



Quick Guides

Turning Event Logs into Modern Day Alerting

May 2016

1 INTRODUCTION

Our staff at Kustodian have been working at Security Operation Centers (SOC) throughout their careers, with the traditional large screens across the walls, and staff responding and preventing incidents for our customers. When it was time to build our SOC someone said why don't you do it how you want to do it not follow everyone else? So we built a SIEM/SOC solution around what we wanted, what suited us. I think all security professionals working the hours we do, would want this, and that is flexibility. We wanted alerts to come to us via email, SMS, SNS or Slack instead of being stuck in front of a screen for 8-12 hour shifts. The beauty of this is we could spend our time doing research, putting in kit into racks or going to the movies. If you are like us we do 12-18hr days, whether it be official work or researching, tinkering or studying. This is the main reason we integrated flexibility into the SIEM.

Whether you use the features or not, they are there for you and for some of our clients this is ideal. Some of our clients work 9-5pm at charities or not for profits. After 5pm they have no visibility until the morning if something is going on, and the same with weekends, holidays and when they are on training. At least this way when they can be watching Game of Thrones and see the alerts and decide what course of action they want to take, then not seeing the alert at all and coming into work on Monday and have a Sony situation.

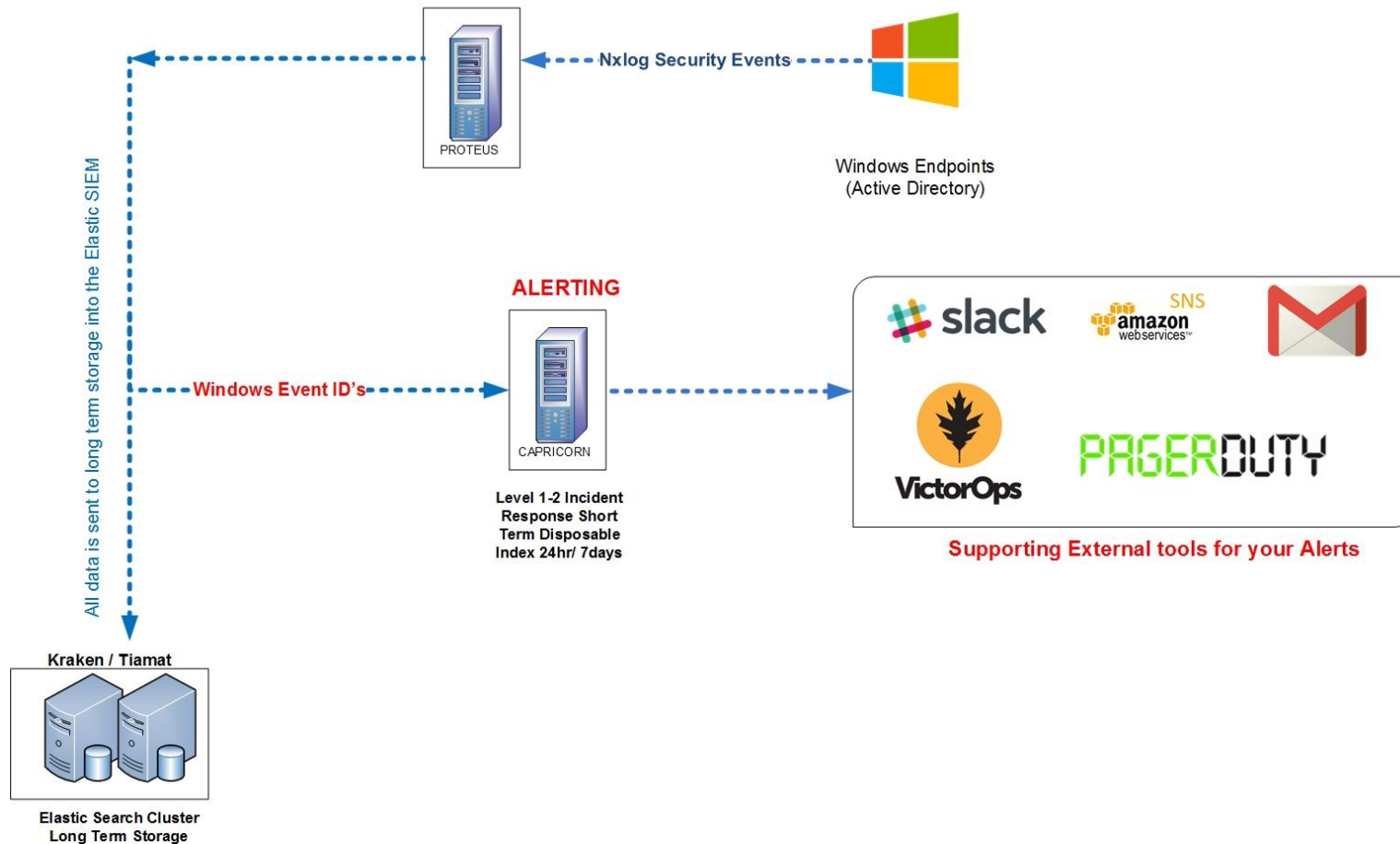
1.1 WHAT ABOUT SECURITY

Of course the traditionalists will say where is the security in that, that's a risk having alerts for clients leaving the SOC. We agree, the beauty of SIEMonster incorporating the power of Graylog Elastic etc, is we can choose what we want to go into those alerts to email or mobile devices. Think of it this way, an alert comes in, and its approximately 30 data fields of information, the client, IP address, user ID, access rights. Now think of an Excel Spreadsheet with these 30 rows of information. You can choose what you want to go into that alert and obfuscate others. For example, for a CISO, you could have Name of Client obfuscated and time of event and what happened. For an outsourced provider you could alert them to the fact that Client A has an issue, which maps to the real client's name that they do not know, they can log a ticket and get it to Level 2 staff who are sitting at the SOC who know the clients details. The choice is yours on what data, can go in an email, Slack alert.

1.2 THIS GUIDE

This guide is intended for those who have minimal experience with SIEMonster, Slack or Graylog, and want alerts to go to their email/slack, as quickly as possible. For those that need more help there are videos at www.siemonster.com or email us, we can help you out.

Overview of Event Log Flow to Graylog



2 INSTALL GUIDE FOR WINDOWS EVENTS

Once SIEMonster has been successfully configured to receive Windows Security Event logs using NXLog, the next stage is to set up a suitable alerting mechanism ie Slack or email.

On the Proteus appliance we can configure Logstash to *fork* event log data to Graylog on Capricorn for the purpose of alerting and pattern analysis. If you have forgotten Proteus sends all the data to Kraken and Tiamat the Elastic Search Databases which is used for traditional lookups ie, "In the last 90 days, who sent who used that workstation id to do this function. It also forks the data you want to Capricorn so you can control your alerting, ie let me know when someone adds themselves to the Domain Admin Group, when someone emails our competitor or when someone is trying to log onto a router/firewall.

This process is defined in the outputs section of the Logstash configuration files, which in this case is the 99-outputs.conf file located at /etc/logstash/conf.d on Proteus. More details can be found in the Siemonster build guide.

Here we can be selective about which events will be sent, or simply send through all events. To help things along the Logstash Windows Event filter already tags failed login attempts for NTLM and Kerberos non-machine accounts. This tag, 'logon_failure', can then be used in the outputs section if required. If you don't want to capture all events but you want to capture say 8 specific events we use the tagging feature in the 10-file specific to windows events to not clutter up 99 file.

Edit the file /etc/logstash/conf.d/99-outputs.conf and find the following section:

In this case Capricorn has the IP 192.168.137.104, adjust for your environment.

```
# Uncomment the follow lines to fork data to Graylog
#   if "logon_failure" in [tags] {
#       gelf {
#           host => "192.168.137.104"
#           port => 12202
#       }
#   }
```

By uncommenting the lines as indicated and restarting Logstash, selective data will be sent to Graylog on Capricorn.

EventID 4771

EventID 4625

Non-machine accounts.

The following example shows how to send only events indicating that an account has been locked out:

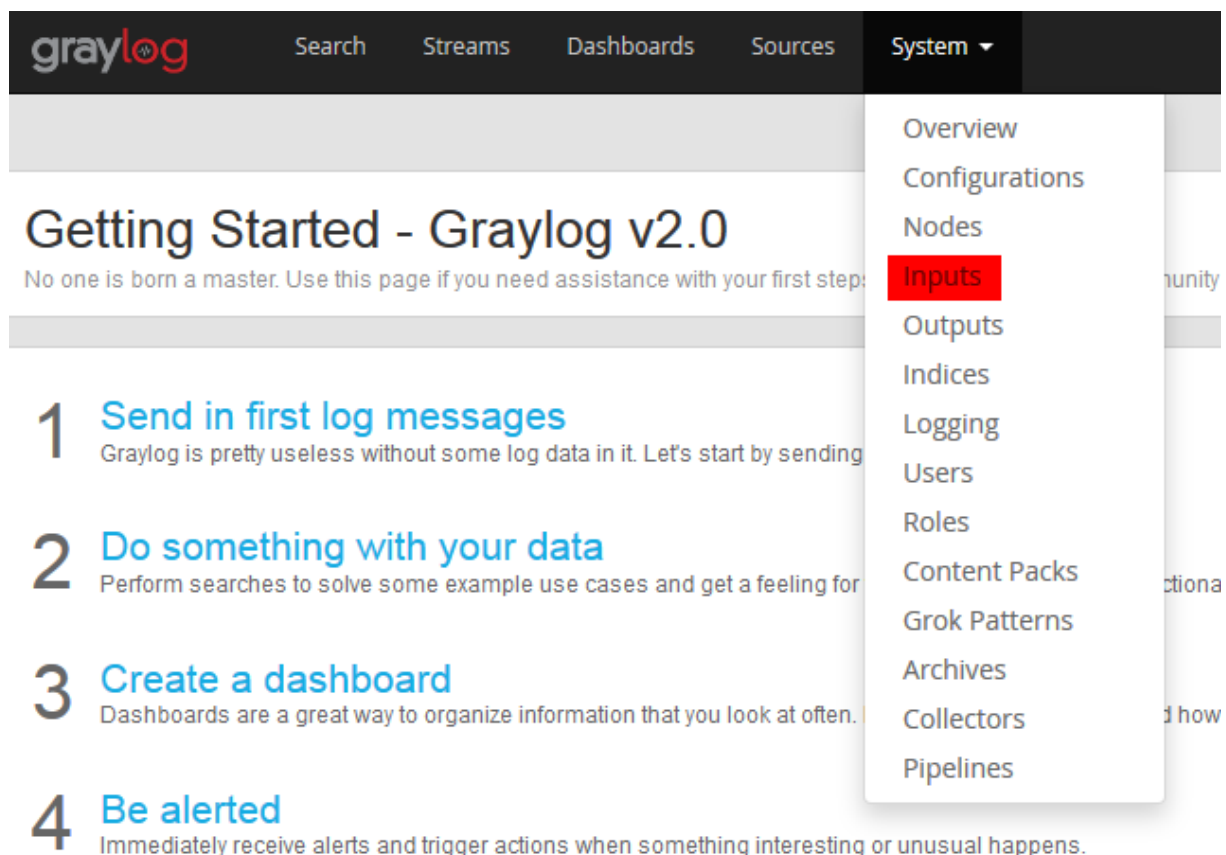
```
# Uncomment the follow lines to fork data to Graylog
if [EventID] == 4740 {
  gelf {
    host => "192.168.137.104"
    port => 12202
  }
}
```

Send all event data to Graylog as follows if you want all data to go nito Graylog for alerts, Just hash out the if statement above gelf and remove a } bracket like below.

```
# Uncomment the follow lines to fork data to Graylog
gelf {
  host => "192.168.137.104"
  port => 12202
}
```

Now that we have event log data being sent to Graylog, a suitable input needs to be configured.

Open up the Graylog web interface – System – Inputs



The screenshot shows the Graylog web interface. The top navigation bar includes links for Search, Streams, Dashboards, Sources, and System. The 'System' menu is open, showing a list of options: Overview, Configurations, Nodes, Inputs (highlighted in red), Outputs, Indices, Logging, Users, Roles, Content Packs, Grok Patterns, Archives, Collectors, and Pipelines. The main content area displays 'Getting Started - Graylog v2.0' with a list of four steps: 1. Send in first log messages, 2. Do something with your data, 3. Create a dashboard, and 4. Be alerted.

Create a new Gelf UDP input and click on Launch new input

GELF UDP

Launch new input

Find more inputs

Give it a title and select the Capricorn node:

Title

Windows Security Events

Select a name of your new input that describes it.

☐ Global

Should this input start on all nodes

Node

c4d55199 / capricorn

On which node should this input start

Change the port to 12202 and Save.

Port

12202

Port to listen on.

Generate some suitable events on the Windows server/workstation such as logging on with the wrong password. Check and refresh the Windows Security Event viewer and ensure that the event has been registered, i.e. Event ID 4625 or 4771.

On Graylog – Inputs – Windows Security Events click on ‘Show received messages’

Show received messages

Manage extractors

Stop input

More actions

Throughput / Metrics

1 minute average rate: 0 msg/s

Network IO: 0B 0B (total: 0B 0B)

Empty messages discarded: 0

The following illustrates an example of successful receipt of messages:

Messages

[Previous](#)
[1](#)
[2](#)
[3](#)
[4](#)

Timestamp ↑

2016-05-08 10:13:24.210

An account failed to log on.

Subject:

Security ID: S-1-5-18

2016-05-08 10:13:23.005

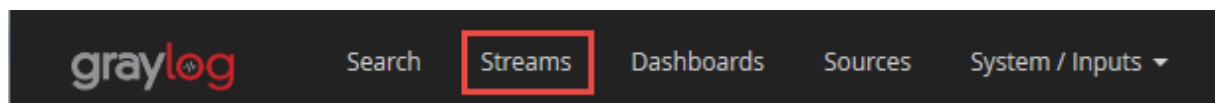
An account failed to log on.

Subject:

Security ID: S-1-5-18

Once this has been accomplished we can set up a 'Stream' based on this input to enable to enable the alerting function.

On the main Graylog menu choose 'Streams'



Create Stream

Create a Stream and provide a title/description

Creating Stream

×

Title

Windows Kerberos Login Failures

Description

4771 Events

Cancel

Save

Find the Stream and click 'Edit Rules'

Edit rules

Select the Input you created:

Select an Input from the list below and click "Load Message" to load the most recent message from this input.

Windows Security Events (org.graylog2.inputs.gelf.udp)

Load Message

Add Stream rule EventID – match exactly – 4771 and save.

Add stream rule

Then click 'I'm done'

New Stream Rule

Field

EventID

Type

match exactly

Value

4771

☐ Inverted

Result: Field *EventID* must match exactly 4771

I'm done!

Back to Streams on the main menu, find the stream you created and click 'Start stream'

Edit rules

Manage Outputs

Manage Alerts

Start stream

More actions ▼

Edit rules

Manage Outputs

Manage Alerts

Pause stream

More actions ▼

Next click on 'Manage Alerts'.

Add new alert condition

Configure conditions that will trigger stream alerts when they are fulfilled.

Message count condition ▼

Trigger alert when there are ☒ more ☐ less

than message in the last minute and

then wait at least minute until triggering a new alert. (grace period)

When sending an alert, include the last message of the stream evaluated for this alert condition.

Create new alert condition

For initial testing set the options to more than 1 message in the last minute. Once testing is completed set these options to suit your environment, e.g. set to more than 10 messages in last minute to indicate a brute force attack.

Click on 'Create new alert condition'

Next, create a Slack callback:

Callbacks

The following callbacks will be performed when this stream triggers an alert.

Slack alarm callback ▼

Add callback

[Find more callbacks](#)

No configured alarm callbacks.

Click on 'Add callback'

You will need the Webhook from your Slack Team account.

This can be found under Browse Apps – Custom Integrations – Incoming WebHooks – Configurations on *YourSlackTeam*

Post to Channel

Start by choosing a channel where your Incoming Webhook will post messages to.

siemonster_alerts ▼

[or create a new channel](#)

Add Incoming WebHooks integration

Create new Slack alarm callback

Webhook URL

Slack "Incoming Webhook" URL

Channel

Name of Slack #channel or @user for a direct message.

Also, add a user from your Slack team and save:

User name (optional)

User name of the sender in Slack

Add a Receiver for notification of alerts via email:

If the receiver has a Graylog account with an email address registered, then just enter the username and click 'Subscribe'

Otherwise an email address can be entered as shown:

Receivers

The following Graylog users will be notified about alerts via email if they

✉ jim@siemonster.com ✖

Username:

Subscribe

Email address:

Subscribe

Send a dummy alert:

Send test alert

Check your Slack account and Email client.



Kraken BOT 4:27 PM ☆

Alert for Graylog stream *Windows Kerberos Logon Failures:*

Dummy alert to test notifications

Stream configuration is now complete.

To check for triggered alerts head back to Streams – Windows Kerberos Login Failures – and click on Manage alerts

Scroll down to 'Triggered alerts' and monitor alerts sent:

Triggered alerts 31 alerts total

Triggered	Condition	Reason
a few seconds ago	21440325-88e6-4dbb-9878-0069ccf5c0d0	Stream had 30 messages in the last 1 minutes
3 minutes ago	21440325-88e6-4dbb-9878-0069ccf5c0d0	Stream had 31 messages in the last 1 minutes



Repeat the setup for NTLM failures matching EventID to 4625



To combine both events 4771 & 4625 into one alert, simply modify the Stream rule as follows:

2. Manage stream rules

Please load a message to check if it would match against these rules and therefore be routed into this stream.

- ☐ A message must match all of the following rules
- ☒ A message must match at least one of the following rules

  EventID must match exactly 4771

  EventID must match exactly 4625

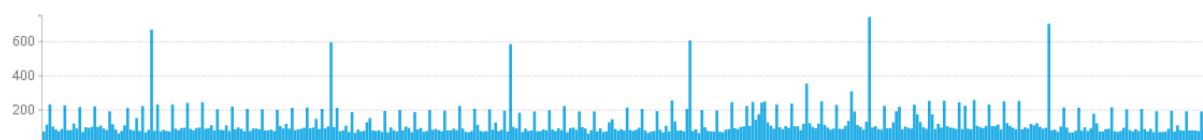
I'm done!

For historical and pattern analysis, head over to System – Inputs – Windows Security Events – Show Received Messages:

 Update every 30 seconds ▼

Histogram

Year, Quarter, Month, Week, Day, Hour, Minute



For quick visualization of top offending users, expand TargetUserName in the Fields display to the left and click 'Quick Values' for a pie chart and list of top failures.:

▼ ☐ TargetUserName

[Generate chart](#)

[Quick values](#)

[Statistics](#)

[World Map](#)

3 INSTALL GUIDE FOR OSSEC (HIDS) ALERTING

This guide is for users who want to extend their SIEM's and alert for OSSEC events on both Windows and Linux boxes.

Note: Build versions

Amazon AWS Build - Users just need to unhash the entries below

VM Build – Post 1st June Entries are already unhashed.

VM Build – Prior to 1st June grab the latest 99-outputs.conf file from here

<https://raw.githubusercontent.com/siemonster/logstash/master/99-outputs.conf> and lines are already unhashed and place them in the /etc/logstash/conf.d/

On Proteus:

Edit the file:

/etc/logstash/conf.d/99-outputs.conf

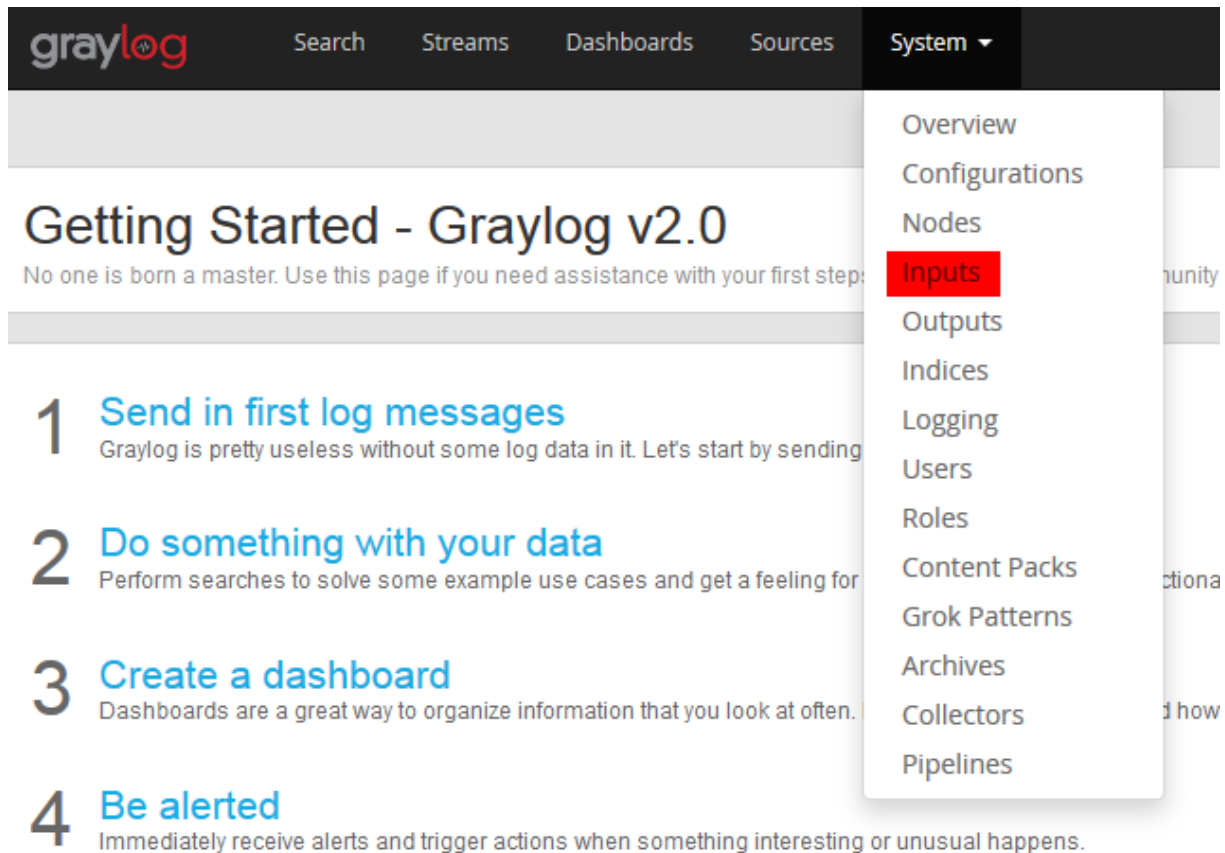
Remove comments from this block, change IP to Capricorn IP

```
output {
  if "ossec" in [tags] {
    elasticsearch {
      hosts => ["localhost:9200"]
      index => "ossec-%{+YYYY.MM.dd}"
      document_type => "ossec"
      template => "/etc/logstash/elastic-ossec-template.json"
      template_name => "ossec"
      template_overwrite => true
    }
  }
  # Uncomment the follow lines to fork OSSEC data to Graylog
  # gelf {
  #   host => "192.168.137.104"
  #   port => 12204
  #   short_message => "full_log"
  # }
}
```

- sudo service logstash restart

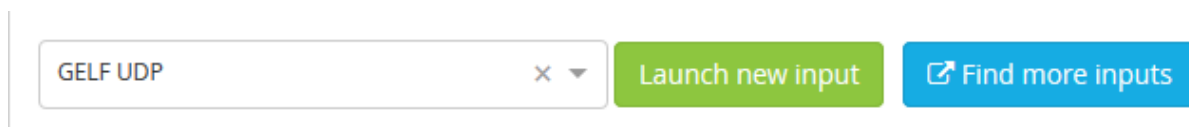
On Capricorn graylog box:
Create a new Gelf UDP input on port 12204

Open up the Graylog web interface – System – Inputs



The screenshot shows the Graylog v2.0 'Getting Started' page. The 'System' menu is open, and 'Inputs' is highlighted in red. The page content includes a list of four steps: 1. Send in first log messages, 2. Do something with your data, 3. Create a dashboard, and 4. Be alerted.

Create a new Gelf UDP input and click on Launch new input



The screenshot shows the 'Launch new input' form. A dropdown menu is open, showing 'GELF UDP' as the selected option. The 'Launch new input' button is highlighted in green.

Give it a new name select Capricorn or AWS instance of Capricorn and click on save

Inputs

Graylog nodes accept data via inputs. Launch or terminate as many as you need.

GELF UDP
Launch

Global inputs 0 configured

There are no global inputs.

Local inputs 2 configured

appliance-gelf-udp GELF UDP RUNNING

On node ★ cbf8aa39 / ip-172-31-23-164.us-west-2.compute.internal

```
allow_override_date: true
bind_address: 0.0.0.0
override_source: <empty>
port: 12201
recv_buffer_size: 1048576
```

Static fields

from_gelf: true ✖

appliance-syslog-udp Syslog UDP RUNNING

On node ★ cbf8aa39 / ip-172-31-23-164.us-west-2.compute.internal

```
allow_override_date: true
bind_address: 0.0.0.0
expand_structured_data: false
```

Launch new *GELF UDP* input

Title

Select a name of your new input that describes it.

☒ **Global**

Should this input start on all nodes

Node

ct
i39 / ip-172-
64.us-west-2.compute.internal

On which node should this input start

Bind address

Address to listen on. For example 0.0.0.0 or 127.0.0.1.

Port

Port to listen on.

Receive Buffer Size (optional)

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

Override source (optional)

The source is a hostname derived from the received packet by default. Set this if you want to override it with a custom string.

Cancel
Save

Start it up if it hasn't already

Local inputs 3 configured

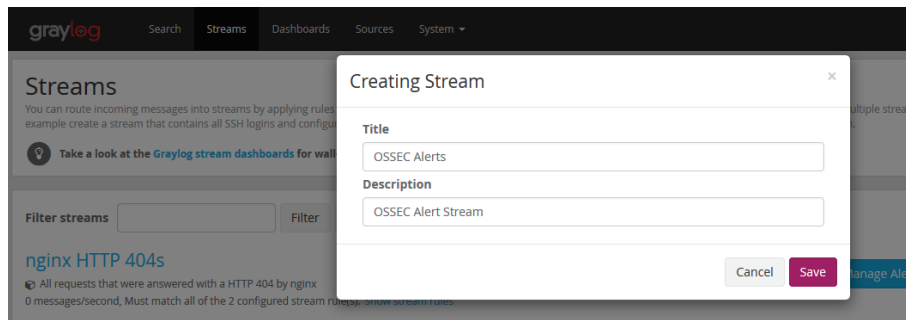
OSSEC Alerts GELF UDP RUNNING

On node ★ cbl 54.us-west-2.compute.internal

```
bind_address: 0.0.0.0
override_source: <empty>
port: 12204
recv_buffer_size: 262144
```

Run putty without any keys or anything and try to logon to Capricorn external IP as an invalid user.

Set up a stream, under Stream, Create Stream and start it up.



Select Edit rules under OSSEC alerts. Load Message and you should see your failed user after clicking on Load Message like below

1. Load a message to test rules

Recent Manual

Select an Input from the list below and click "Load Message" to load the most recent message from this input.

OSSEC Alerts (org.graylog2.inputs.gelf.udp.GELFUDPin) Load Message

2. Manage stream rules

Please load a message to check if it would match against these rules and therefore be routed into this stream.

- ☒ A message must match all of the following rules
- ☐ A message must match at least one of the following rules

No rules defined.

I'm done!

192.168.15.104/streams/574bc03833e2fd039a913408/edit

graylog
Search
Streams
Dashboards
Sources
System
2

Rules of Stream »OSSEC Alerts»

This screen is dedicated to an easy and comfortable creation and manipulation of stream rules. You can see the effect configured stream rules have on message matching here.

1. Load a message to test rules

Recent Manual

Select an Input from the list below and click "Load Message" to load the most recent message from this input.

All OSSEC events (org.graylog2.inputs.gelf.udp.GELFUC) Load Message

✉ 257be610-261e-11e6-99d1-005056901d15

Timestamp

2016-05-30 14:22:33.000

Stored in index

graylog_0

AgentName

capricorn

beat_hostname

capricorn

beat_name

capricorn

count

1

decoder_name

sshd

decoder_parent

sshd

facility

gelf-rb

full_log

May 30 14:22:31 capricorn sshd[18973]: Failed password for invalid user bad from 192.168.15.231 port 59211 sshd

Create a rule based on rule Alert Level by clicking on Add Stream rule in Section 1 before Manage Stream Rules.

New Stream Rule ✕

Field

Type

greater than
▼

Value

☐ Inverted

Result: Field *rule_AlertLevel* must be greater than 4

The server will try to convert to strings or numbers based on the matcher type as good as it can.

[Take a look at the matcher code on GitHub](#)

Regular expressions use Java syntax. 💡

Cancel

Save

Add alert condition under Manage Alerts next to OSSEC Alerts

Configured alert conditions

Message count condition

Alert is triggered when there are more than 0 messages in the last minute. Grace period: 1 minute. Including last message in alert notification.

Trigger alert when there are ☒ more ☐ less

than messages in the last minute and

then wait at least minutes until triggering a new alert. (grace period)

When sending an alert, include the last messages of the stream evaluated for this alert condition.

Edit condition

Delete condition

Save

For testing purposes initially set the trigger to more than 0 (zero) messages in a minute, with a wait (grace) period of 1 minute, and include the last 1 message in the alert.

Click 'Save' when done.

Typically, in a production scenario you would be looking at alerting on OSSEC rules at a level higher than 9. To understand the process and mechanics of alerting we have used this example so that an end user can quickly grasp the concepts and receive Slack and Email alerts with minimal configuration as a proof of concept.

Next create a Slack Callback.

Callbacks

The following callbacks will be performed when this stream triggers an alert.

Slack alarm callback

Add callback

Find more callbacks

No configured alarm callbacks.

If you do not have a Slack Callback option available simply run the following commands on the Capricorn server.

```
sudo wget -O /opt/graylog/plugin/graylog-plugin-slack-2.1.0.jar
https://github.com/Graylog2/graylog-plugin-slack/releases/download/2.1.0/graylog-plugin-slack-2.1.0.jar
```

```
sudo graylog-ctl restart
```

Wait a minute or 2 for Graylog to restart.

Add your Webhook URL

This can be found under Browse Apps – Custom Integrations – Incoming WebHooks – Configurations on YourSlackTeam

Post to Channel

Start by choosing a channel where your Incoming Webhook will post messages to.

siemonster_alerts

or [create a new channel](#)

Add Incoming WebHooks integration

Create new Slack alarm callback

Webhook URL

https://hooks.slack.com/services/YOURWEBHOOK

Slack "Incoming Webhook" URL

Channel

#alert_channel

Name of Slack #channel or @user for a direct message.

Also, add a user from your Slack team and save:

User name (optional)

User name of the sender in Slack

Click on Save

Emoji to use as the icon for this message (overrides icon URL)

☐ Short mode (optional)

Enable short mode? This strips down the Slack message to the bare minimum to take less space in the chat room. Not used in alarm callback but only in the message output module.

Cancel

Save

In order to create an additional Email Callback (Alerts will be sent to both Slack and Email), ensure that your email details have been setup in the Graylog configuration file.

Configure email for Graylog

Edit this file: `/opt/graylog/conf/graylog.conf`

Find this section and put your details in.

```
# Email transport
transport_email_enabled = true
transport_email_hostname = smtp.gmail.com
transport_email_port = 587
transport_email_use_auth = true
transport_email_use_tls = true
transport_email_use_ssl = false
transport_email_auth_username = jim@siemonster.com
transport_email_auth_password = '
transport_email_subject_prefix = [siemonster]
transport_email_from_email = alerts@siemonster.com
```

Find this section and use external IP or DNS name of Capricorn.

```
#
transport_email_web_interface_url = http://alerts.siemonster.com
```

Then `sudo graylog-ctl restart`

Next create an Email Callback

Callbacks

The following callbacks will be performed when this stream triggers an alert.

Email Alert Callback ▼

Add callback

Find more callbacks

Change the sender to your email address or an email address within your domain.
 Modify the Email Subject to preference,
 Click on Save.

Create new Email Alert Callback ×

Sender

The sender of sent out mail alerts

E-Mail Subject

The subject of sent out mail alerts

E-Mail Body (optional)

```
#####
Alert Description: ${check_result.resultDescription}
Date: ${check_result.triggeredAt}
Stream ID: ${stream.id}
Stream title: ${stream.title}
Stream description: ${stream.description}
${if stream_url}Stream URL: ${stream_url}${end}

Triggered condition: ${check_result.triggeredCondition}
#####
```

The template to generate the body from

Click on 'Send test alert' to check Slack and Email configuration is correct. Before you do that add your email address you configured earlier into the email address subscribe below.

any other email address to the alert receivers if it has no Graylog user associated.

Email address:

To verify, try to logon via SSH as an invalid user to the Capricorn machine.

Check the Triggered alerts log at the bottom of the Alert Configuration page.

Triggered alerts 15 alerts total

Show: 10 ▼

Triggered	Condition	Reason
4 hours ago	4735b41f-d7c5-4f70-837c-4cd4d95b68af	Stream had 2 messages in the last 1 minutes with trigger condition more than 0 messages. (Current grace time: 1 minutes)

[Show callbacks](#)