CSE 406 Computer Security Project - Graylog

1805049 - Somonindro Roy
1805031 - Nahian Shabab

August 2023

1 Overview

A centralized Log Management System (LMS) to aggregate, organize, and make sense of data from various sources like devices, applications, and operating systems

2 Core Features

2.1 Streams

Streams operate as a form of tagging for incoming messages. Streams route messages into categories in real time, and team rules instruct Graylog to route messages into the appropriate stream.Streams are used to route data for storage into an index. They are also used to control access to data, and route messages for parsing, enrichment, and other modification. Streams then determine which messages to archive.

2.2 Search

The Graylog Search page is the interface used to search logs directly. Graylog uses a simplified syntax, very similar to Lucene. Relative or absolute time ranges are configurable from drop down menus. Searches may be saved or visualized as dashboard widgets that may be added directly to dashboards from within the search screen.Users may configure their own views and may choose to see either a summary or complete data from event messages.

2.3 Dashboards

Graylog Dashboards are visualizations or summaries of information contained in log events. Each dashboard is populated by one or more widgets. Widgets visualize or summarize event log data with data derived from field values such as counts, averages, or totals. Users can create indicators, charts, graphs, and maps to visualize the data. Dashboard widgets and dashboard layouts are configurable. Graylog's role-based access controls dashboard access. Users can import and export dashboards via content packs.

2.4 Alerts

Alerts are created using Event Definitions that consist of Conditions. When a given condition is met it will be stored as an Event and can be used to trigger a notification.

2.5 Content Pack

Content packs accelerate the set-up process for a specific data source. A content pack can include inputs/extractors, streams, dashboards, alerts, and pipeline processors. For example, users can create custom inputs, streams, dashboards, and alerts to support a security use case. Users can then export the content pack and import it on a newly installed Graylog instance to save configuration time and effort. Users may download content packs which are created, shared and supported by other users via the Graylog Marketplace.

2.6 Index

An Index is the basic unit of storage for data in OpenSearch and Elasticsearch. Index sets provide configuration for retention, sharding, and replication of the stored data.Values, like retention and rotation strategy, are set on a per-index basis, so different data may be subjected to different handling rules.

2.7 Sidecar

Graylog Sidecar is an agent to manage fleets of log shippers, like Beats or NXLog. These log shippers are used to collect OS logs from Linux and Windows servers. Log shippers read logs written locally to a flat file, and then send them to a centralized log management solution. Graylog supports management of any log shipper as a backend.

2.8 Processing Pipeline

Graylog's Processing Pipelines enable the user to run a rule, or a series of rules, against a specific type of event. Tied to streams, pipelines allow routing, denylisting, modification, and enrichment of messages as they flow through Graylog.

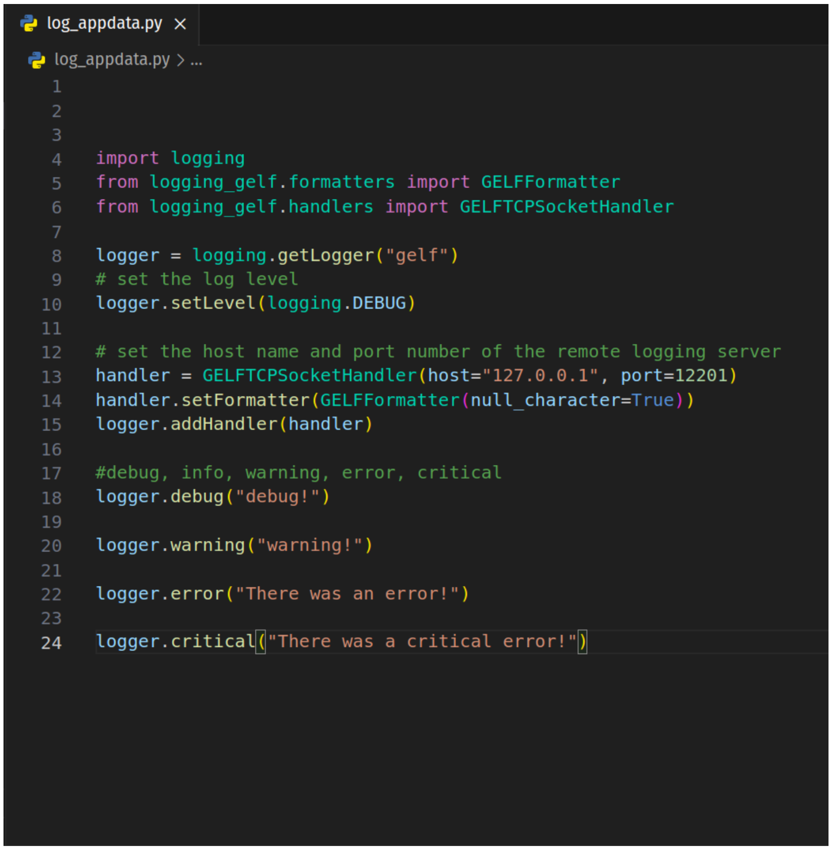
3 Feature Demonstration

3.1 Logging From Python

3.1.1 Installing packages

You have to install logging_gelf package by the command ”pip install logging” in terminal.

3.1.2 Python Code

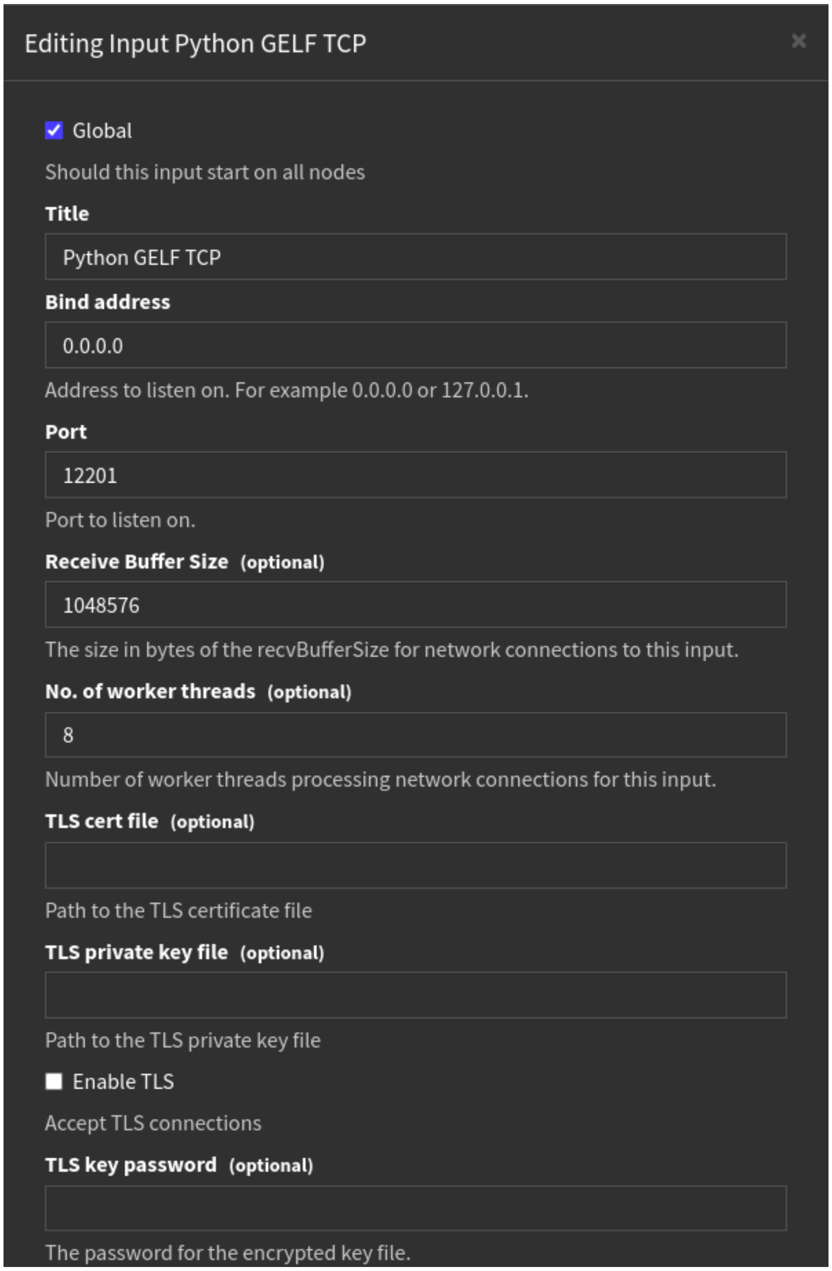


```python



import logging
from logging_gelf.formatters import GELFFormatter
from logging_gelf.handlers import GELFTCPSocketHandler

logger = logging.getLogger("gelf")
# set the log level
logger.setLevel(logging.DEBUG)

# set the host name and port number of the remote logging server
handler = GELFTCPSocketHandler(host="127.0.0.1", port=12201)
handler.setFormatter(GELFFormatter(null_character=True))
logger.addHandler(handler)

#debug, info, warning, error, critical
logger.debug("debug!")

logger.warning("warning!")

logger.error("There was an error!")

logger.critical("There was a critical error!")
```

Figure 1: Python Code

3.1.3 Setting up Input

You have to set up input configuration.



Editing Input Python GELF TCP

☑ Global

Should this input start on all nodes

Title

Python GELF TCP

Bind address

0.0.0.0

Address to listen on. For example 0.0.0.0 or 127.0.0.1.

Port

12201

Port to listen on.

Receive Buffer Size (optional)

1048576

The size in bytes of the recvBufferSize for network connections to this input.

No. of worker threads (optional)

8

Number of worker threads processing network connections for this input.

TLS cert file (optional)

Path to the TLS certificate file

TLS private key file (optional)

Path to the TLS private key file

□ Enable TLS

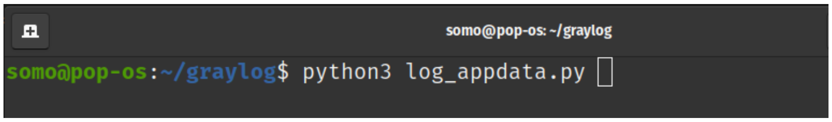
Accept TLS connections

TLS key password (optional)

The password for the encrypted key file.

Figure 2: Input Configuration

3.1.4 Running the script

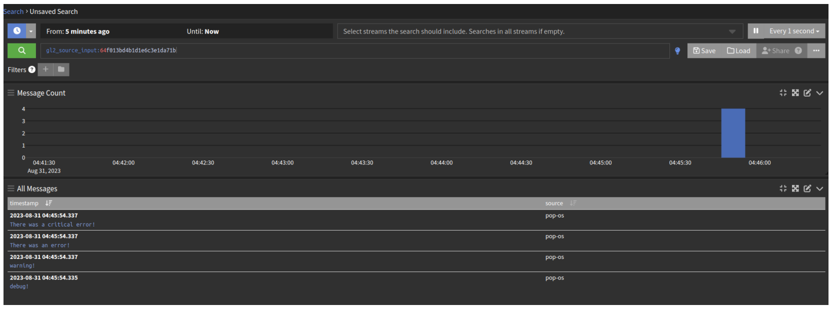


```
somo@pop-os:~/graylog$ python3 log_appdata.py
```

Figure 3: Running the script

3.1.5 Message Logged in Graylog

You will see the following messages in Graylog:



From: **5 minutes ago** Until: **Now**

Select streams the search should include. Searches in all streams if empty.

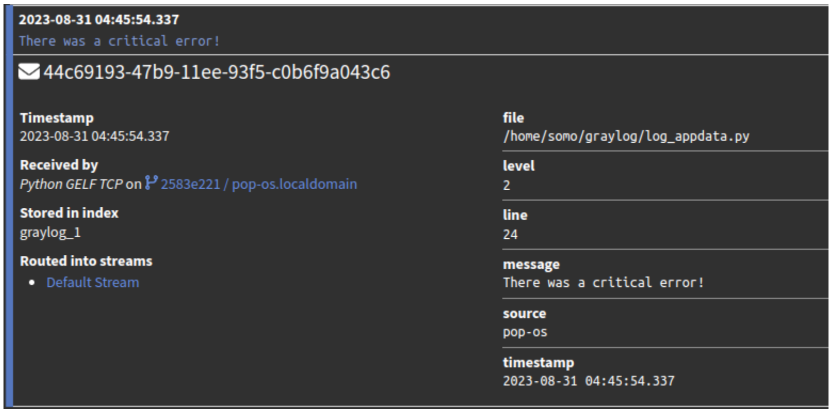
Every 1 second

```
gl2_source_input:64f013bd4b1d1e6c3e1da71b
```

timestamp	source
2023-08-31 04:45:54.337 There was a critical error!	pop-os
2023-08-31 04:45:54.337 There was an error!	pop-os
2023-08-31 04:45:54.337 warning!	pop-os
2023-08-31 04:45:54.335 debug!	pop-os

Figure 4: Messages logged in graylog

The details of the message can also be shown. Details include name of the file, the line number , timestamp etc.



2023-08-31 04:45:54.337

There was a critical error!

44c69193-47b9-11ee-93f5-c0b6f9a043c6

Timestamp
2023-08-31 04:45:54.337

Received by
Python GELF TCP on 2583e221 / pop-os.localdomain

Stored in index
graylog_1

Routed into streams

- Default Stream

file
/home/somo/graylog/log_appdata.py

level
2

line
24

message
There was a critical error!

source
pop-os

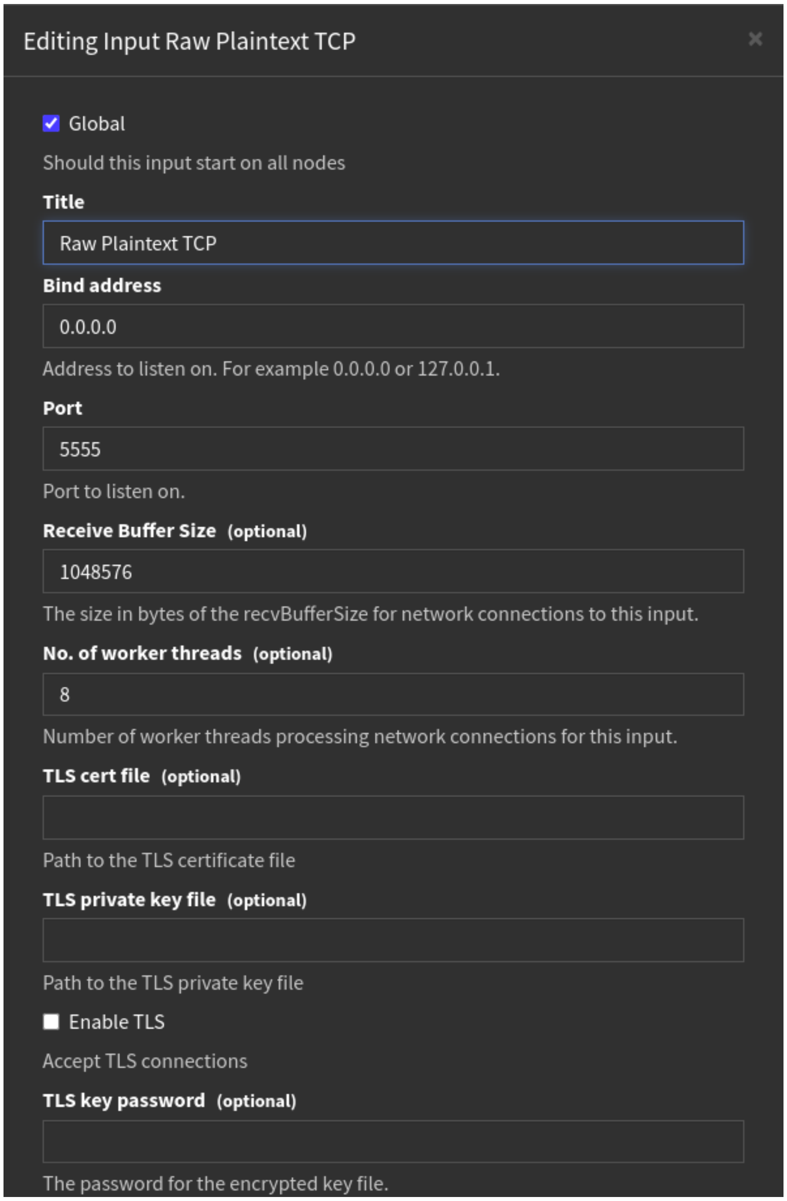
timestamp
2023-08-31 04:45:54.337

Figure 5: Message Details

3.2 Raw PlainText TCP

3.2.1 Setting up input

Here, you have to set up input for raw plaintext TCP.



Editing Input Raw Plaintext TCP

☑ Global

Should this input start on all nodes

Title

Raw Plaintext TCP

Bind address

0.0.0.0

Address to listen on. For example 0.0.0.0 or 127.0.0.1.

Port

5555

Port to listen on.

Receive Buffer Size (optional)

1048576

The size in bytes of the recvBufferSize for network connections to this input.

No. of worker threads (optional)

8

Number of worker threads processing network connections for this input.

TLS cert file (optional)

Path to the TLS certificate file

TLS private key file (optional)

Path to the TLS private key file

□ Enable TLS

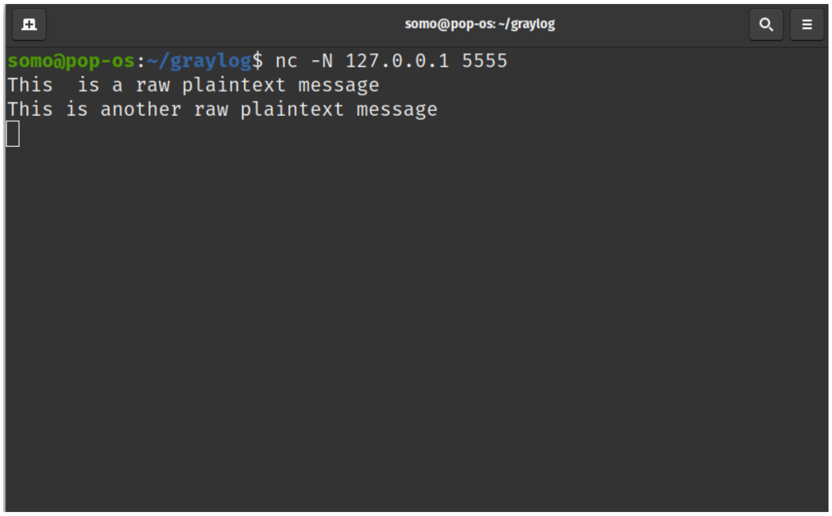
Accept TLS connections

TLS key password (optional)

The password for the encrypted key file.

Figure 6: Setting up input

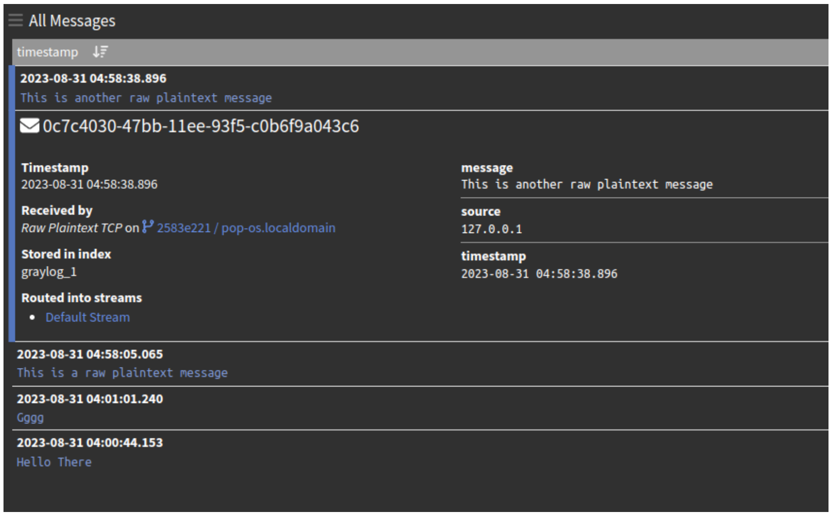
3.2.2 Opening Server



```
somo@pop-os:~/graylog$ nc -N 127.0.0.1 5555
This  is a raw plaintext message
This is another raw plaintext message
```

Figure 7: Server started listening

3.2.3 Message Details



2023-08-31 04:58:38.896
This is another raw plaintext message

✉ 0c7c4030-47bb-11ee-93f5-c0b6f9a043c6

Timestamp
2023-08-31 04:58:38.896

Received by
Raw Plaintext TCP on 2583e221 / pop-os.localdomain

Stored in index
graylog_1

Routed into streams
- Default Stream

message
This is another raw plaintext message

source
127.0.0.1

timestamp
2023-08-31 04:58:38.896

2023-08-31 04:58:05.065
This is a raw plaintext message

2023-08-31 04:01:01.240
Gggg

2023-08-31 04:00:44.153
Hello There

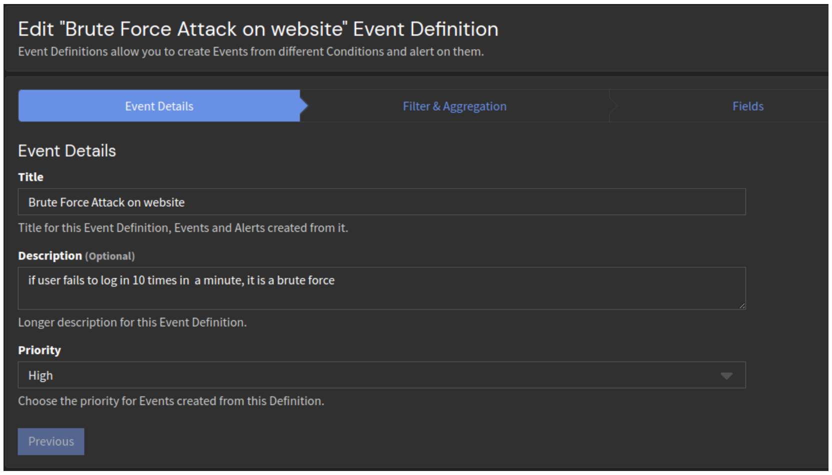
Figure 8: Message Details

3.3 Alert and Events

Events are conditions from log messages that can be used to show alerts and send notification via an email or remote server.

3.3.1 Creating an Event

Here we are creating an event and alert system to detect brute force attacks. If an user fails to log in 10 times under a minute, an event will be created.



Edit "Brute Force Attack on website" Event Definition

Event Definitions allow you to create Events from different Conditions and alert on them.

Event Details

Title

Brute Force Attack on website

Title for this Event Definition, Events and Alerts created from it.

Description (Optional)

if user fails to log in 10 times in a minute, it is a brute force

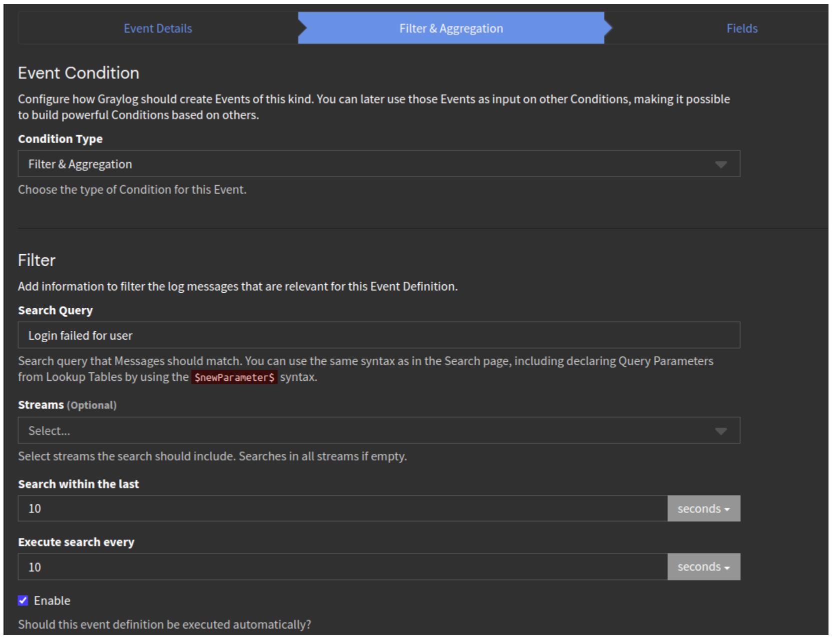
Longer description for this Event Definition.

Priority

High

Choose the priority for Events created from this Definition.

Figure 9: Event Details



Event Condition

Configure how Graylog should create Events of this kind. You can later use those Events as input on other Conditions, making it possible to build powerful Conditions based on others.

Condition Type

Filter & Aggregation

Choose the type of Condition for this Event.

Filter

Add information to filter the log messages that are relevant for this Event Definition.

Search Query

Login failed for user

Search query that Messages should match. You can use the same syntax as in the Search page, including declaring Query Parameters from Lookup Tables by using the `$newParameter$` syntax.

Streams (Optional)

Select...

Select streams the search should include. Searches in all streams if empty.

Search within the last

10 seconds

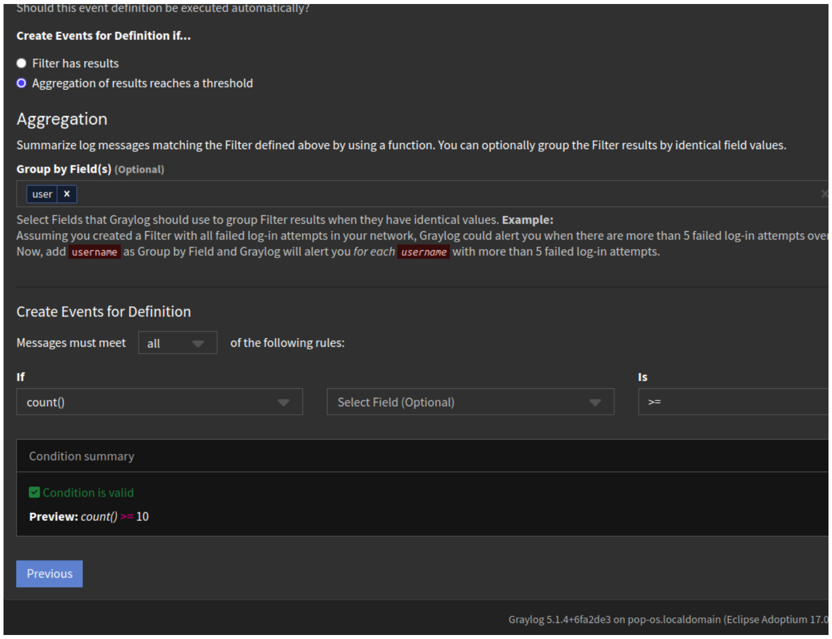
Execute search every

10 seconds

☑ Enable

Should this event definition be executed automatically?

Figure 10: Event Filter



Should this event definition be executed automatically?

Create Events for Definition if...

○ Filter has results

● Aggregation of results reaches a threshold

Aggregation

Summarize log messages matching the Filter defined above by using a function. You can optionally group the Filter results by identical field values.

Group by Field(s) (Optional)

user ×

Select Fields that Graylog should use to group Filter results when they have identical values. **Example:**
Assuming you created a Filter with all failed log-in attempts in your network, Graylog could alert you when there are more than 5 failed log-in attempts ove
Now, add `username` as Group by Field and Graylog will alert you *for each* `username` with more than 5 failed log-in attempts.

Create Events for Definition

Messages must meet all of the following rules:

If count() Select Field (Optional) **Is** >=

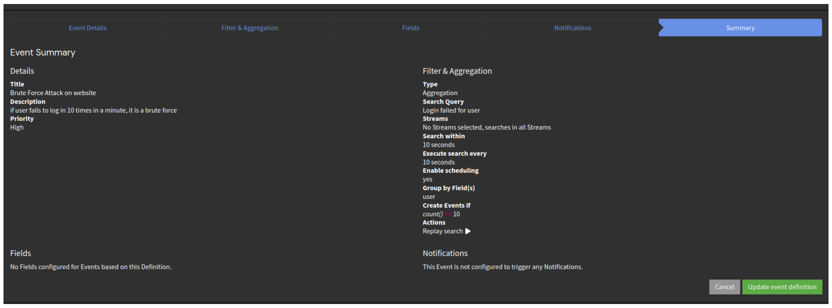
Condition summary

☑ Condition is valid

Preview: *count()* >= 10

Graylog 5.1.4+6fa2de3 on pop-os.localdomain (Eclipse Adoptium 17.0

Figure 11: Event Definition



Event Summary

Details

Title
Brute Force Attack on website
Description
if user fails to log in 10 times in a minute, it is a brute force
Priority
High

Filter & Aggregation

Type
Aggregation
Search Query
Login failed for user
Streams
No Streams selected, searches in all Streams
Search within
10 seconds
Execute search every
10 seconds
Enable scheduling
yes
Group by Field(s)
user
Create Events if
count() >= 10
Actions
Replay search ▶

Fields

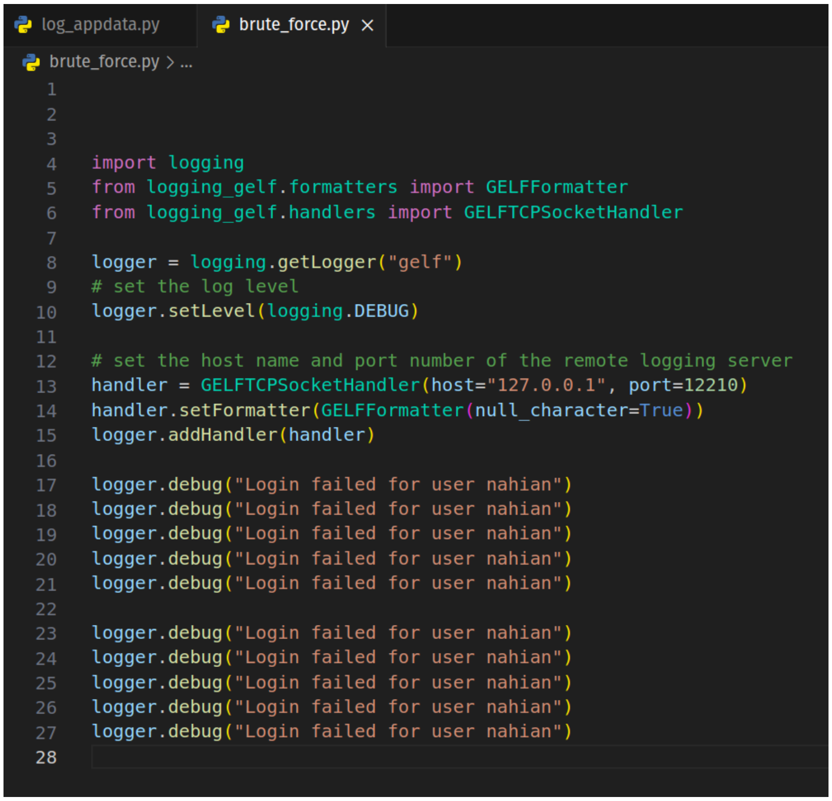
No Fields configured for Events based on this Definition.

Notifications

This Event is not configured to trigger any Notifications.

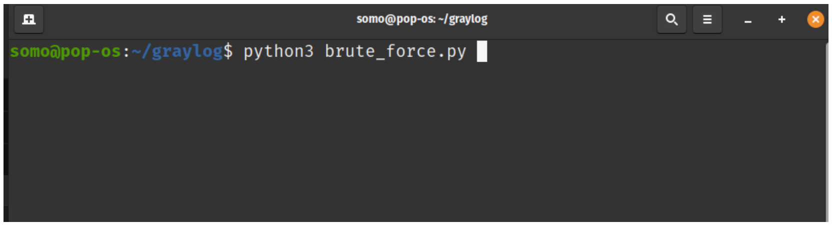
Figure 12: Event Summary

3.3.2 Simulating brute force attack



```python



import logging
from logging_gelf.formatters import GELFFormatter
from logging_gelf.handlers import GELFTCPSocketHandler

logger = logging.getLogger("gelf")
# set the log level
logger.setLevel(logging.DEBUG)

# set the host name and port number of the remote logging server
handler = GELFTCPSocketHandler(host="127.0.0.1", port=12210)
handler.setFormatter(GELFFormatter(null_character=True))
logger.addHandler(handler)

logger.debug("Login failed for user nahian")
logger.debug("Login failed for user nahian")
logger.debug("Login failed for user nahian")
logger.debug("Login failed for user nahian")
logger.debug("Login failed for user nahian")

logger.debug("Login failed for user nahian")
logger.debug("Login failed for user nahian")
logger.debug("Login failed for user nahian")
logger.debug("Login failed for user nahian")
logger.debug("Login failed for user nahian")

```

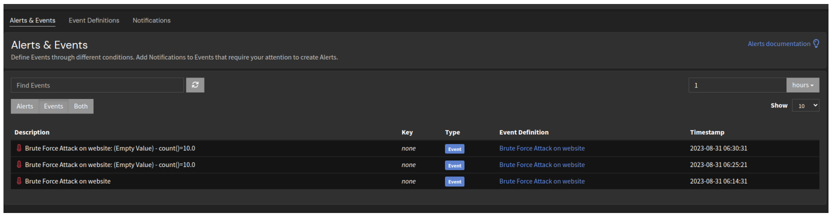
Figure 13: Bruteforce Attack



```
somo@pop-os:~/graylog$ python3 brute_force.py
```

Figure 14: Running Bruteforce Attack

3.3.3 Event Log



Alerts & Events

Define Events through different conditions. Add Notifications to Events that require your attention to create Alerts.

1 hours

Show 10

Description	Key	Type	Event Definition	Timestamp
Brute Force Attack on website: (Empty Value) - count()=10.0	*none*	Event	Brute Force Attack on website	2023-08-31 06:30:31
Brute Force Attack on website: (Empty Value) - count()=10.0	*none*	Event	Brute Force Attack on website	2023-08-31 06:25:21
Brute Force Attack on website	*none*	Event	Brute Force Attack on website	2023-08-31 06:14:31

Figure 15: Alert in Graylog

4 Youtube demonstration

The demonstration link is https://youtu.be/ko7xcU1z02c