

4. IMPLEMENTASI SISTEM

Pada bab ini akan dibahas mengenai implementasi sistem yang telah dirancang sesuai dengan penjelasan yang ada pada bab 3 yaitu analisis dan desain sistem. Sistem akan diimplementasikan dengan menggunakan Grafana, Loki, dan Promtail yang akan di install pada virtual server Ubuntu.

4.1 Instalasi Grafana

Tahapan dalam instalasi Grafana adalah :

1. *Import GPG key*

```
sudo mkdir -p /etc/apt/keyrings/  
  
wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee  
/etc/apt/keyrings/grafana.gpg > /dev/null
```

2. Menambahkan *repository* untuk *stable releases*

```
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable  
main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
```

3. *Update list packages*

```
sudo apt-get update
```

4. *Install Grafana*

```
sudo apt-get install grafana
```

4.2 Instalasi Loki dan Promtail

Setelah melakukan instalasi Grafana, selanjutnya akan dilakukan instalasi Loki yang berguna untuk mengelola data *log*, dan juga melakukan instalasi Promtail yang berguna sebagai agen untuk *push* data *log* dari *server* menuju Loki.

Command untuk instalasi Loki dan Promtail :

```
sudo apt-get install loki promtail
```

4.3 Konfigurasi Rsyslog

Konfigurasi pada Rsyslog dilakukan untuk mengatur nama dan lokasi *file log* yang dikirimkan oleh firewall. Konfigurasi ini bisa ditemukan pada *path /etc/rsyslog.conf*.

Konfigurasi rsyslog.conf :

```
$RepeatedMsgReduction on

$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog

$WorkDirectory /var/spool/rsyslog

$IncludeConfig /etc/rsyslog.d/*.conf

$Template TmplAuth, "/var/log/sangfor/syslog.log"

global(processInternalMessages="on")

module(load="impstats") # config.enabled=`echo $ENABLE_STATISTICS`
module(load="imptcp")
module(load="imudp" TimeRequery="500")
module(load="mmjsonparse")
module(load="mmtf8fix")

input(type="imptcp" port="514")
input(type="imudp" port="514")

*. * action(type="omfwd" protocol="tcp" target="localhost" port="1514"
Template="RSYSLOG_SyslogProtocol23Format" TCP_Framing="octet-counted" KeepAlive="on")
```

4.4 Konfigurasi Promtail

Konfigurasi pada Promtail dilakukan untuk menentukan *URL* dari Loki dan lokasi *log* yang ingin dikirimkan.

Konfigurasi Promtail config.yml :

```
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://192.168.182.4:3100/loki/api/v1/push

scrape_configs:
- job_name: sangfor
  static_configs:
  - targets:
    - localhost
    labels:
      job: sangfor
      __path__: /var/log/sangfor/*.log
```

4.5 Konfigurasi Loki

Konfigurasi pada Loki dilakukan untuk menghubungkan Loki pada Grafana, selain itu juga untuk menentukan *settings* yang ingin dipasang pada Loki yang berguna untuk proses dalam mengelola data *log*.

Konfigurasi Loki config.yml :

```
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096

common:
  instance_addr: 192.168.182.4
  path_prefix: /tmp/loki
  storage:
    filesystem:
      chunks_directory: /tmp/loki/chunks
      rules_directory: /tmp/loki/rules
  replication_factor: 1
  ring:
    kvstore:
      store: inmemory

query_range:
  results_cache:
    cache:
      embedded_cache:
        enabled: true
        max_size_mb: 500
    parallelise_shardable_queries: true

chunk_store_config:
  chunk_cache_config:
    embedded_cache:
      enabled: true
      max_size_mb: 500

query_scheduler:
  max_outstanding_requests_per_tenant: 4096

querier:
  max_concurrent: 4096

limits_config:
  split_queries_by_interval: 24h
  max_query_series: 100000
  max_entries_limit_per_query: 100000

frontend:
  max_outstanding_per_tenant: 4096
  compress_responses: true

schema_config:
  configs:
    - from: 2023-10-01
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

ruler:
  alertmanager_url: http://192.168.182.4:9093
```

4.6 Konfigurasi Logrotate

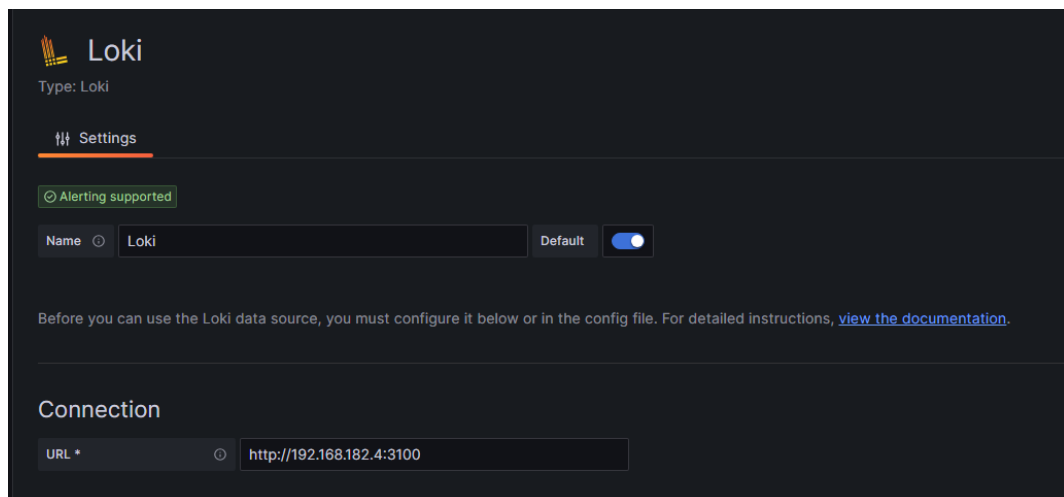
Konfigurasi pada *logrotate* dilakukan agar data *log* tidak menumpuk. Data *log* akan *dirotate* setiap sebulan sekali, data yang lama akan dikompres sehingga tidak memakan banyak memori.

Konfigurasi *logrotate.d/rsyslog* :

```
/var/log/sangfor/syslog.log
{
    monthly
    missingok
    rotate 4
    compress
    delaycompress
    notifempty
}
```

4.7 Penambahan Data Source Loki

Data source dapat ditambahkan melalui *navigation bar* di Grafana. Pada skripsi ini akan digunakan *data source* yaitu Loki untuk mengelola data *log*. Sebelum menambahkan *data source*, pengguna harus mengkonfigurasinya terlebih dahulu. Setelah menambahkan *data source*, pengguna dapat mengaksesnya melalui *explore* pada *naigation bar*.



Gambar 4.1 Data Source Loki



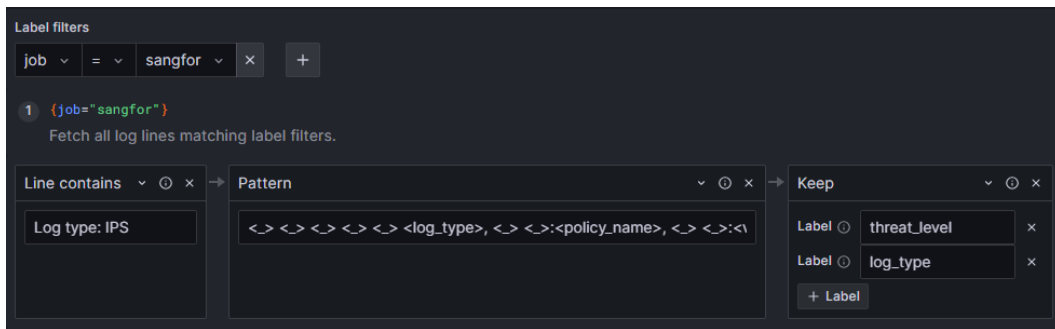
Gambar 4.2 Halaman Explore

4.8 Pembuatan Dashboard Grafana

Pada fitur *dashboard*, akan dilakukannya visualisasi dari hasil *log* yang akan dikelola oleh Loki. Terdapat beberapa visualisasi yang akan dibuat pada *dashboard* ini, yaitu visualisasi berbentuk *donut chart* untuk melihat total keseluruhan ancaman, kemudian terdapat visualisasi yaitu *stat* yang akan digunakan untuk melihat total ancaman keseluruhan, total ancaman tinggi, total ancaman sedang, dan total ancaman rendah dalam bentuk numerik. Ada juga visualisasi *time-series* yang memiliki kegunaan untuk melihat pergerakan total ancaman dari waktu ke waktu. Visualisasi lainnya yang akan dibuat adalah *bar chart*, visualisasi ini akan digunakan untuk memantau kerentanan, ancaman, serta ip teratas. Pada *dashboard* ini juga akan terdapat *logs panel* yang berguna untuk melihat *line log*, dan akan divisualisasikan ke dalam bentuk *table* agar lebih menarik dan lebih mudah untuk dipantau. Visualisasi akan dibuat untuk *log IPS* dan *WAF*, pengguna hanya perlu mengganti *target* dari “**Log type: IPS**” menjadi “**Log type: WAF**” saja.

4.8.1 Pembuatan Visualisasi Donut Chart

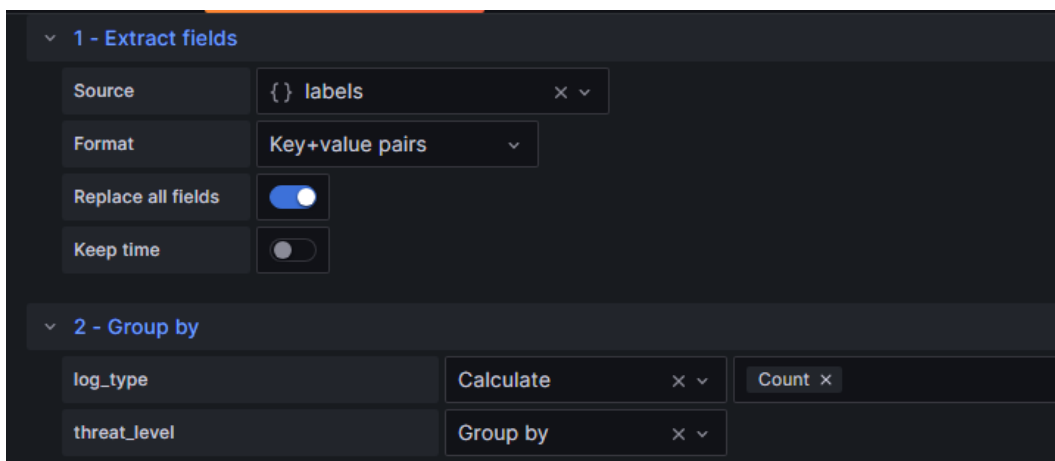
Pada pembuatan visualisasi *donut chart*, pengguna perlu memilih panel *pie chart*, kemudian melakukan pengaturan pada *chart* dengan memilih tipe *donut*. Agar *donut chart* dapat menampilkan hasil total ancaman, *log* harus dikelola terlebih dahulu dengan bantuan Loki.



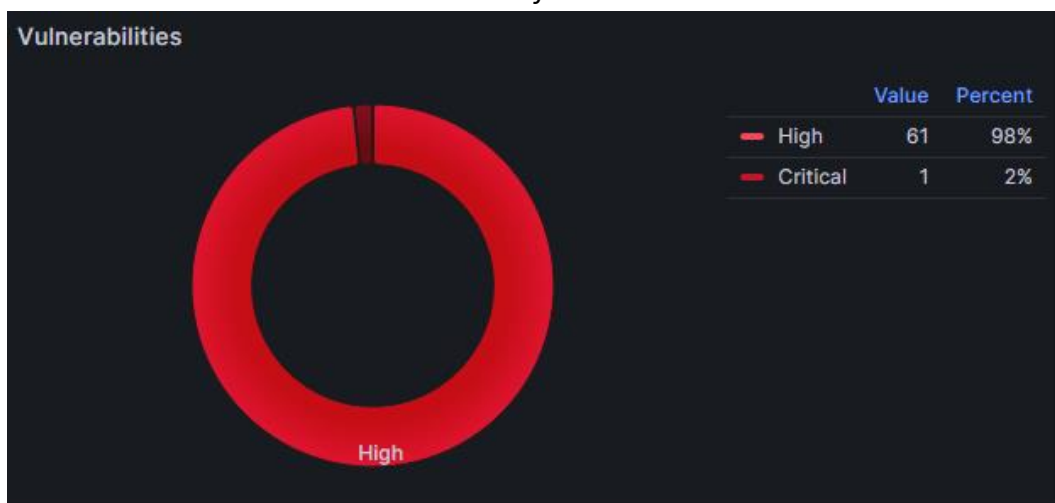
Gambar 4.3 Query Donut Chart

Query code:

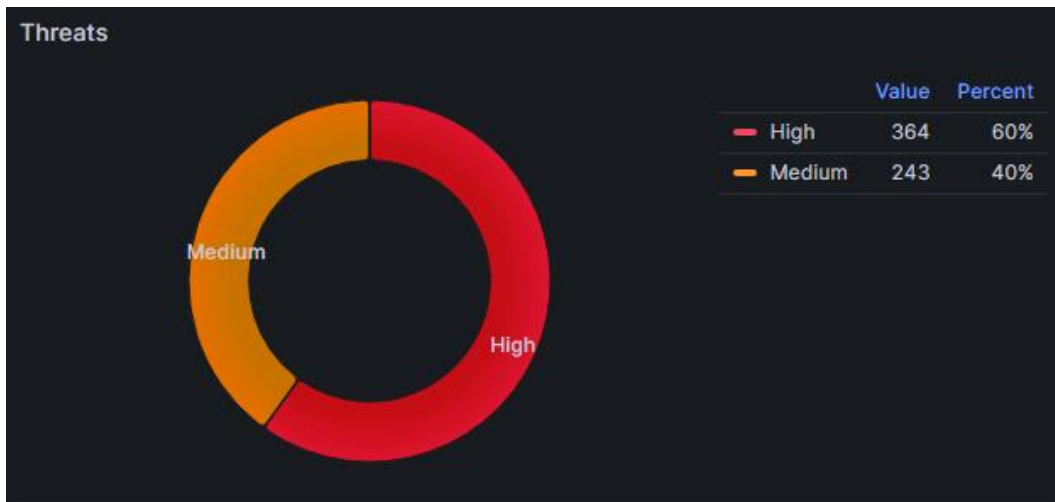
```
{job="sangfor"} |= `Log type: IPS` | pattern `<> <> <> <> <> <log_type>, <> <>:<policy_name>, <> <>:<vulnerability_id>, <> <> <vulnerability_name>, <> <>:<src_ip>, <> <>:<src_port>, <> <>:<dst_ip>, <> <>:<dst_port>, <>:<protocol>, <> <>:<attack_type>, <> <>:<threat_level>, <>:<action>` | keep threat_level, log_type
```



Gambar 4.4 Transform Donut Chart



Gambar 4.5 Donut Chart (IPS)

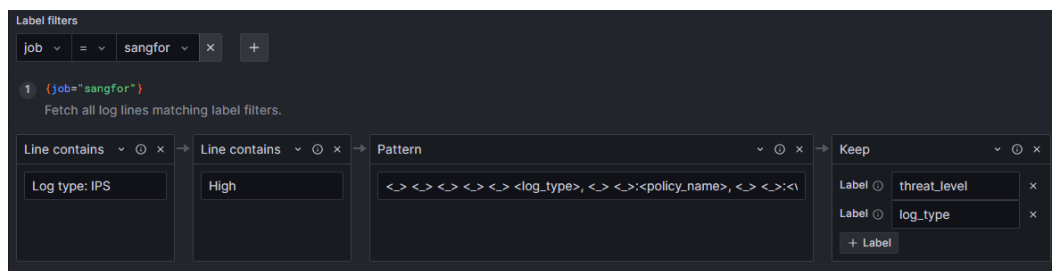


Gambar 4.6 Donut Chart (WAF)

Pada gambar 4.3.1 log dikelola agar dapat menghasilkan *line* yang berisi *log type: ips*. Kemudian setelah itu *log* di *parsing*, dan menyimpan label dengan nama *threat_level* dan *log_type*. Setelah itu pada gambar 4.3.2, hasil *log* yang telah di *parsing* ditransformasikan agar dapat dikelompokkan dan dijumlah.

4.8.2 Pembuatan Visualisasi Stat

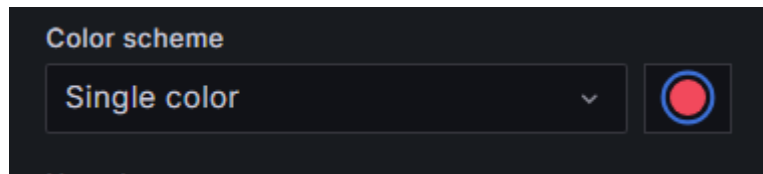
Pembuatan visualisasi ini memiliki cara yang sama dengan *membuat donut chart* dalam mengelola log. Namun, untuk membuat visualisasi ini memiliki perbedaan dalam memilih panel, pengguna harus memilih panel *stat*. Kemudian, mencari *line* yang memiliki *value* tertentu (*High*, *Medium*, *Low*). Selain itu, untuk menentukan warna, pengguna dapat mengaturnya pada *color scheme* yang terdapat di pengaturan panel.



Gambar 4.7 Query Stat

Query code:

```
{job="sangfor"} |= `Log type: IPS` |= `threat level:Critical` | pattern `<_>
<_> <_> <_> <_> <log_type>, <_> <_>:<policy_name>, <_> <_>:<vulnerability_id>,
<_> <_> <vulnerability_name>, <_> <_>:<src_ip>, <_> <_>:<src_port>, <_>
<_>:<dst_ip>, <_> <_>:<dst_port>, <_>:<protocol>, <_> <_>:<attack_type>, <_>
<_>:<threat_level>, <_>:<action>` | keep threat_level,log_type
```



Gambar 4.8 Color Scheme



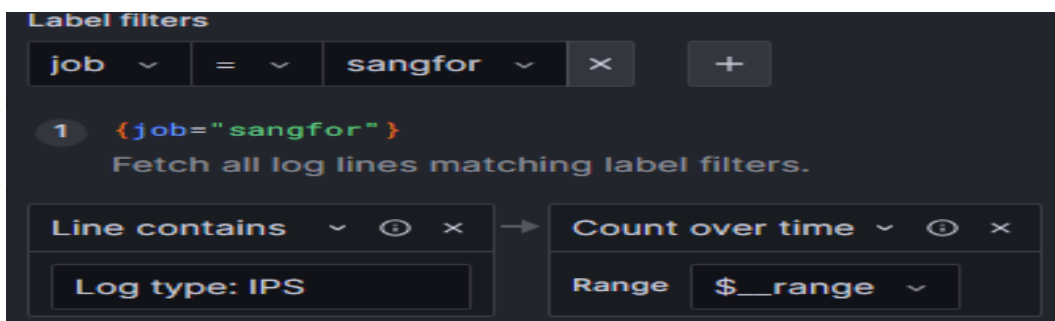
Gambar 4.9 Stat (IPS)



Gambar 4.10 Stat (WAF)

4.8.3 Pembuatan Visualisasi Time Series

Pada pembuatan visualisasi *time series*, log hanya perlu dikelola untuk mencari line dengan tipe *log ips*, kemudian dihitung dalam jangka waktu tertentu.



Gambar 4.11 Query Time Series

```
count_over_time({job="sangfor"} |> `Log type: IPS` [$_range])
```



Pada pembuatan visualisasi *bar chart* dan *table*, pengguna harus melakukan *parsing log* terlebih dahulu agar isi *log* dapat dibaca dengan *detail*. Parsing log dapat dilakukan dengan menggunakan *pattern* (<_>). Untuk menciptakan sebuah *bar chart*, hal yang pertama yang perlu dilakukan adalah memilih panel *bar chart*. Setelah itu melakukan *parsing log* agar dapat melakukan *grouping*. Pada gambar berikut ini menunjukkan *parsing log* dan hanya menyimpan *label_threat_level* dan *vulnerability_name* saja untuk membuat *top vulnerabilities* :



```
{job="sangfor"} |= `Log type: IPS` | pattern `< > < > < > <log_type>,  
< > < >:<policy_name>,< > < >:<vulnerability_id>,< > < >  
<vulnerability_name>,< > < >:<src_ip>,< > < >:<src_port>,< > < >:<dst_ip>,  
< > < >:<dst_port>,< >:<protocol>,< > < >:<attack_type>,< >  
< >:<threat level>,< >:<action>` | keep threat level, vulnerability name
```

1 - Extract fields

Source: { } labels

Format: Key+value pairs

Replace all fields: ☒

Keep time: ☐

Gambar 4.15 Transformation Bar Chart (1)

2 - Group by

threat_level	Calculate	Count
vulnerability_name	Group by	

Gambar 4.16 Transformation Bar Chart (2)

3 - Sort by

Field: threat_level (count)

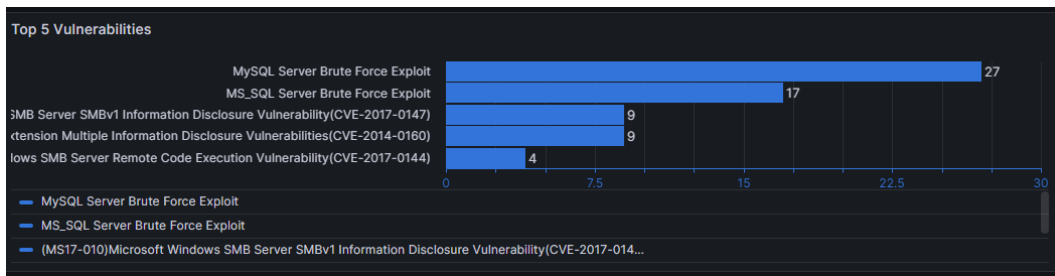
4 - Limit

Limit: 5

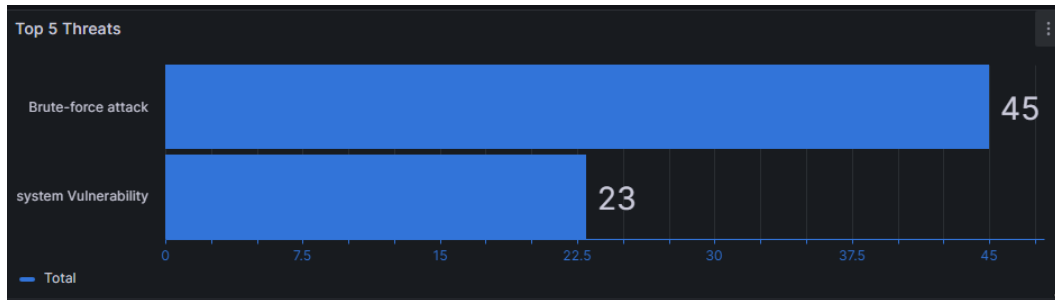
Gambar 4.17 Transformation Bar Chart (3)

Pada *transformation* diatas, *extract fields* berguna untuk memunculkan label – label yang telah berhasil di *parsing*. Kemudian *Group by* berfungsi untuk mengelompokkan isi label. *Sort by* berguna untuk melakukan *ascending* atau *descending*, dan yang terakhir adalah *Limit* yang berguna untuk membatasi hasil yang akan divisualisasikan.

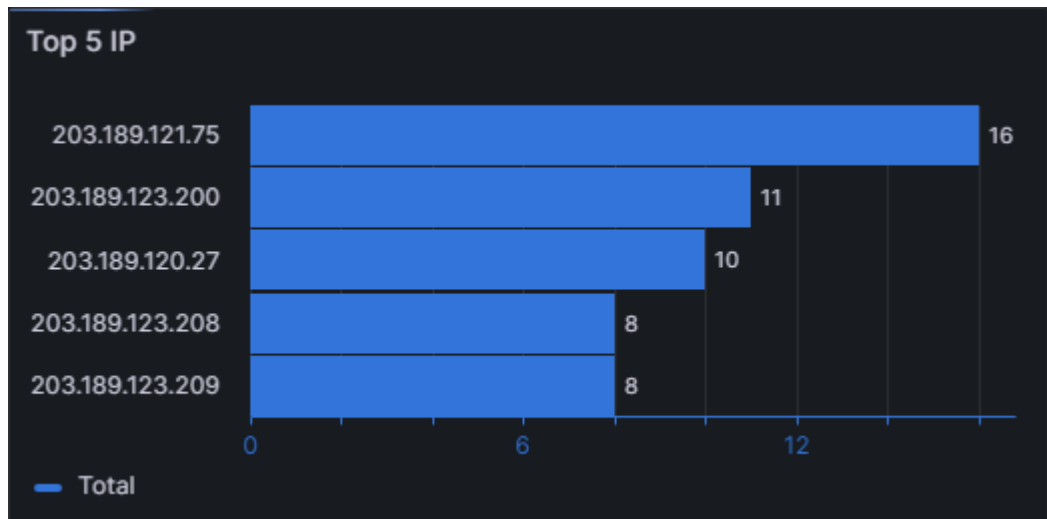
Kemudian berikut ini adalah contoh dari *bar chart* yang sudah dibuat untuk *log type* IPS dan WAF. Melalui visualisasi *bar chart* ini, pengguna dapat melihat *top 5 vulnerabilities*, *threats*, dan juga *ip* pada *log* IPS, sedangkan pada *log* WAF pengguna dapat melihat *top 5 threats* dan juga *top 5 URL*.



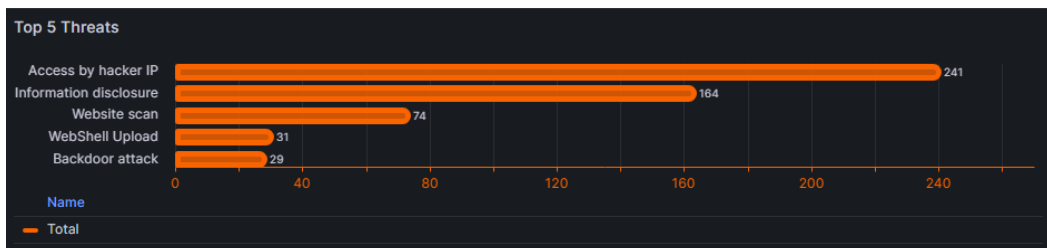
Gambar 4.18 Bar Chart Vulnerabilities (IPS)



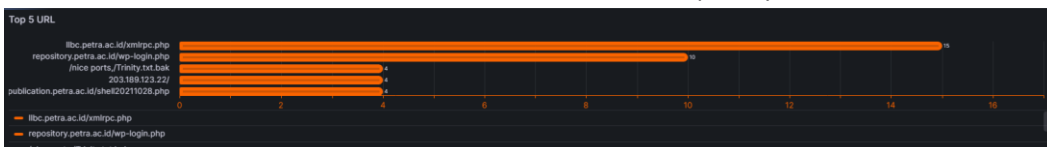
Gambar 4.19 Bar Chart Threats (IPS)



Gambar 4.20 Bar Chart IP (IPS)



Gambar 4.21 Bar Chart Threats (WAF)



Gambar 4.22 Bar Chart URL (WAF)

4.8.5 Pembuatan Visualisasi Table

Sama seperti pada pembuatan panel *bar chart*, pada panel ini perlu untuk melakukan *parsing* terlebih dahulu, kemudian hasilnya di *extract* agar dapat digunakan. Hal pertama yang perlu dilakukan adalah memilih *panel table*, kemudian melakukan *parsing log*, setelah itu melakukan *transformation extract fields* dan *sort time* agar hasil *table* dimulai dari yang terkini terlebih dahulu.

[illegible]

Gambar 4.23 Query Parsing Log Table

Query code:

```
[{job="sangfor"} |= `Log type: IPS` | pattern `<_> <_> <_> <_> <_> <log_type>, <_> <_>:<policy_name>, <_> <_>:<vulnerability_id>, <_> <_> <vulnerability_name>, <_> <_>:<src_ip>, <_> <_>:<src_port>, <_> <_>:<dst_ip>, <_> <_>:<dst_port>, <_>:<protocol>, <_> <_>:<attack_type>, <_> <_>:<threat_level>, <_>:<action>` | label_format Source_IP=src_ip | label_format Source_Port=src_port | label_format Dest_IP=dst_ip | label_format Dest_Port=dst_port | label_format Threat_Level=threat_level | label_format Attack_Type=attack_type | label_format Vuln_id=vulnerability_id | label_format Vuln_Name=vulnerability_name | drop filename, job, log_type, protocol, policy_name
```

1 - Extract fields

Source

{ } labels

×

▼

Format

Key+value pairs

▼

Replace all fields

☒

Keep time

☒

2 - Format time

Time Field

Time

3 - Sort by

Field

Time

Gambar 4.24 Transformation Table

Attack_Type	Dest_IP	Dest_Port	Source_IP	Source_Port	Threat_Level	Vuln_Name
Brute-force attack	203.189.123.208	22	103.10.227.157	42732	High	SSH Server Brute Force Exploit
Brute-force attack	203.189.123.203	22	59.56.73.141	40212	High	SSH Server Brute Force Exploit
Brute-force attack	203.189.121.75	3389	223.247.159.199	57357	High	RDP Server Brute Force Exploit
Brute-force attack	203.189.121.75	3389	36.133.110.87	59747	High	RDP Server Brute Force Exploit

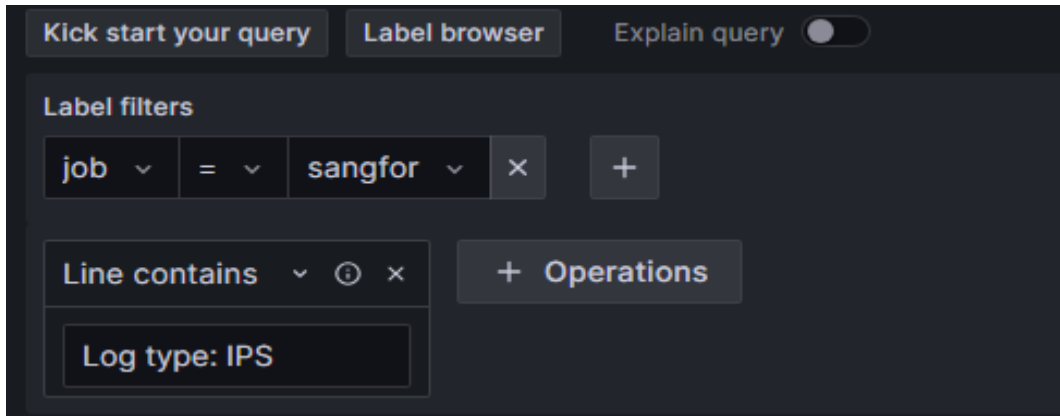
Gambar 4.25 Table (IPS)

Attack_type ▾	Dest_IP ▾	Dest_Port ▾	Source_IP ▾	Source_Port ▾	Threat_Level ▾	URL ▾
Brute-force login to website	203.189.123.200	443	172.104.61.57	60572	High	telematics.petra.ac.id/wp-login.php
Code injection	203.189.120.27	80	20.38.39.187	50109	High	genta.petra.ac.id/wp-admin/css/colors/blue/bl
Information disclosure	203.189.120.27	80	139.180.137.234	64539	Medium	www.ksislife.petra.ac.id/wp-content/
Information disclosure	203.189.120.27	80	139.180.137.234	64839	High	www.ksislife.petra.ac.id/wp-admin/

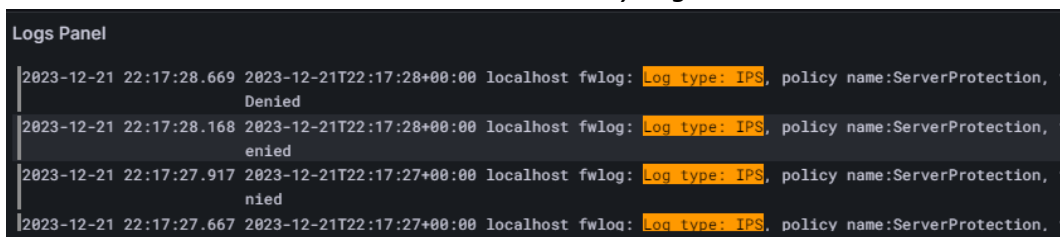
Gambar 4.26 Table (WAF)

4.8.6 Pembuatan Visualisasi Logs Panel

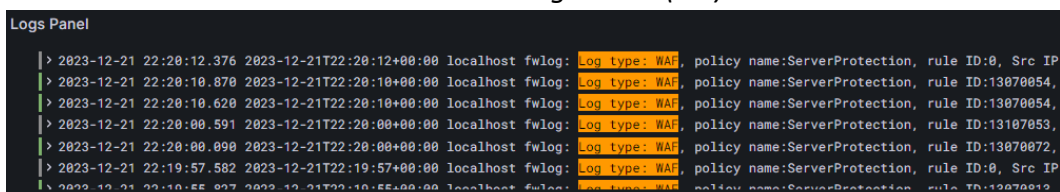
Pembuatan *logs panel* sangat sederhana, hanya perlu melakukan *query* biasa untuk mendapatkan hasil *log*. Panel ini digunakan untuk melihat *logs* yang ada secara mentah (*raw*).



Gambar 4.27 Query Log



Gambar 4.28 Logs Panel (IPS)



Gambar 4.29 Logs Panel (WAF)

4.9 Konfigurasi Grafana.ini dan Defaults.ini

Konfigurasi pada *Grafana.ini* perlu dilakukan terlebih dahulu sebelum membuat *Alert*. Konfigurasi pada *Grafana.ini* dilakukan agar *Grafana* dapat mengirimkan email dengan bantuan *gmail* dari *admin* serta dapat mengakses *view alerting* secara publik. Path dari *Grafana.ini* dan *Defaults.ini* sebagai berikut :

Path *Grafana.ini* : `/etc/grafana/grafana.ini`

Path *Defaults.ini* : `/usr/share/grafana/conf/defaults.ini`

Konfigurasi SMTP pada Grafana.ini :

```
##### SMTP / Emailing #####
[smtp]
enabled = true
host = smtp.gmail.com:587
user = c14190023@john.petra.ac.id
password = 
cert_file = 
key_file = 
skip_verify = true
from_address = c14190023@john.petra.ac.id
from_name = Grafana
ehlo_identity =
```

Gambar 4.30 SMTP Configuration

```
##### Server #####
[server]
# Protocol (http, https, h2, socket)
protocol = http

# Minimum TLS version allowed. By default, this value is empty. Accepted values: 1.0, 1.1, 1.2, 1.3
min_tls_version = ""

# The ip address to bind to, empty will bind to all interfaces
http_addr =

# The http port to use
http_port = 3000

# The public facing domain name used to access grafana from a browser
domain = Grafana

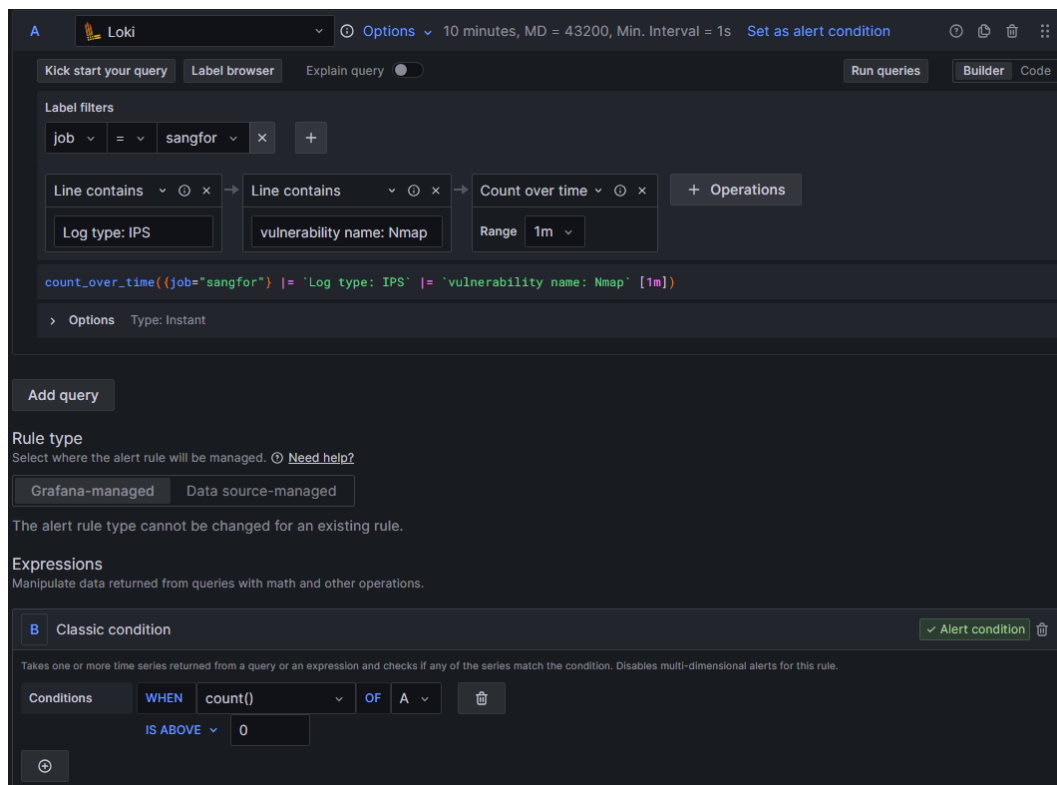
# Redirect to correct domain if host header does not match domain
# Prevents DNS rebinding attacks
enforce_domain = false

# The full public facing url
root_url = http://192.168.182.4:3000
```

Gambar 4.31 Server Configuration Grafana.ini and Defaults.ini

4.10 Pembuatan Alert

Pada skripsi ini melakukan uji coba pembuatan *alerting* untuk memunculkan notifikasi peringatan jika ada *query* yang ke *trigger*. Ada beberapa serangan pada log IPS seperti *RDP Server Brute Force Exploit*, *SMB Server Brute Force Exploit*, *Scan Tool Friendly-Scanner Detection*, dan lainnya. Kemudian untuk log WAF ada beberapa serangan seperti *Information Disclosure*, *OS command Injection*, *Path Traversal*, dan lainnya. Namun pada percobaan ini, *query* yang dicoba adalah mencari log dengan line yang memiliki *vulnerability name: Nmap*, *Log type: IPS* dan juga mencari log dengan line yang memiliki *attack type : OS command injection*, *Log type: WAF*. Tujuan pemilihan log tersebut adalah untuk mengirimkan notifikasi peringatan jika log IPS mendeteksi adanya *Nmap scanning* dan log WAF mendeteksi adanya *OS Command Injection* pada jaringan UK Petra. Notifikasi akan dikirimkan melalui email ketika *alert state firing*.



Gambar 4.32 Alert Rule Grafana

Update contact point

Name *

PTIK

Integration

Email

Addresses

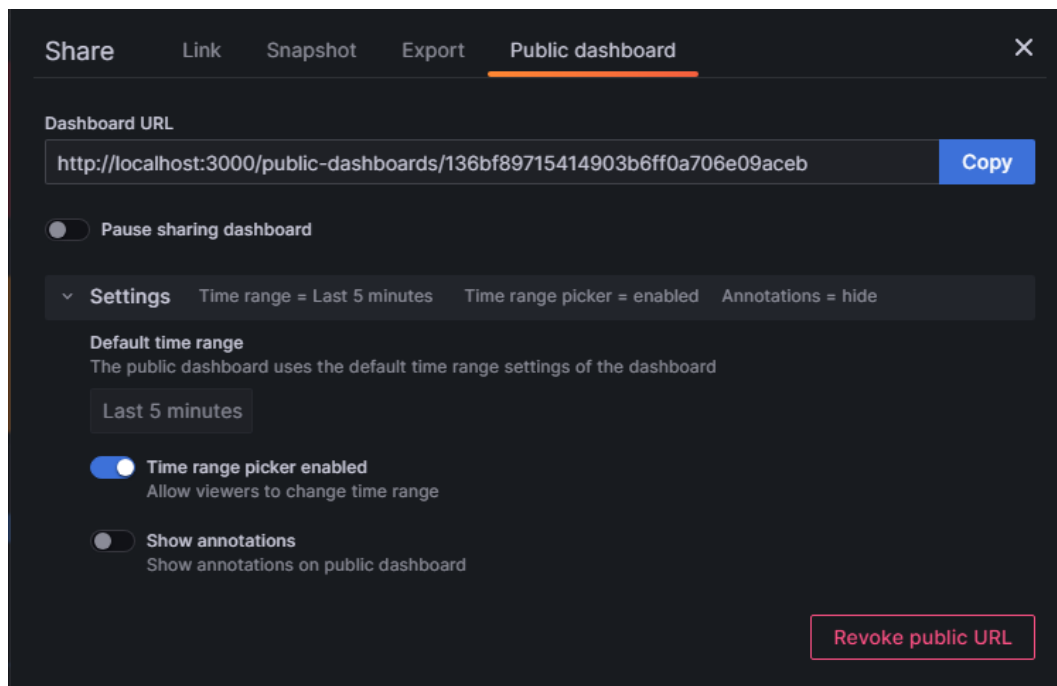
You can enter multiple email addresses using a ";", "\n" or "," separator

c14190023@john.petra.ac.id

Gambar 4.33 Contact Point Alert

4.11 Pembuatan Display Dashboard

Display dashboard untuk publik dapat dibuat ketika visualisasi pada *dashboard* sudah dibuat. Setelah membuat semua visualisasi yang ada, selanjutnya pengguna hanya perlu mengklik tombol *share*, kemudian *public dashboard* dapat dibuat. Pengguna dapat mengakses *public dashboard* dengan cara mengganti "*localhost*" pada *link* yang telah muncul dengan *ip address* yang digunakan.



Gambar 4.34 Pembuatan Display Dashboard