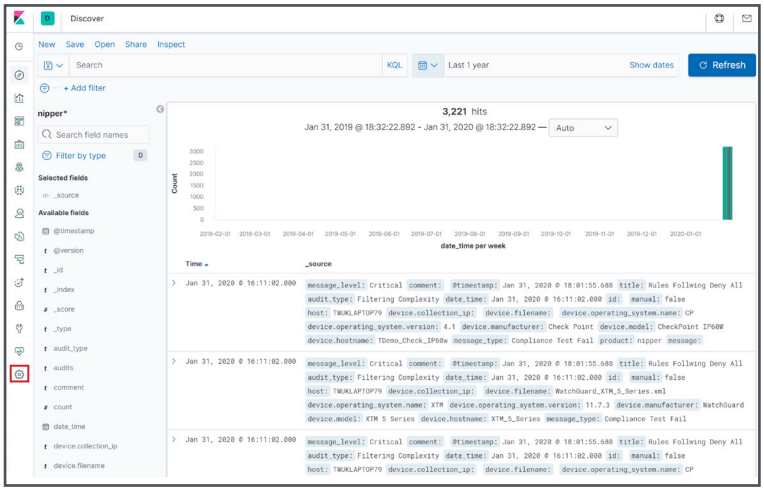


» Now you should see the data loaded into Elastic. In this case there are 3221 records.

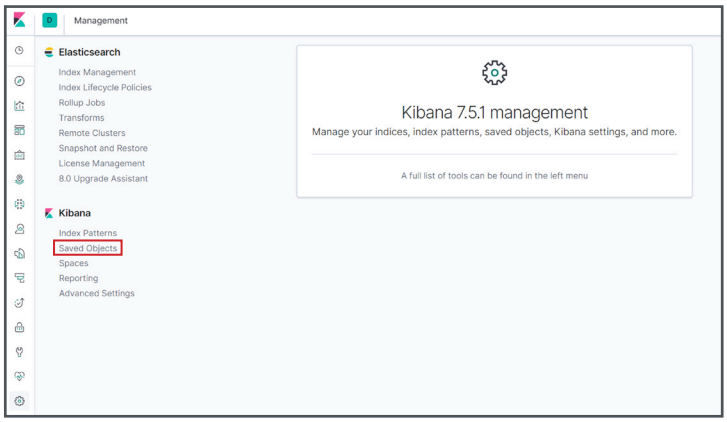


» Load in the dashboard

» And select the ‘Settings’ menu again



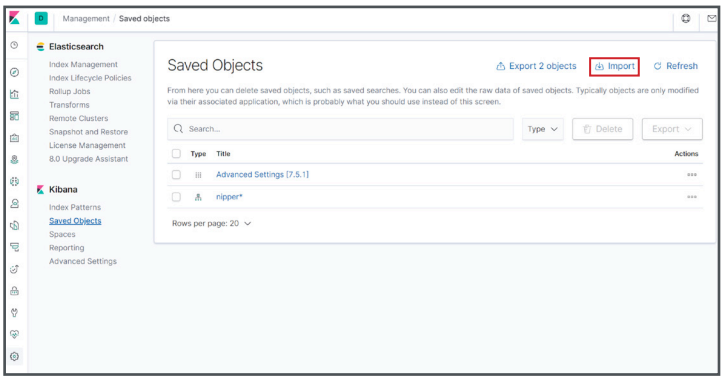
» Select the ‘Saved Objects’ link



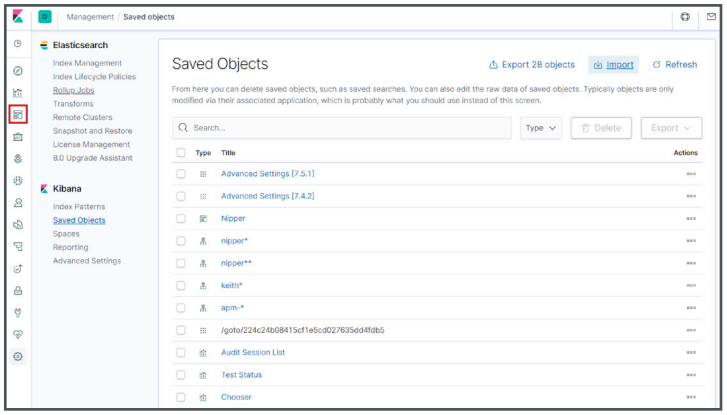
» Click the ‘Import Objects’ button

» Now from the requester, import the nipper_kibana_dashboard.ndjson provided in the Nipper_Elastic_Ingest zip file.

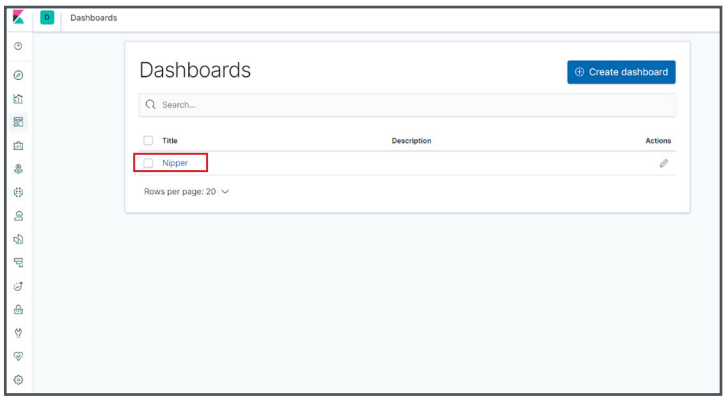
This file contains the definitions of example visualisations, as well as a dashboard containing those visualisations.



» Select the ‘Dashboard’ icon



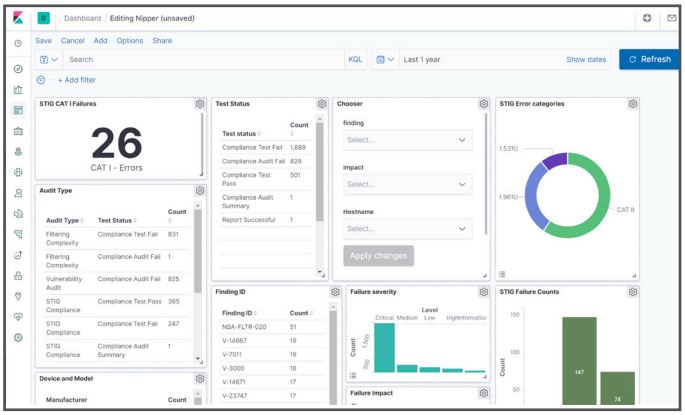
» And finally, click on the Nipper dashboard link.



Step 6

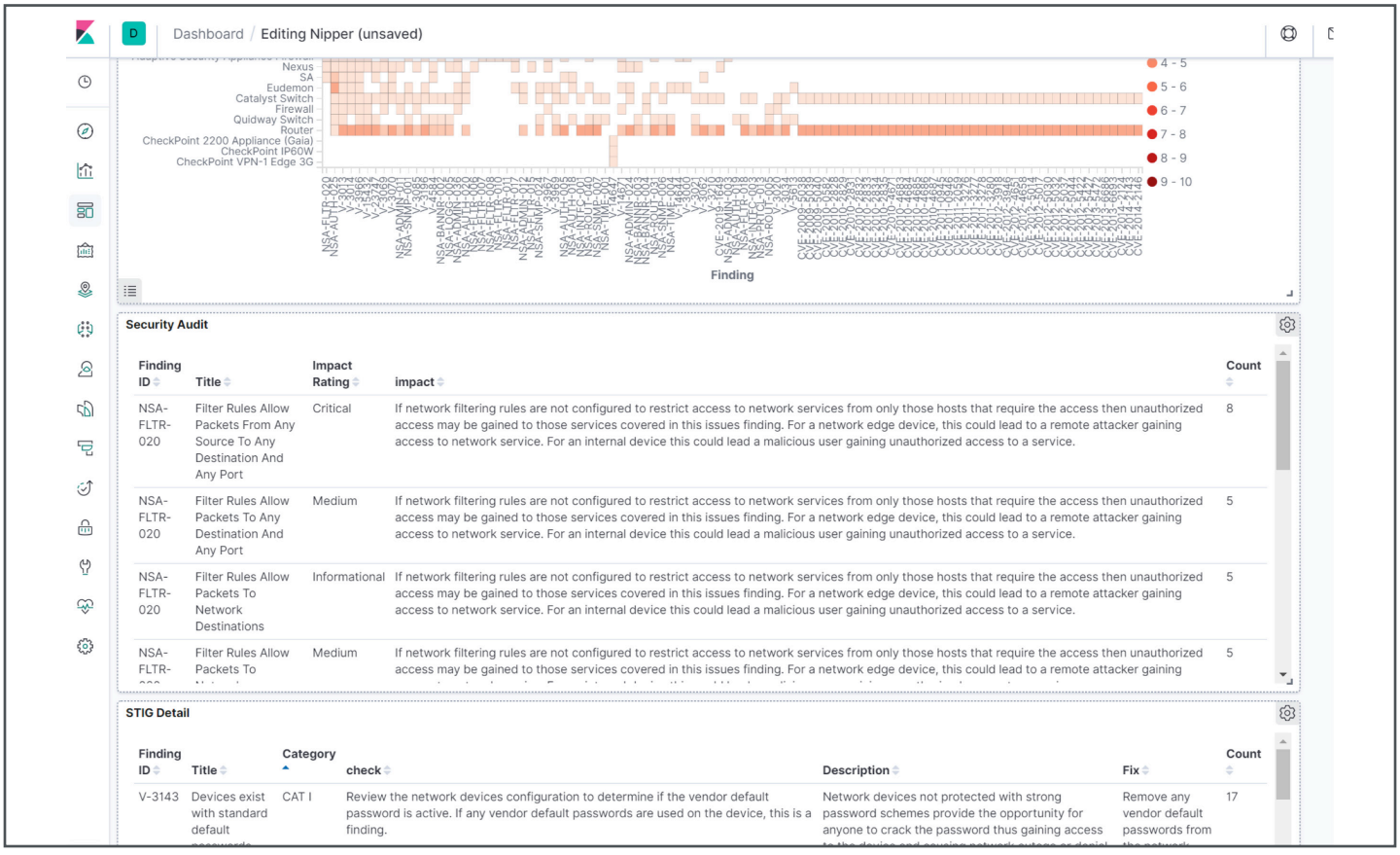
Exploring the Data

» You will now be presented with a dashboard like this allowing you to click and filter on the results in the usual Kibana manner



» Scroll down the dashboard to see heat maps and detailed audit findings and vulnerabilities

Here you can explore your security posture from different angles, filtering by categories of error and drilling down to precise detail about devices/models impacted and how to mitigate risks...



Conclusion and Further Help

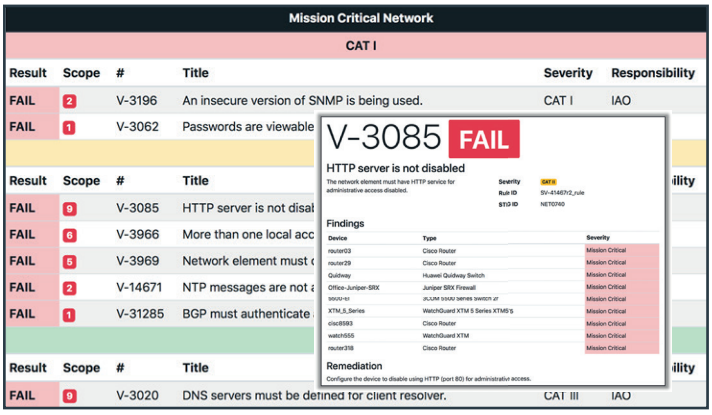
If you have followed this guide, you will see how quick and easy it is to aggregate your Nipper audit reports in Elasticsearch.

Now you can explore your data in Kibana, prioritize your risks and use Nipper's exact technical fixes to help remediate any vulnerabilities or issues on your network.

If you would like any help or advice about the steps or scripts included in this guide, simply contact our dedicated Support team on:

Tel: (+44)1905 888 785
Email: support@titania.com

Our solution advisors will be more than happy to help walk you through this or any other auditing processes with our Nipper software.



Example analytics shows the prioritization of remediation that can be achieved when audit data is combined with value chain data on the mission criticality of the device/ network.

About Nipper

Nipper accurately audits the security of firewalls, switches and routers to detect exploitable misconfigurations that pose risk to the network, prioritized by criticality. Applying Nipper's compliance lens to the findings also provides the evidence needed to assure compliance with RMFs including DISA RMF, NIST 800-53/171, STIG, CMMC and PCI. All findings are output as an easy-to-read report, or a JSON for integration with SIEM, GRC and other data visualization systems.

Nipper's risk remediation advice and exact technical fixes for misconfigurations also support and accelerate the process of becoming secure and compliant.

About Titania

Protecting over 25 million people globally, Titania software is trusted to secure the world's most critical networks against preventable attacks. Nipper intelligently automates configuration auditing to analyze misconfigurations and validate your network security against the latest risk management frameworks, assurance and compliance standards.

Stay secure and complaint with Nipper. Find out more.

titania.com/products/nipper/