Fluent-Bit Documentation

Yehia A. Mostafa

---

1. Installation of Fluent Bit

Follow the official Fluent Bit installation guide for Ubuntu:

🔗 Fluent Bit Installation on Ubuntu

Important:

- **Do NOT** install via `snap install fluent-bit` — the snap version is outdated and may cause plugin/config issues.
- The installation from the official repository installs Fluent Bit in `/opt/fluent-bit/bin/fluent-bit`.

---

2. Configuring Parsers

By default, the main config file is `/etc/fluent-bit/fluent-bit.conf`.

Inside it, look for:

```
parsers_file parsers.con
```

This means Fluent Bit will load custom parsers from `/etc/fluent-bit/parsers.conf`.

We will add a parser for UFW firewall logs.

Parser Definition (`/etc/fluent-bit/parsers.conf`):

```
[PARSER]
Name   UFW-PARSE
Format regex
Regex ^(?<Timestamp>\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{6}\+\d{2}:\d{2})\s+(?<Hostname>\S+)\s+\S+\s+\S+\s+(?<Action>\w+)\W+\S+\s+\S+\s+\w+\W+(?<MAC>(\d+|\S+):(\d+|\S+):(\d+|\S+):(\d+|\S+):(\d+|\S+))\s+\w+\W+(?<SR
C_IP>\d+.\d+.\d+.\d+)\s+\w+\W+(?<DST_IP>\d+.\d+.\d+.\d+)\s+\w+\W+\w+\s+\w+\W+\w+\s+\w+\W+\w+\s+\w+\W+\w+\s+\w+\W+\w+\s+\w+\s+\w+\W+(?<Protocol>\w+)\s+\w+\W+(?<Source_Port>\d+)\s+\w+\W+(?<Destination_Port>\d+)
```

3. Creating a Test Log File

Make a file containing UFW logs:

```
nano /home/ubuntuserver/ufw.txt
```

Paste in some UFW logs for testing.

4. Configuring Fluent Bit Input

We will use the **tail** plugin to read the `ufw.txt` file in real time.

Example `fluent-bit.conf` section:

```
[INPUT]
    Name            tail
    Tag            ufw.logs
    Path            /home/ubuntuserver/ufw.txt
    Parser          UFW-PARSE
    DB             /var/log/flb_ufw.db
    Mem_Buf_Limit     5MB
    Refresh_Interval  5
```

5. Running Fluent Bit for Testing

Run:

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

Then, in another terminal:

- Cut and re-paste the contents of `ufw.txt` to simulate new log lines.
- `tail` reads only *new lines* at the end of the file, so this is necessary for testing.

6.1 Sending Logs to terminal/console

```
[OUTPUT]
    Name        stdout
    Match       *
```

6.2 Sending Logs to Elasticsearch/Kibana

Add this output section to `fluent-bit.conf`:

```
[OUTPUT]
    Name            es
  Match           *
  Host          192.168.29.132
  Port          9200
  Index         fluent-bit-ufw
  Suppress_Type_Name On
  HTTP_User       elastic
  HTTP_Passwd      Yehia5050
  tls          On
  tls.verify     Off
```

Notes:

- `Suppress_Type_Name On` removes the deprecated `_type` field (required for ES 7+).
- The `Index` name should match the one you create in Kibana.

7. Kibana Setup

1. Go to **Stack Management → Index Management → Create Index** → Name it (e.g., `fluent-bit-ufw`).

2. Go to **Discover → Data Views → Create Data View** and use the same index name.
3. Now your logs will appear in Kibana under that data view.

8. Elasticsearch API (devtools) Index Creation for Security Logs

We can also manually create an index for security events (SSH failures, firewall blocks, etc.):

```
PUT security-logs
{
  "mappings": {
    "properties": {
      "@timestamp": { "type": "date" },
      "source.ip": { "type": "ip" },
      "destination.ip": { "type": "ip" },
      "destination.port": { "type": "integer" },
      "user.name": { "type": "keyword" },
      "event.module": { "type": "keyword" },
      "event.action": { "type": "keyword" },
      "message": { "type": "text" },
      "rule.name": { "type": "keyword" },
      "network.bytes": { "type": "long" }
    }
  }
}
```

9. Adding Test Security Events

```
POST /security-logs/_create/1
{
  "@timestamp": "2025-08-12T08:00:00Z",
  "source.ip": "203.0.113.10",
  "destination.ip": "192.168.1.10",
  "destination.port": 22,
  "user.name": "admin",
  "event.module": "auth",
  "event.action": "ssh_login_failure",
  "message": "Failed password for admin from 203.0.113.10 port 52213 ssh2",
  "network.bytes": 350
}

POST /security-logs/_create/3
{
  "@timestamp": "2025-08-12T08:05:00Z",
  "source.ip": "198.51.100.25",
  "destination.ip": "192.168.1.20",
  "destination.port": 443,
  "user.name": "-",
  "event.module": "firewall",
  "event.action": "block",
  "rule.name": "BLOCK",
  "message": "Firewall blocked access from 198.51.100.25 to 192.168.1.20 port 443",
  "network.bytes": 0
}
```

10. Querying the Data

Simple GET queries (query string search):

```
GET /security-logs/_search?q=event.action:ssh_login_failure
GET /security-logs/_search?q=event.action:block
```

Or Advanced JSON queries:

```
GET /security-logs/_search
{
  "query": {
    "match": { "event.action": "ssh_login_failure" }
  }
}

GET /security-logs/_search
{
  "query": {
    "match": { "event.action": "block" }
  }
}
```

---

https://www.elastic.co/docs/api/doc/elasticsearch/