

ELK Setup, Log Ingestion & Attack Detection

WE Innovate X Zero\$exploit

Supervised by : Ahmed Halwagy – Zeyad Mazen – Adel Ahmed

Prepared by : [Omar Hassan](#)



zero\$exploit
Cyber Security  trusted partner



EG|CERT



Required Tasks

- Installing & configuring Elasticsearch
- Installing & configuring Kibana
- Connecting Elasticsearch with kibana
- Installing & configuring Fluentbit
- Installing & Configuring Winlogbeat
- Writing detection rules & simulating a suspicious activity



Requirements

- VMware / Virtual Box
- Windows 10/11 ISO – Ubuntu (20.0/22.0/24.0) ISO
- 16 GB RAM – 60 GB Disk Space
- 4 CPU Cores

Ubuntu Machine


Setting	Recommended
RAM	5-6 GB
Disk	20-30 GB
CPU	2-3 Cores
Network	NAT

Windows Machine

Setting	Recommended
RAM	2-3 GB
Disk	30-40 GB
CPU	1-2 Cores
Network	NAT


PHASE 1 : Installing & configuring Elasticsearch

Updating Ubuntu packages



```
$ sudo apt update  
$ sudo apt upgrade -y
```

Installing required packages & dependencies



```
$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |  
sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg  
  
$ sudo apt-get install apt-transport-https  
  
$ echo "deb [signed-by=/usr/share/keyrings/elasticsearch-  
keyring.gpg] https://artifacts.elastic.co/packages/9.x/apt stable  
main" | sudo tee /etc/apt/sources.list.d/elasticsearch-9.x.list  
  
$ sudo apt-get update && sudo apt-get install elasticsearch
```

Configuring elasticsearch.yml



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

Lines to be uncommented : Network.host & http.port

Lines to be added :

discovery.type: single-node

```
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 0.0.0.0
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# ----- Discovery -----
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
#discovery.seed_hosts: ["host1", "host2"]
#
#bootstrap.memory_lock: false
discovery.type: single-node
```

Enabling & starting the Elasticsearch service

```
$ sudo systemctl enable elasticsearch
$ sudo systemctl start elasticsearch
$ sudo systemctl status elasticsearch
```

If you receive **active(running)** then everything is working

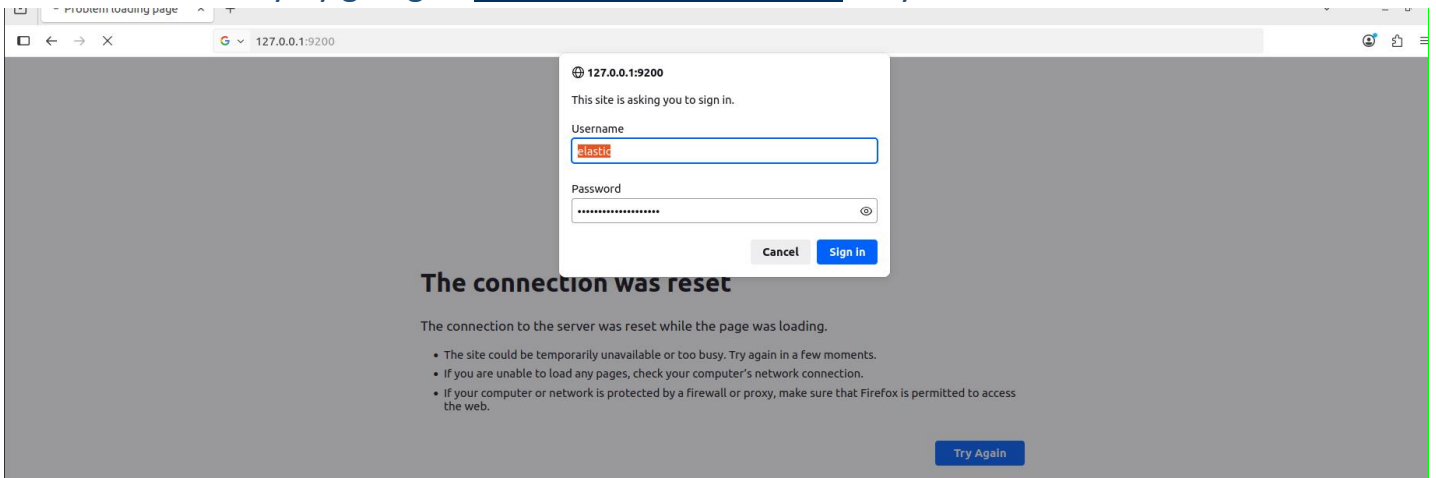
```
omar@omar-virtual-machine:~$ sudo systemctl enable elasticsearch
omar@omar-virtual-machine:~$ sudo systemctl start elasticsearch
omar@omar-virtual-machine:~$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-08-14 20:57:52 EEST; 44min ago
     Docs: https://www.elastic.co
   Main PID: 951 (java)
    Tasks: 99 (limit: 6841)
   Memory: 1.5G
      CPU: 3min 58.995s
   CGroup: /system.slice/elasticsearch.service
           └─ 951 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=/usr/share/elasticsearch/bin/e>
           └─ 1548 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouc>
           └─ 1991 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

20:54:26 14 أغسطس 2025 omar-virtual-machine systemd[1]: Starting Elasticsearch...
20:57:52 14 أغسطس 2025 omar-virtual-machine systemd[1]: Started Elasticsearch.
lines 1-15/15 (END)
```

Restarting the elasticsearch service

```
$ sudo systemctl restart elasticsearch
```

Check connectivity by going to <https://127.0.0.1:9200/> on your web browser on Ubuntu.

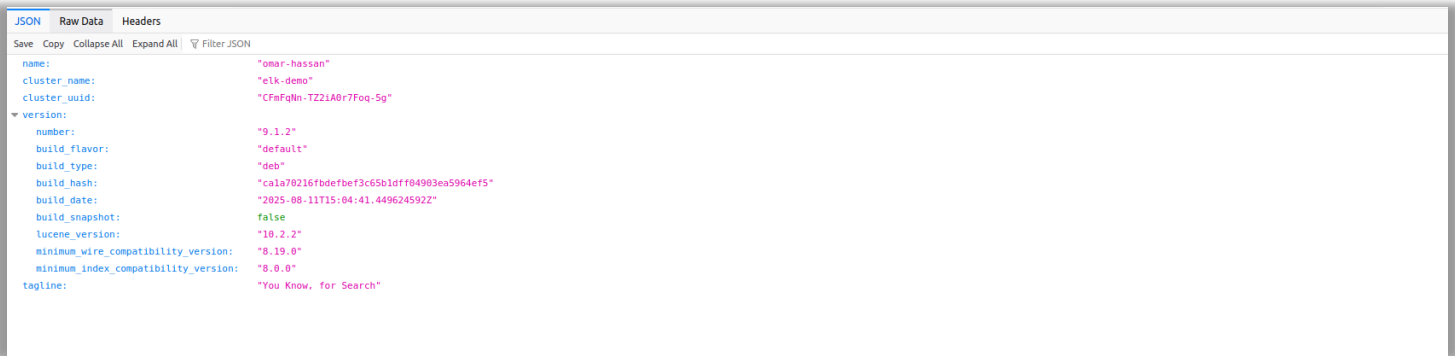


On the first go it should ask you about a username & password , the default username is **elastic** , the password should be reset using the following command.

```
$ sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic
```

Then you can save your password in a txt file for later use , you can now login into elasticsearch using username elastic & the password displayed in the terminal when you reset it.

After logging in



PHASE 2 : Installing & configuring Kibana

```
$ sudo apt-get update && sudo apt-get install kibana
$ sudo nano /etc/kibana/kibana.yml
```

Only uncomment the server.port & server.host

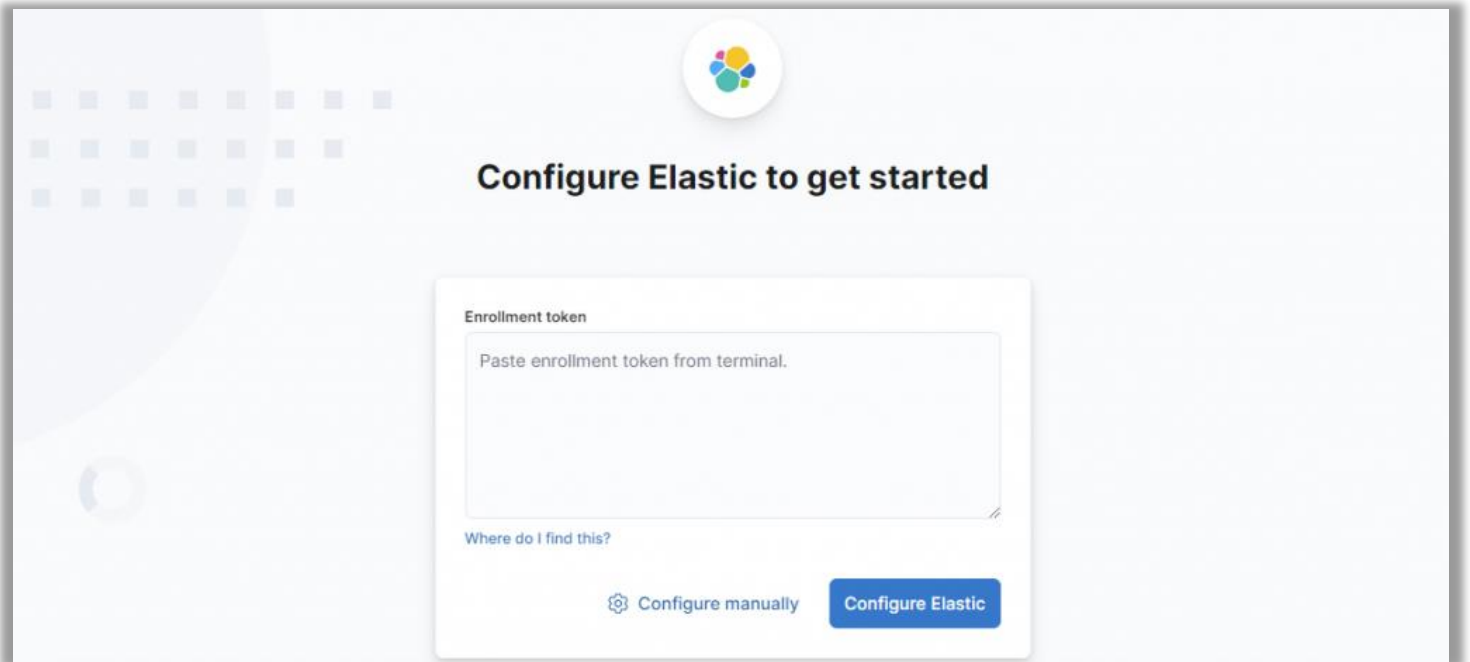
```
# ===== System: Kibana Server =====
# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601

# 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: "0.0.0.0"
```


Enabling & starting the kibana.service

```
$ sudo systemctl enable kibana.service
$ sudo systemctl start kibana.service
```

To check connectivity go to <http://127.0.0.1:5601> , and you will then be asked for an **enrollment token**.

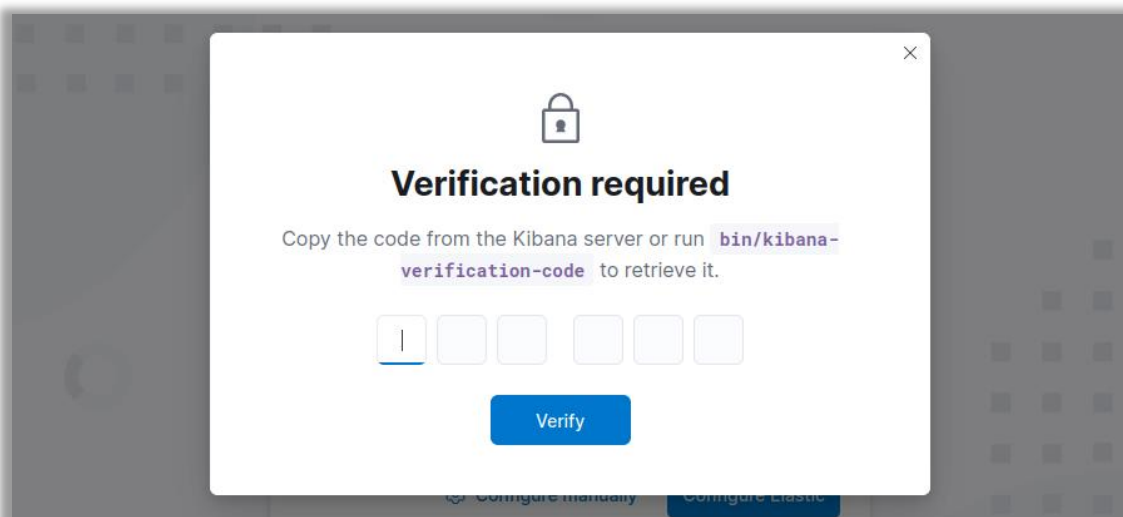
The image shows the Elastic configuration page. At the top center is the Elastic logo. Below it, the heading "Configure Elastic to get started" is displayed. A central form titled "Enrollment token" contains a text area with the placeholder "Paste enrollment token from terminal." Below the text area is a link "Where do I find this?". At the bottom of the form are two buttons: "Configure manually" with a gear icon and "Configure Elastic" in blue.

PHASE 3 : Connecting elasticsearch with kibana

A terminal window with a dark background and three colored window control buttons (red, green, yellow) at the top left. It displays a command to create an enrollment token for Kibana.

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

After getting your token and inserting it into Kibana, a verification code will be created

A modal dialog box titled "Verification required" with a lock icon. It instructs the user to copy a code from the Kibana server or run a command. Below the text is a six-digit verification code input field, with the first digit '1' already entered. A "Verify" button is at the bottom.

Copy the code from the Kibana server or run `bin/kibana-verification-code` to retrieve it.

1

Verify

Getting verification code

```
$ sudo cd /usr/share/kibana  
$ sudo ./kibana-verification-code
```



Configure Elastic to get started

- ✓ Saving settings
- ✓ Starting Elastic
- Completing setup

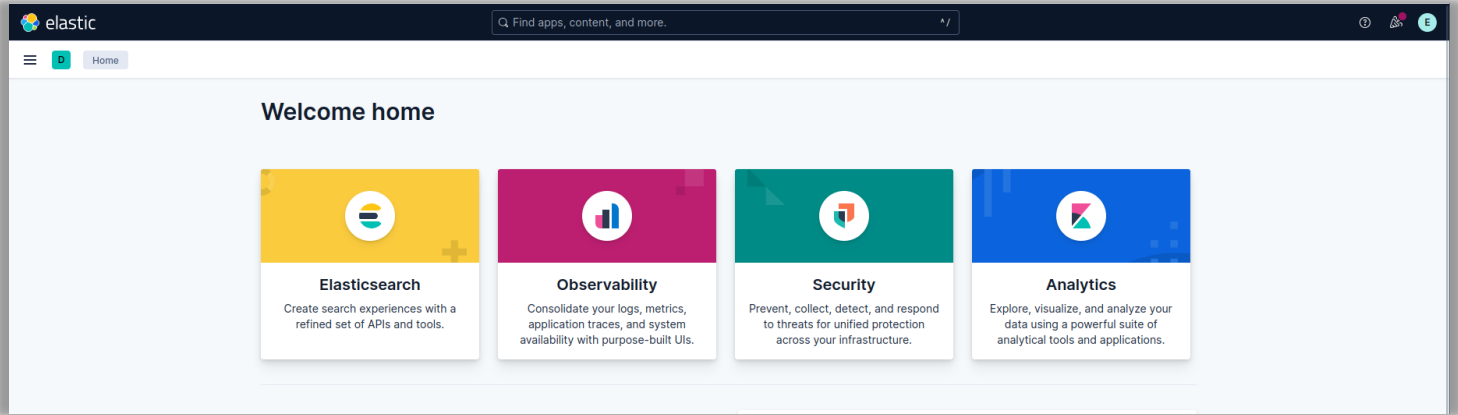


Welcome to Elastic

Username
elastic

Password

Log in

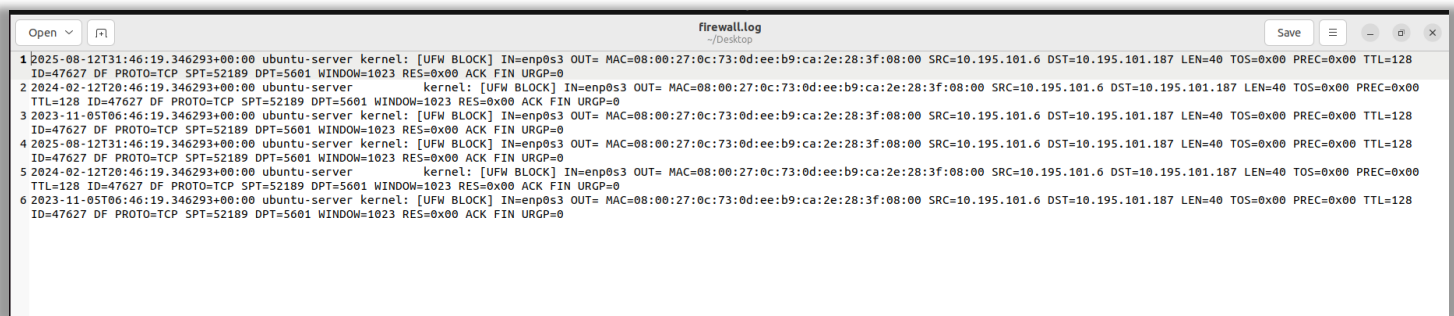


PHASE 4 : Installing & Configuring Fluentbit & sending logs

```
$ sudo apt-get update
$ sudo apt-get install fluent-bit
$ sudo systemctl enable fluent-bit
$ sudo systemctl start fluent-bit
```

For a simple simulation of logs

Create a .log file for example firewall.log and add a few logs to that file and then save it.



Configuring the fluent-bit.conf file

```
$ sudo nano /etc/fluent-bit/fluent-bit.conf
```

Allow these configurations but only change the path to the **path** where you stored the logs , change the **http_passwd** value to your current elastic password and modify the **index** as you wish.

```
[INPUT]
name tail
tag ufw_logs
path /home/omar/Desktop/firewall.log

# Read interval (sec) Default: 1
# $interval_sec 1

[OUTPUT]
name es
match *
host 127.0.0.1
port 9200
index my-ufw-logs
http_user elastic
http_passwd 4XLiS00mq6IpYMx349mW
tls On
tls.verify Off
trace_output On
suppress_type_name On
```

MAKE SURE THE WORDS ARE ALIGNED TO AVOID ANY SYNTAX ERROR AS THIS IS A SENSITIVE FILE

Configuring the parsers.conf file

```
$ sudo nano /etc/fluent-bit/parsers.conf
```

Now we have to create a parser for our log using this syntax

Parser is different from one log to another so find your format

To create & test your own regex against your logs [click here](#).

```
GNU nano 6.2 /etc/fluent-bit/parsers.conf
[PARSER]
  Name      ufw-firewall
  Format     regex
  Regex      ^(?<event_timestamp>\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2})\.\d+\.\d{2}:\d{2}).*SRC=(?<from_ip>\d+\.\d+\.\d+\.\d+)\sDST=(?<to_ip>\d+\.\d+\.\d+\.\d+)\s
  Time_Key   time
  Time_Format %Y-%m-%dT%H:%M:%S.%L%z
  Time_Keep  On

[PARSER]
  Name      apache
  Format     regex
  Regex      ^(?<host>[^\s]*) [^\s]* (?<user>[^\s]*) \[(?<time>[^\s]*)\] "(?<method>\S+)(?: +(?<path>[^\s"]*)?(?: +\S*)?)?" (?<code>[^\s]*) (?<size>[^\s]*)?(?:
  Time_Key   time
  Time_Format %d/%b/%Y:%H:%M:%S %z

[PARSER]
  Name      apache2
  Format     regex
  Regex      ^(?<host>[^\s]*) [^\s]* (?<user>[^\s]*) \[(?<time>[^\s]*)\] "(?<method>\S+)(?: +(?<path>[^\s"]*) +\S*)?" (?<code>[^\s]*) (?<size>[^\s]*)?(?: "(?<ref>[^\s"]
  Time_Key   time
  Time_Format %d/%b/%Y:%H:%M:%S %z

[PARSER]
  Name      apache_error
  Format     regex
  Regex      ^[[^\s]* (?<time>[^\s]*)\] \[(?<level>[^\s]*)\](?: \[pid (?<pid>[^\s]*)\])?( \[client (?<client>[^\s]*)\])? (?<message>.*)$
  Read 147 lines
```

Check you configuration

```
$ /opt/fluent-bit/bin/fluent-bit -c //etc/fluent-bit/fluent-bit.conf
```

```
omar@omar-virtual-machine:~$ /opt/fluent-bit/bin/fluent-bit -c //etc/fluent-bit/fluent-bit.conf
Fluent Bit v4.0.7
* Copyright (C) 2015-2025 The Fluent Bit Authors
* Fluent Bit is a CNCF sub-project under the umbrella of Fluentd
* https://fluentbit.io

[2025/08/14 23:12:04] [ info] [fluent bit] version=4.0.7, commit=, pid=7330
[2025/08/14 23:12:04] [ info] [storage] ver=1.5.3, type=memory, sync=normal, checksum=off, max_chunks_up=128
[2025/08/14 23:12:04] [ info] [simd  ] SSE2
[2025/08/14 23:12:04] [ info] [cmetrics] version=1.0.5
[2025/08/14 23:12:04] [ info] [ctraces ] version=0.6.6
[2025/08/14 23:12:04] [ info] [sp] stream processor started
[2025/08/14 23:12:04] [ info] [engine] Shutdown Grace Period=5, Shutdown Input Grace Period=2
```

Everything works and you can't view the logs in Kibana ? try adding more logs or copy & pasting the same ones to simulate a real log update

Index Management

[Index Management docs](#)

IndicesData StreamsIndex TemplatesComponent TemplatesEnrich Policies

Update your Elasticsearch indices individually or in bulk. [Learn more.](#)

☐ Include hidden indices

☐ Include rollout indices

Lifecycle status

Lifecycle phase

Reload indices

Create index

<input type="checkbox"/> Name	Health	Status	Primaries	Replicas	Documents count	Storage size	Data stream
<input type="checkbox"/> metrics-endpoint.metadata_current_default	green	open	1	0	0	249b	
<input type="checkbox"/> my-ufw-logs	yellow	open	1	1	18	17.03kb	

Rows per page: 10

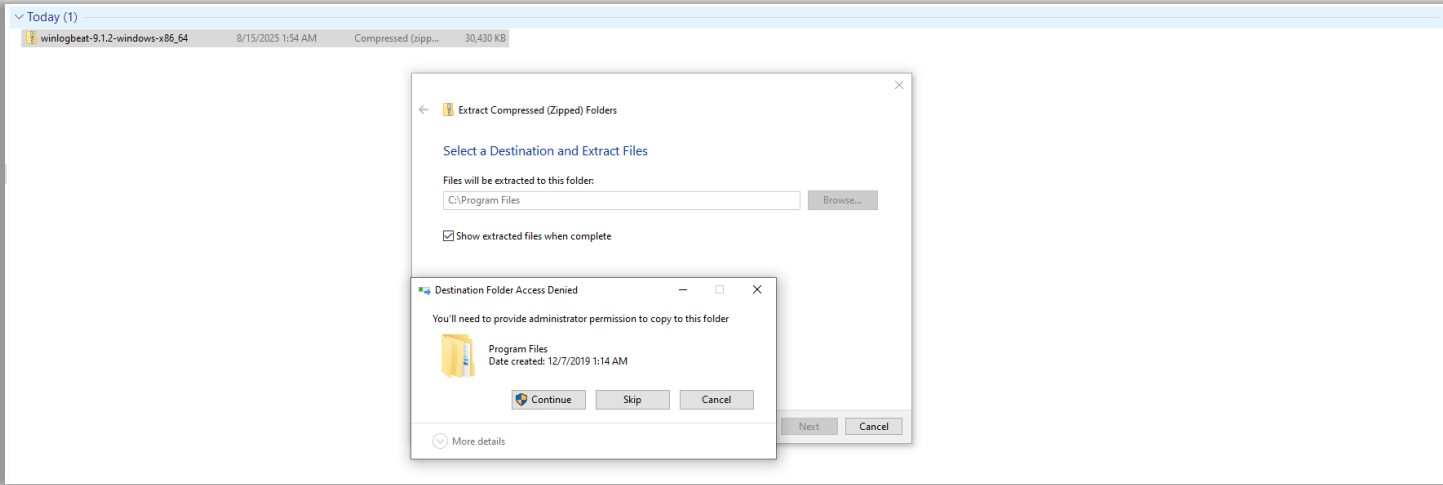
<

1

>

PHASE 5 : Installing & Configuring Winlogbeat & sending logs

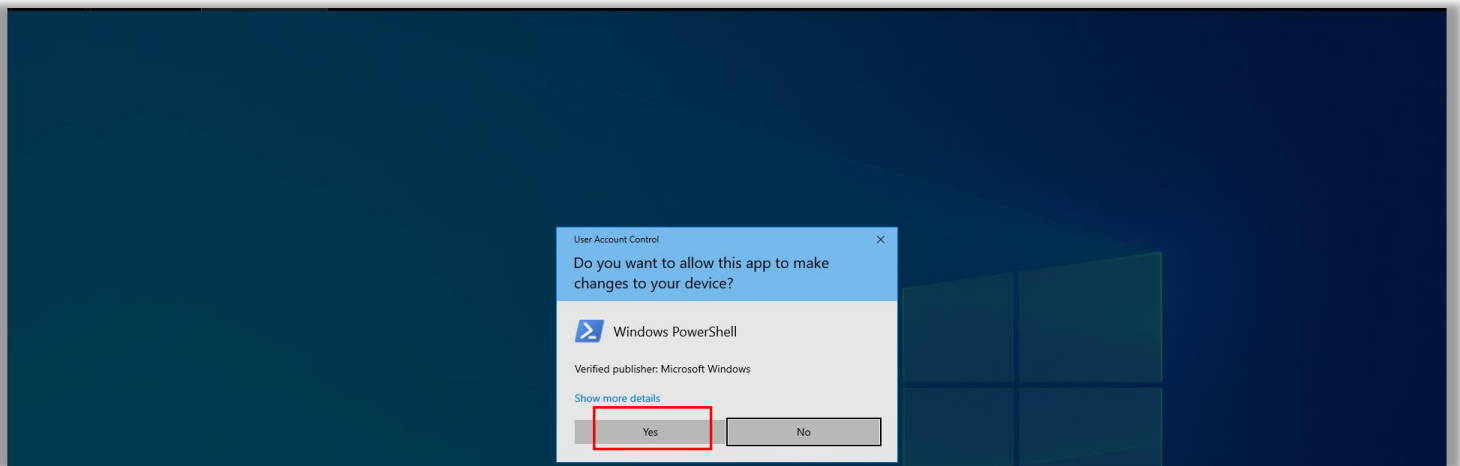
Download winlogbeat zip file from [here](#) & then extract to "C:\Program files"



Preferably rename the folder to winlogbeat instead of winlogbeat.version

Windows Security	12/7/2019 1:31 AM	File folder
WindowsPowerShell	12/7/2019 1:31 AM	File folder
Winlogbeat	8/15/2025 1:57 AM	File folder

Run Powershell as Administrator and grant permissions to edit the winlogbeat.yml file



```
Takeown /F "C:\Program Files\Winlogbeat\winlogbeat.yml" /A
```

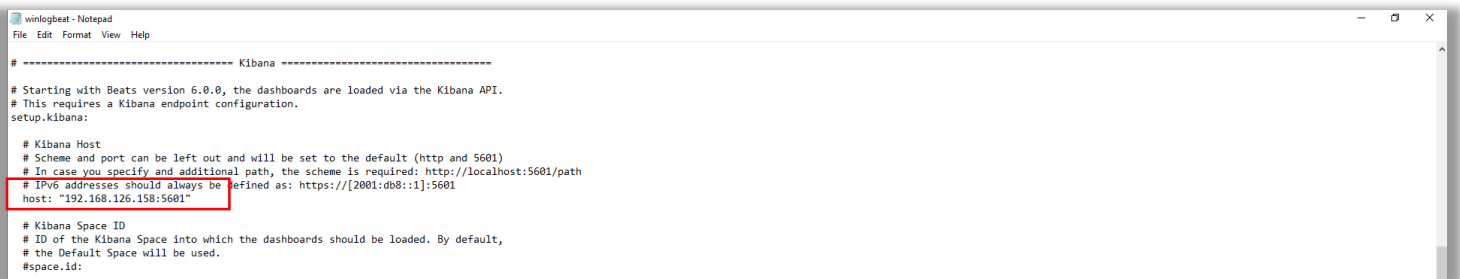
```
Iccls "C:\Program Files\Winlogbeat\winlogbeat.yml" /Grant "Administrators:F" /T
```

```
Administrator: Windows PowerShell
PS C:\Windows\system32> Takeown /F "C:\Program Files\Winlogbeat\winlogbeat.yml" /A

SUCCESS: The file (or folder): "C:\Program Files\Winlogbeat\winlogbeat.yml" now owned by the administrators group.
PS C:\Windows\system32> Iccls "C:\Program Files\Winlogbeat\winlogbeat.yml" /Grant "Administrators:F" /T
processed file: C:\Program Files\Winlogbeat\winlogbeat.yml
Successfully processed 1 files; Failed processing 0 files
PS C:\Windows\system32>
```

Now editing the winlogbeat.yml file inside the winlogbeat folder (using notepad)

Uncomment the host field under kibana and modify your IP address



Uncomment the host field under Elasticsearch Output and modify your IP address , uncomment the protocol (Only If you used https in Elasticsearch) , uncomment the username & password and adjust them , and finally add this part :

```
ssl:
  enabled: true
  certificate_authorities: ["C:/Program Files/Winlogbeat/http_ca.crt"]
```

If you don't have ssl enabled in elasticsearch then change " enabled : true " -> "enabled : false "



```
winlogbeat - Notepad
File Edit Format View Help
# Configure what output to use when sending the data collected by the beat.

# ----- Elasticsearch Output -----
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["192.168.126.158:9200"]

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

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  username: "elastic"
  password: "4Xl1S0mq6IpYMc349ml"
  ssl:
    enabled: true
    certificate_authorities: ["C:/Program Files/Winlogbeat/http_ca.crt"]

  # Pipeline to route events to security, sysmon, or powershell pipelines.
  pipeline: "winlogbeat-%[agent.version]}-routing"
```

Adding the http_ca.crt to allow connection with elasticsearch (SKIP THIS PART IF SSL IS DISABLED)

To get the certificate go to Ubuntu , go to `/etc/elasticseach/certs` and there you will find http_ca.crt , you can copy the content inside that file then paste it into a txt file on your windows machine and change the extension to .crt to transfer it using a USB , after that add it to `C:/Program Files/Winlogbeat`

MAKE SURE ELASTICSEARCH & KIBANA ARE RUNNING

```
.\winlogbeat.exe test config -c .\winlogbeat.yml -e
```

```
PS C:\Windows\system32> cd "C:\Program Files\Winlogbeat"
PS C:\Program Files\Winlogbeat> .\winlogbeat.exe test config -c .\winlogbeat.yml -e
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.814-0700","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).configure","file.name":"instance/beat.go","file.line":82},"ecs.version":"1.6.0"}
[{"C:\\Program Files\\Winlogbeat\\Data path: [C:\\Program Files\\Winlogbeat\\data] Logs path: [C:\\Program Files\\Winlogbeat\\logs"],"service.name":"winlogbeat","ecs.version":"1.6.0"}]
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.857-0700","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).configure","file.name":"instance/beat.go","file.line":83},"ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.887-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).createBeater","file.name":"instance/9.1.2 (FIPS-distribution: false)","service.name":"winlogbeat","ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.889-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance/nlogbeat","system.info":{"beat":{"path":{"config":"C:\\Program Files\\Winlogbeat","data":"C:\\Program Files\\Winlogbeat\\data","home":"C:\\Program Files\\Winlogbeat\\9ce747"},"ecs.version":"1.6.0"}}},"ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.889-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance/nlogbeat","system.info":{"build":{"commit":"b036c1c565cf24c9b7206056322344d20cb9dba60","libbeat":"9.1.2","time":"2025-08-11T13:56:43.000Z","version":"9.1.2"},"ecs.version":"1.6.0"}}},"ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.889-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance/nlogbeat","system.info":{"go":{"os":"windows","arch":"amd64","max_procs":1,"version":"go1.24.4"},"ecs.version":"1.6.0"}}},"ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.902-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance/nlogbeat","system.info":{"host":{"architecture":"x86_64","native_architecture":"x86_64","boot_time":"2025-08-15T06:43:52-07:00","name":"DESKTOP-A2JN342","ip":["192.168.126.143"],"fe80::74c0:17c9:2a36:8dbc4170 (WinBuild.160101.0800)","mac":["08:0c:29:f4:a5:29"],"0c:9a:3c:20:1a:38"},"os":{"type":"windows","family":"windows","platform":"windows","name":"Windows 10 Pro","version":"10.0","major":10,"minor":0,"patch":25209,"id":["09f0b707-fbe4-4c70-b4b6-f03c5f44afbe"],"ecs.version":"1.6.0"}}},"ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.906-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance/nlogbeat","system.info":{"process":{"cwd":"C:\\Program Files\\Winlogbeat","exe":"C:\\Program Files\\Winlogbeat\\winlogbeat.exe","name":"winlogbeat.exe","pid":2108,"ppid":5056,"start_time":"2025-08-15T07:34:20.906-0700"},"ecs.version":"1.6.0"}}},"ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.990-0700","log.logger":"elasticsearch.esclientleg","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/esleg/eslegclient.NewConnection","file.name":"elasticsearch/connection.go","file.line":132},"message":"Elasticsearch url: https://192.168.126.10:9200","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.990-0700","log.logger":"publisher","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/publisher/pipeline.LoadWithSettings","file.name":"publisher/pipeline/load.go","file.line":100},"message":"Loading index template","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.993-0700","log.logger":"winlogbeat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/beater.New","file.name":"beater/winlogbeat.go","file.line":100},"message":"Loaded index template","service.name":"winlogbeat","ecs.version":"1.6.0"}
Config OK
PS C:\Program Files\Winlogbeat>
```

```
.\winlogbeat.exe setup -e
```

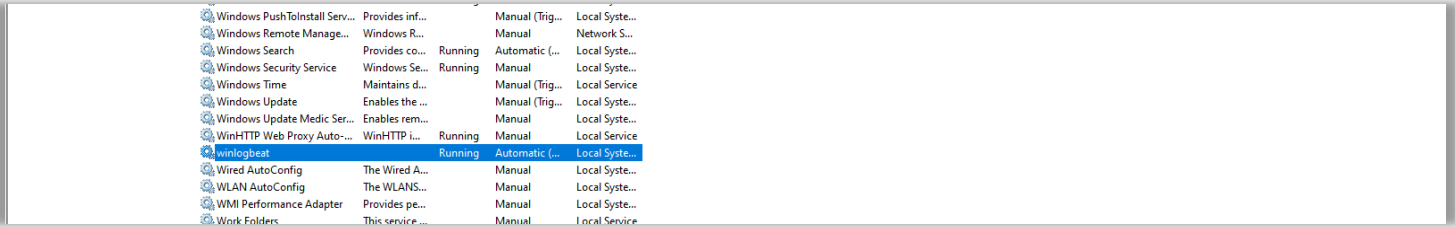
```
Select Administrator Windows PowerShell
{"log.level":"info","@timestamp":"2025-08-16T00:04:03.142-0700","log.logger":"template","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/template.(*TemplateBuilder).buildBody","file.name":"template/load.go","file.line":262},"message":"Existing template will be overwritten, as overwrite is enabled.","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:03.903-0700","log.logger":"template_loader","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/template.(*ESLoader).loadTemplate","file.name":"template/load.go","file.line":176},"message":"Try loading template winlogbeat-9.1.2 to Elasticsearch","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.078-0700","log.logger":"template_loader","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/template.(*ESLoader).load","file.name":"template/load.go","file.line":133},"message":"Template with name \\winlogbeat-9.1.2\\ loaded.","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.092-0700","log.logger":"template_loader","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/template.(*ESLoader).load","file.name":"template/load.go","file.line":149},"message":"Data stream with name \\winlogbeat-9.1.2\\ already exists.","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.108-0700","log.logger":"index-management","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/idxmgmt.(*IndexManager).Setup","file.name":"idxmgmt/index_support.go","file.line":299},"message":"Loaded index template.","service.name":"winlogbeat","ecs.version":"1.6.0"}
Index setup finished.
Loading dashboards (Kibana must be running and reachable)
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.149-0700","log.logger":"kibana","log.origin":{"function":"github.com/elastic/elastic-agent-libs/kibana.NewClientWithConfigDefault","file.name":"kibana/client.go","file.line":181},"message":"Kibana url: http://192.168.126.158:5601","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:05.122-0700","log.logger":"kibana","log.origin":{"function":"github.com/elastic/elastic-agent-libs/kibana.NewClientWithConfigDefault","file.name":"kibana/client.go","file.line":181},"message":"Kibana url: http://192.168.126.158:5601","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.925-0700","log.logger":"processors.add_cloud_metadata","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/processors/add_cloud_metadata.(*addCloudMetadata).init.func1","file.name":"add_cloud_metadata/add_cloud_metadata.go","file.line":100},"message":"add_cloud_metadata: hosting provider type not detected.","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:40.318-0700","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).loadDashboards","file.name":"instance/beat.go","file.line":1070},"message":"Kibana dashboards successfully loaded.","service.name":"winlogbeat","ecs.version":"1.6.0"}
Loaded dashboards
{"log.level":"info","@timestamp":"2025-08-16T00:04:40.331-0700","log.logger":"esclientleg","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/esleg/eslegclient.NewConnection","file.name":"eslegclient/connection.go","file.line":132},"message":"Elasticsearch url: https://192.168.126.158:9200","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:40.390-0700","log.logger":"esclientleg","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/esleg/eslegclient.(*Connection).Ping","file.name":"eslegclient/connection.go","file.line":324},"message":"Attempting to connect to Elasticsearch version 9.1.2 (default)","service.name":"winlogbeat","ecs.version":"1.6.0"}
Loaded ingest pipelines
PS C:\Program Files\Winlogbeat>
```

winlogbeat test output

```
Administrator: Windows PowerShell
PS C:\Program Files\Winlogbeat> .\winlogbeat.exe test output
elasticsearch: https://192.168.126.158:9200...
parse url... OK
connection...
parse host... OK
dns lookup... OK
addresses: 192.168.126.158
dial up... OK
TLS...
security: server's certificate chain verification is enabled
handshake... OK
TLS version: TLSv1.3
dial up... OK
talk to server... OK
version: 9.1.2
PS C:\Program Files\Winlogbeat>
```

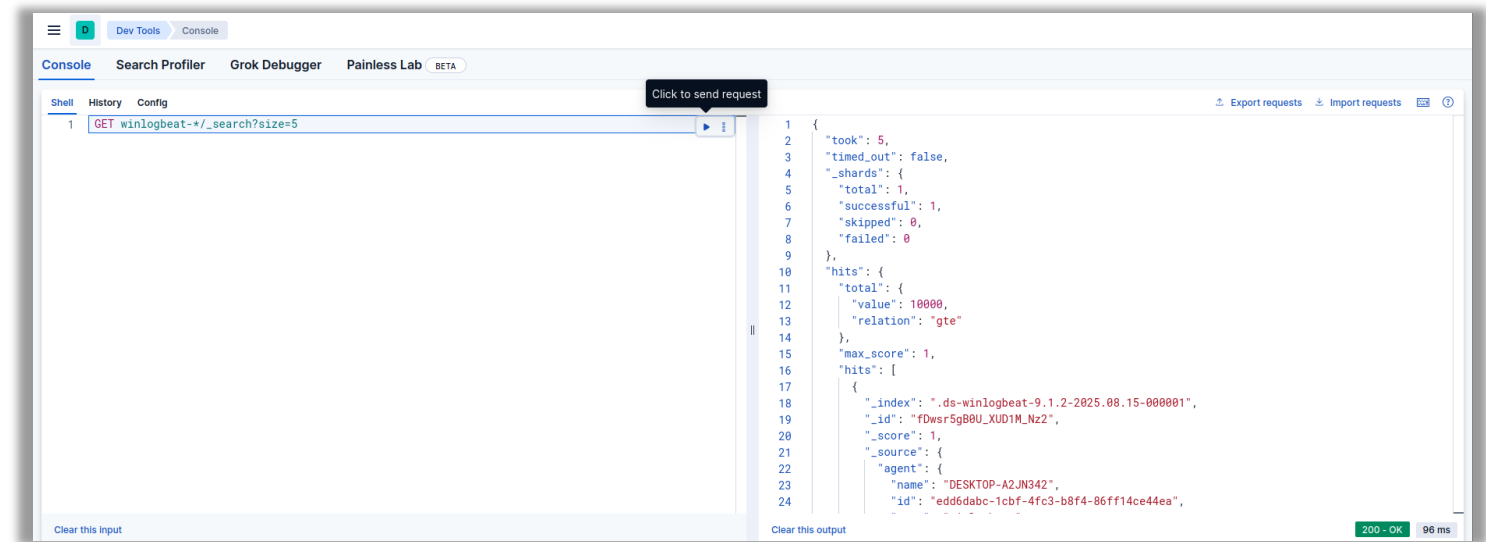
Start-Service winlogbeat

And then you can check if it's running using services.msc or `Get-Service winlogbeat`

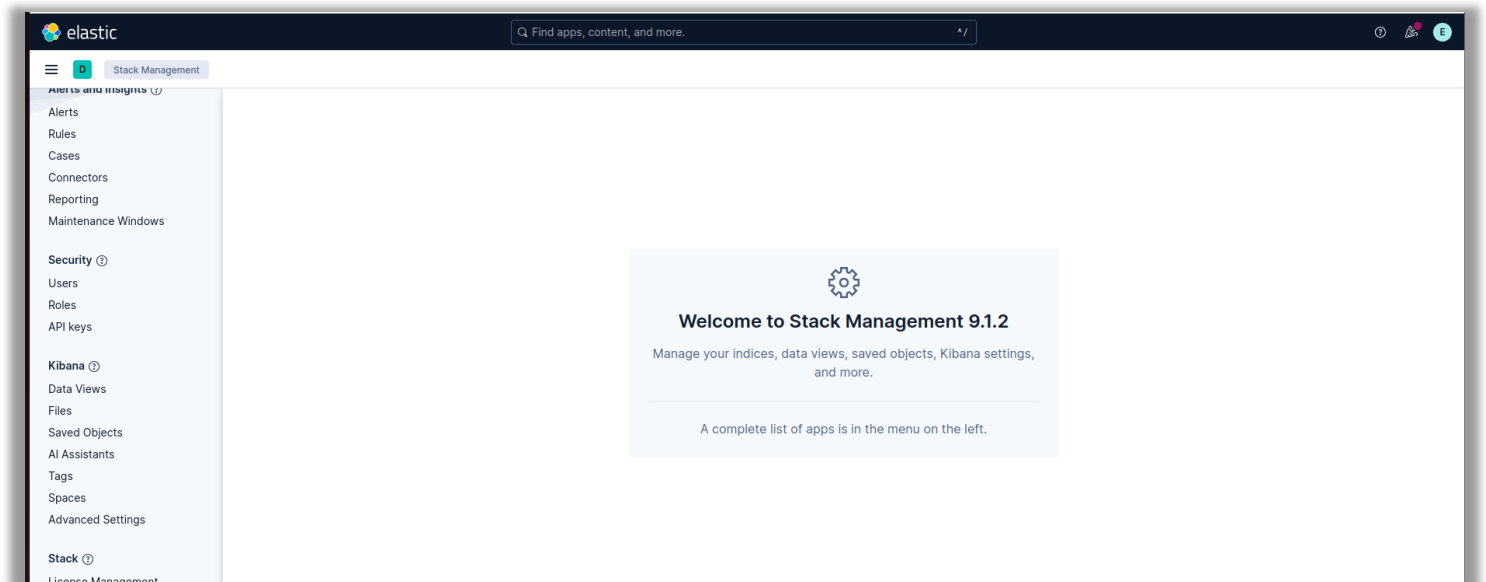


Viewing the logs using kibana dashboard

Testing first using Dev tools : `GET winlogbeat-*/_search?size=5`



Stack Management > Data View



Index Pattern : winlogbeat-*

Select @timestamp as the time filter

The screenshot shows the 'Create data view' dialog in the Elastic Data Views interface. The 'Name' field is set to 'winlogbeat-*'. The 'Index pattern' field is also set to 'winlogbeat-*'. The 'Timestamp field' dropdown is set to '@timestamp'. A message at the top right states 'Your index pattern matches 1 source.' Below this, a table shows 'winlogbeat-9.1.2' as the 'Matching source'. The 'Rows per page' is set to 10. A 'Save data view to Kibana' button is at the bottom right.

Create data view

Name: winlogbeat-*

Index pattern: winlogbeat-*

Timestamp field: @timestamp

Rows per page: 10

Save data view to Kibana

Confirm logs and select the winlogbeat-* index

The screenshot shows the Elastic Discover interface. The 'Data view' is set to 'winlogbeat-*'. The KQL query is '@timestamp'. The 'Auto interval' is set to 'No breakdown'. The 'Documents (4)' tab is selected, showing a list of documents. The first document is a security event from 'DESKTOP-A2JN342'.

Discover

Filter your data using KQL syntax: @timestamp

Documents (4)

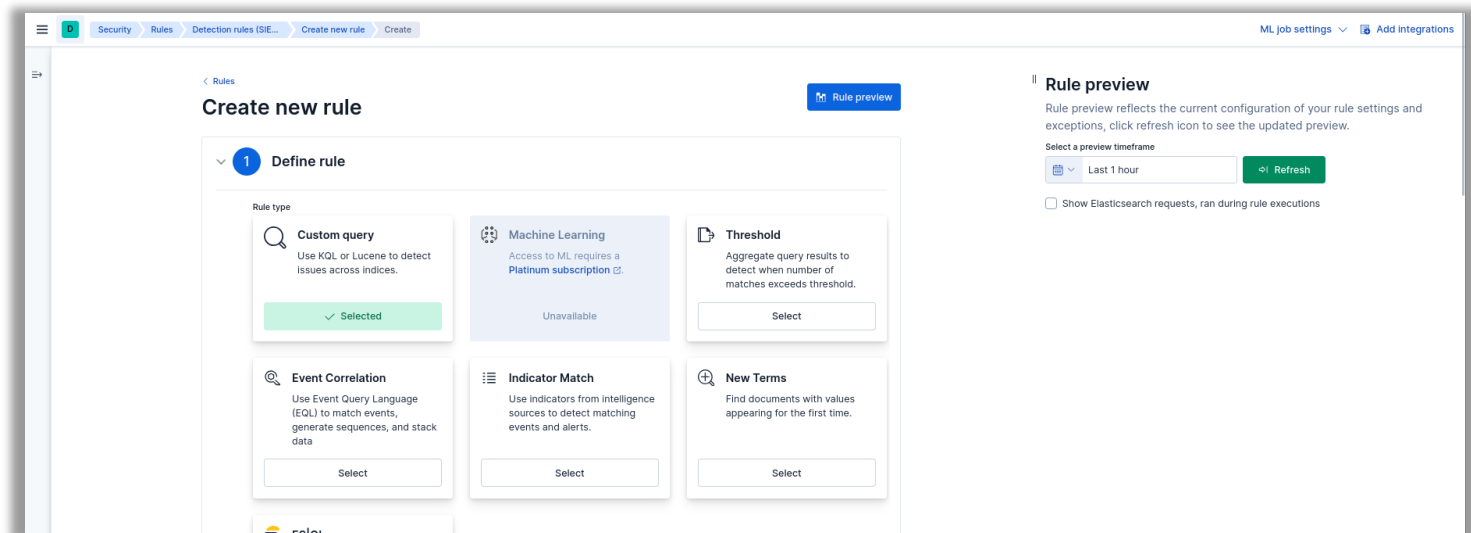
Document
<pre>@timestamp Aug 15, 2025 @ 22:25:38.816 agent.name DESKTOP-A2JN342 agent.type winlogbeat agent.version 9.1.2 ecs.version 8.17.0 event.action logged-in-special event.category iam event.code 4672 event.created Aug 15, 2025 @ 22:25:40.534 event.ingested Aug 15, 2025 @ 22:25:54.828 event.kind event event.module security event.outcome success event.provider Microsoft-Windows-Security-Auditing-</pre>
<pre>@timestamp Aug 15, 2025 @ 22:25:38.816 agent.name DESKTOP-A2JN342 agent.type winlogbeat agent.version 9.1.2 ecs.version 8.17.0 event.action logged-in event.category authentication event.code 4624 event.created Aug 15, 2025 @ 22:25:40.534 event.ingested Aug 15, 2025 @ 22:25:54.828 event.kind event event.module security event.outcome success event.provider Microsoft-Windows-Security-Audit-</pre>
<pre>@timestamp Aug 15, 2025 @ 22:25:38.787 agent.name DESKTOP-A2JN342 agent.type winlogbeat agent.version 9.1.2 ecs.version 8.17.0 event.action logged-in-special event.category iam event.code 4672 event.created Aug 15, 2025 @ 22:25:40.529 event.ingested Aug 15, 2025 @ 22:25:54.827 event.kind event event.module security event.outcome success event.provider Microsoft-Windows-Security-Audit-</pre>
<pre>@timestamp Aug 15, 2025 @ 22:25:38.787 agent.name DESKTOP-A2JN342 agent.type winlogbeat agent.version 9.1.2 ecs.version 8.17.0 event.action logged-in event.category authentication event.code 4624 event.created Aug 15, 2025 @ 22:25:40.529 event.ingested Aug 15, 2025 @ 22:25:54.827 event.kind event event.module security event.outcome success event.provider Microsoft-Windows-Security-Audit-</pre>

PHASE 6 : Threat detection rules & simulating suspicious activity

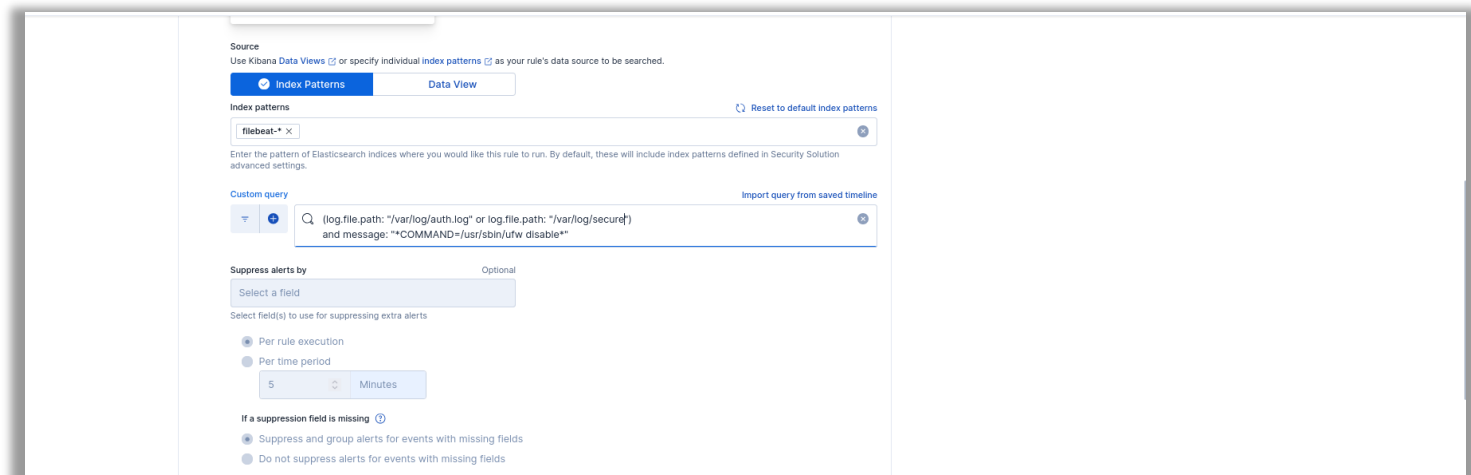
To demonstrate how security monitoring works, we'll simulate a basic but suspicious activity: an attacker disabling the Linux firewall (UFW).

In real-world scenarios, detection rules are far more sophisticated , incorporating behavioral analysis, anomaly detection, and threat intelligence . But this simplified example helps illustrate the core principles.

Security>rules>detection rules (SIEM) > create new rule



Choosing the wanted index pattern and writing our custom query (KQL) to detect the disabling of ufw firewall , **make sure to test your query in discover first.**



Specifying a name , description , selecting the severity and risk score

2

About rule

Name

Unauthorized Firewall Disable Attempt Detected

Description

This alert triggers when system logs indicate the local firewall service (UFW) has been stopped or disabled on the SIEM host (omar@omar-virtual-machine). Firewall tampering is a common attacker technique to remove network security controls and enable lateral movement. This activity matches MITRE ATT&CK technique T1562.001 (Impair Defenses: Disable or Modify System Firewall).

Default severity

Select a severity level for all alerts generated by this rule.

High

Severity override

☐ Use source event values to override the default severity.

Default risk score

Select a risk score for all alerts generated by this rule.

0255075100

73

Risk score override

☐ Use a source event value to override the default risk score.

Tags

T1562.001

Optional

Leave on Default

Tags

T1562.001

3

Schedule rule

Runs every

5 Minutes

Rules run periodically and detect alerts within the specified time frame.

Additional look-back time

1 Minutes

Adds time to the look-back period to prevent missed alerts.

Continue

Testing using the command : `sudo ufw disable`

```
omar@omar-virtual-machine:~$ sudo ufw disable
[sudo] password for omar:
Firewall stopped and disabled on system startup
omar@omar-virtual-machine:~$
```

View alerts through security>alerts

