Beginner Guide: Running the BCC DNS 2024 Dataset in a Containerized ELK Stack

1 Overview

This guide provides a step-by-step walkthrough for setting up a beginner-friendly experiment using the BCC DNS 2024 dataset within a fully containerized ELK (Elasticsearch, Logstash, Kibana) stack. The goal is to ingest labeled DNS traffic data into Elasticsearch using Logstash, explore it in Kibana, and integrate connectivity with n8n triage and agent workflows.

All components are deployed using Docker Compose, and configuration files are kept minimal and explicit to support learning and modification.

2 Prerequisites

Before starting, ensure the following are installed on your system:

- Docker Desktop (Windows, macOS, or Linux)
- Docker Compose v2
- At least 16 GB of RAM recommended
- Sufficient disk space for Elasticsearch indices (50+ GB recommended)

Verify Docker installation:

```
docker --version
docker compose version
```

3 Project Directory Structure

Create a new project directory (example: `bcc-dns-elk`) with the following structure:

```
bcc-dns-elk/
 docker-compose.yml
 logstash/
    bcc_dns_2024.conf
 data/
```

```
    dns/
        output-of-benign-*.csv
        output-of-malware-*.csv
        output-of-phishing-*.csv
        output-of-spam-*.csv
```

The `data/dns` directory will contain the CSV files from the BCC DNS 2024 dataset.

4 Docker Compose Stack

The ELK stack consists of three core services:

- Elasticsearch (data storage and indexing)
- Logstash (CSV ingestion and labeling)
- Kibana (visualization and exploration)

4.1 docker-compose.yml

Create a `docker-compose.yml` file with the following contents:

```
services:

elasticsearch:
image: docker.elastic.co/elasticsearch/elasticsearch:8.19.2
container_name: elasticsearch
environment:
- discovery.type=single-node
- xpack.security.enabled=false
- xpack.ml.enabled=false
- ES_JAVA_OPTS=-Xms4g -Xmx4g
volumes:
- esdata:/usr/share/elasticsearch/data
ports:
- "9200:9200"
restart: unless-stopped

logstash:
image: docker.elastic.co/logstash/logstash:8.19.2
container_name: logstash
volumes:
- ./logstash/bcc_dns_2024.conf:/usr/share/logstash/pipeline/bcc_dns_2024.conf
- ./data/dns:/data
depends_on:
- elasticsearch
```

```
restart: unless-stopped

kibana:
image: docker.elastic.co/kibana/kibana:8.19.2
container_name: kibana
environment:
- ELASTICSEARCH_HOSTS=[http://elasticsearch:9200](http://elasticsearch:9200)
ports:
- "5601:5601"
depends_on:
- elasticsearch
restart: unless-stopped

volumes:
esdata:
```

5 Logstash Pipeline Configuration

Logstash is responsible for ingesting CSV files, assigning labels based on filenames, parsing timestamps, and indexing records into Elasticsearch.

5.1 $\mathbf{bcc}_{d}ns_{2}024.conf$

Place the following configuration file in the `logstash/` directory:

```
input {
file {
path => "/data/output-of-*.csv"
start_position => "beginning"
sincedb_path => "/dev/null"
mode => "read"
}
}

filter {
csv {
autodetect_column_names => true
skip_header => true
}

ruby {
code => '
p = event.get("[log][file][path]") || event.get("path")
if p
```

```
if p.include?("output-of-malware")
event.set("label", "malware"); event.set("label_family","malicious"); event.set("
    is_malicious", true)
elsif p.include?("output-of-phishing")
event.set("label", "phishing"); event.set("label_family","malicious"); event.set("
    is_malicious", true)
elsif p.include?("output-of-spam")
event.set("label", "spam"); event.set("label_family","malicious"); event.set("
    is_malicious", true)
elsif p.include?("output-of-benign")
event.set("label", "benign"); event.set("label_family","benign"); event.set("is_malicious
    ", false)
end
end
'
}

mutate {
add_field => {
"event.dataset" => "cic.dns_2024"
"data_source" => "BCC-CIC DNS Dataset (2024)"
}
}

if [timestamp] {
date {
match => ["timestamp", "ISO8601", "yyyy-MM-dd HH:mm:ss", "yyyy-MM-dd HH:mm:ss.SSS"]
target => "@timestamp"
timezone => "UTC"
}
}
}

output {
elasticsearch {
hosts => ["http://elasticsearch:9200"]
index => "cic-dns-2024"
}
}
```

6 Starting the Stack

From the project root directory, start all services:

```
docker compose up -d
```

Initial startup may take several minutes while Elasticsearch initializes.

7 Verification

Confirm all containers are running:

```
docker compose ps
```

Check Logstash ingestion logs:

```
docker compose logs -f logstash
```

8 Accessing Kibana

Open a web browser and navigate to:

`http://localhost:5601`

Create a data view using the index pattern:

`cic-dns-2024`

Set `@timestamp` as the time field when prompted.

9 Next Steps

Once data is visible in Kibana, users can:

- Explore DNS traffic distributions
- Compare benign vs malicious activity
- Test searches and build dashboards
- Switch over to n8n for Elastic query intergration

This environment provides a clean, reproducible baseline for DNS security experimentation using labeled benchmark data.