



SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF CYBERSECURITY

Assignment 3

ACME-02

PRACTICAL NETWORK DEFENSE

Professor:

Angelo Spognardi

Students:

Federica Bianchi, 1891952

Giovanna Camporeale, 2086227

Francesco Rizzo, 1955615

Alessio Vigilante, 1839055

Contents

1	Brainstorming	2
2	Implementation	2
2.1	Import the relevant logs	2
2.2	Create alerts	3
2.3	Create at least one helpful dashboard	5
3	Test of the configuration	6
4	Final remarks	7

1 Brainstorming

First of all, we did a recap of the previous assignment and familiarized with the Graylog web interface, accessible at the address <http://100.100.1.10:9000>.

We watched the video provided to learn how to configure the Graylog and then decided which hosts should have sent the logs. In particular, based on the policies of the second assignment we decided that only the hosts of the DMZ and the DNS server should use the log systems, since the Clients and the External services network are not allowed to access the Internal servers network.

2 Implementation

Let's see how we configured the Graylog from the web interface.

2.1 Import the relevant logs

1. After deciding which hosts should use the log service, we needed to enable each specific host (namely webserver, proxyserver and dnsserver) to redirected all the logs to port 1514 of the Graylog server.

We did this by creating a file in `/etc/rsyslog.d/graylog.conf` with the following content:

```
GNU nano 5.4 /etc/rsyslog.d/graylog.conf *
$PreserveFQDN on
*,* @100.100.1.10:1514:RSYSLOG_SyslogProtocol23Format
```

Figure 1: File `/etc/rsyslog.d/graylog.conf`

2. We created an input for the system. In particular, we decided to create just one input that regards all the ACME services. To create an input, we went on the web interface of the Graylog, to the section *System* and selected *Inputs*. Then, we have chosen as input the Syslog UDP and the following settings:

ACME services (syslog) Syslog UDP RUNNING

On node ★ 2aae584a / graylog

[Show received messages](#) [Manage extractors](#) [Stop input](#) [More actions](#)

```
allow_override_date: true
bind_address: 0.0.0.0
expand_structured_data: false
force_rdns: false
number_worker_threads: 2
override_source: <empty>
port: 1514
recv_buffer_size: 262144
store_full_message: true
```

Throughput / Metrics
1 minute average rate: 0 msg/s
Network IO: 0B (total: 3.4KiB)
Empty messages discarded: 0

Figure 2: ACME services input

2.2 Create alerts

1. In order to create an alert in the Graylog, we need to create the corresponding event. This events contains conditions that, when are satisfied, trigger an alert. For this assignment, first of all we created the events of the failed login for the webserver, proxyserver and dnsserver.

We went to the section *Alerts & Events*, then clicked on *Event Definitions*. We decided to generate an event whenever a user fails a login more than 3 times. In order to define the correct query to capture this event, we tried to fail some logins to inspect the log message sent to the Graylog. This is an example of that:

```
2023-05-16 10:29:53.000 +00:00
webserver.acme-02.test login[7197]: FAILED LOGIN (4) on '/dev/tty1' FOR 'UNKNOWN', Authentication failure
```

Figure 3: Log for a failed login on the webserver.

We also noticed that the numbers of the failed logins are reset after a specific interval of time.

So we defined the following query:

Filter

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

Search Query

'FAILED LOGIN \{4\}' AND source:dnsserver.acme-02.test

Figure 4: Example of Search Query to catch the event "Failed Login" in the dnsserver.

After that, we also created a notification, in particular an email notification, to associate to the alert.

In addition to this three alerts, we also decided to handle the event of an ssh access with an invalid username. So we tried to access with a not existing username "Luisa" and looked at the logs in order to define the correct Search Query for the event. We obtained the following logs:

```
2023-05-16 13:26:42.000 +00:00
proxyserver.acme-02.test sshd[2092]: Failed none for invalid user luisa from 100.100.253.6 port 61449 ssh2
2023-05-16 13:26:41.000 +00:00
proxyserver.acme-02.test sshd[2092]: Invalid user luisa from 100.100.253.6 port 61449
```

Figure 5: Example of Search Query to catch the event "Failed Login" in the dnsserver.

Then, we created the event with the following Search Query:

Filter

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

Search Query

message:ssh AND "invalid user"

Figure 6: Example of Search Query to catch the event "Failed Login" in the dnsserver.

So, at the end, we had the following events:

The screenshot displays the 'Event Definitions' page in Graylog. At the top, there are tabs for 'Alerts & Events', 'Event Definitions', and 'Notifications'. Below the tabs, a message states: 'Create new Event Definitions that will allow you to search for different Conditions and alert on them.' A link to 'Graylog's new Alerting system let you define more flexible and powerful rules. Learn more in the documentation' is provided. The main area contains a search bar with 'Find Event Definitions', 'Find', 'Reset', and 'Create Event Definition' buttons. A 'Show' dropdown is set to '10'. Four event definitions are listed:

- DNS Failed Login** (Filter & Aggregation): Triggers an alert for the DNS server whenever the login is failed more than 3 times. Runs every 1 minutes, searching within the last 1 minute. Triggers 1 Notification. [Show details](#)
- Proxyserver Failed Login** (Filter & Aggregation): Triggers an alert for the Proxy server whenever the login is failed more than 3 times. Runs every 1 minutes, searching within the last 1 minute. Triggers 1 Notification. [Show details](#)
- SSH Invalid Login** (Filter & Aggregation): Shows invalid logins to the hosts. Runs every 1 minutes, searching within the last 1 minute. Does **not** trigger any Notifications. [Show details](#)
- Webserver Failed Login** (Filter & Aggregation): Triggers an alert for the Web server whenever the login is failed more than 3 times. Runs every 1 minutes, searching within the last 1 minute. Triggers 1 Notification. [Show details](#)

Each event definition has 'Edit', 'Share', and 'More' buttons.

Figure 7: Events overview

2.3 Create at least one helpful dashboard

We decided to create three different dashboards: one for the failed login (to the webserver, proxyserver and dnsserver), one for the successful ones and one for the ssh login with an invalid user.

To create the dashboards, we went to **Systems/input** and then to all the logs received. Here we defined the search query that we wanted, in order to select the logs for the dashboard, and then clicked on *export to dashboard*.

To show them, we decided to select a pie chart.

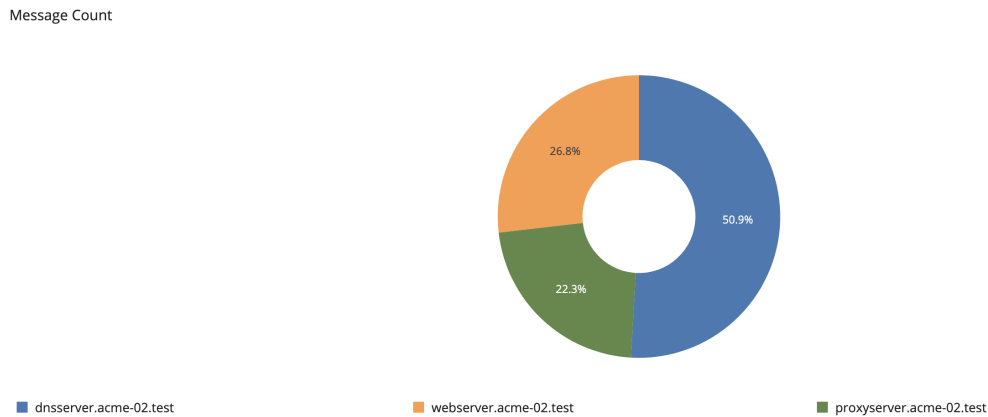


Figure 8: Dashboard for the Failed Logins.

The others are showed in the Section "Test of the configuration".

3 Test of the configuration

1. In order to test if the redirection of the logs worked, we simply checked that the logs show up in the *Inputs* section after selecting *Show received messages*.

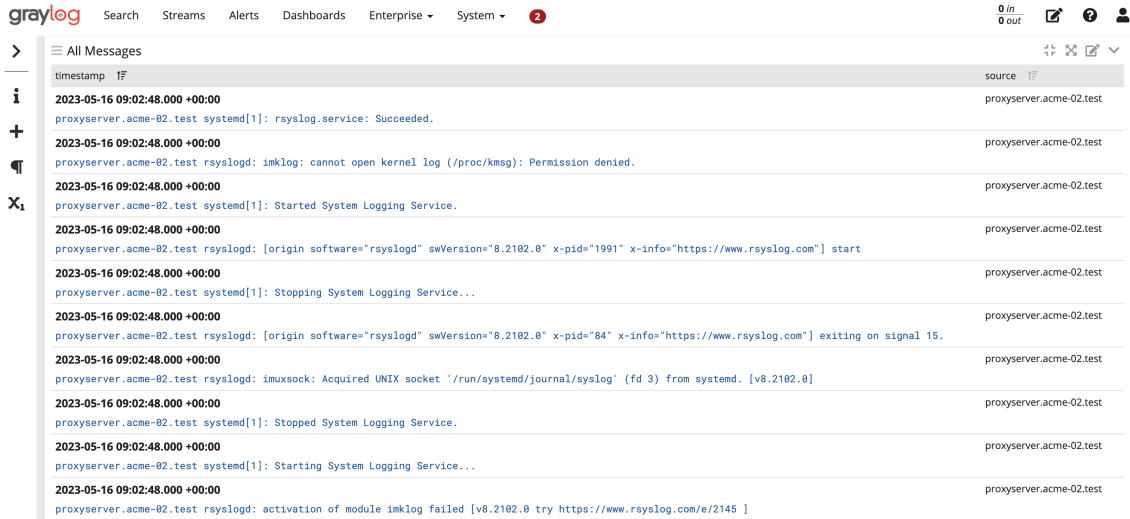


Figure 9: Logs of DMZ and DNS server

2. In order to test the right configuration of alarms, we simulated a scenario in which there were three failed logins for each host, and then we verified if the corresponding notification appeared in the *Alerts&Events* section.

SSH Invalid Login	none	Alert	SSH Invalid Login	2023-05-17 10:11:43
Proxyserver Failed Login	none	Alert	Proxyserver Failed Login	2023-05-16 10:56:50
Webserver Failed Login	none	Alert	Webserver Failed Login	2023-05-16 10:29:53
Proxyserver Failed Login	none	Alert	Proxyserver Failed Login	2023-05-16 10:26:50
DNS Failed Login	none	Alert	DNS Failed Login	2023-05-16 10:10:29

Figure 10: Alert testing

3. The test of creation of a dashboard is visible in the section *Create at least one helpful dashboard*, in particular in the **Figure 8**. We checked this also for the other alerts:

Message Count

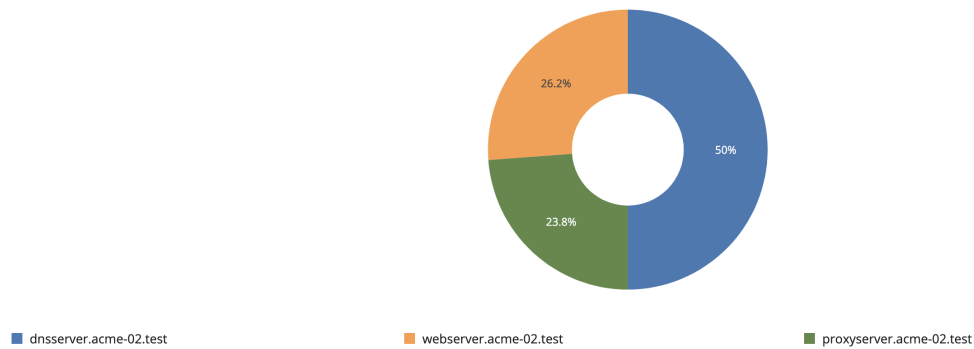


Figure 11: Dashboard of the successful logins.

Message Count

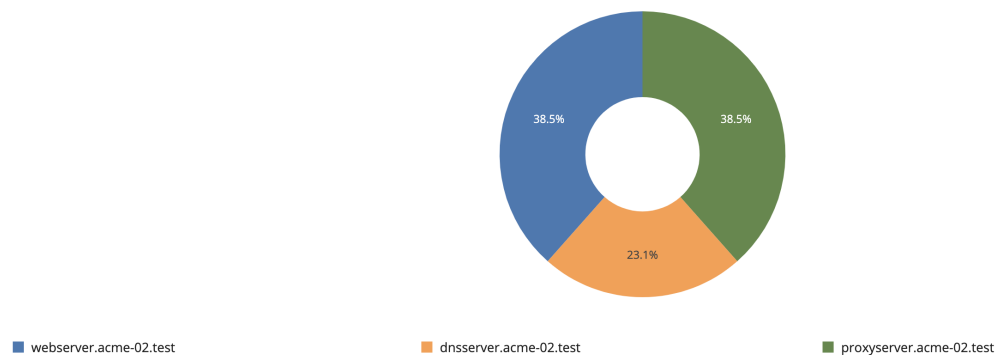


Figure 12: Dashboard for the ssh logins with invalid users.

4 Final remarks

This assignment allowed us to go into detail about the log services, in particular about the Graylog.

The implementation was pretty straightforward after we watched the guide. The only difficult we encountered was the creation of the notification associated to a specific event.