

Linux - practicum week 3

Dennis van der Laan

Zorg ervoor dat alle voortgang wordt bijgehouden in de Git repository.

Opdracht 1: RegEx

In de file apache-access-log.txt vind je een access-log.

Schrijf een regex die mailadressen herkent van het bedrijf "shaw". Het bedrijf staat zowel onder de tld com en .net geregistreerd.

In de basis volgt het bedrijf Shaw de conventies van email-naamgeving van RFC 822.

Daarboven op hebben ze de volgende naamgevingssystemetiek.

- Generieke opbouw van een mailadres bij Shaw is: <mailnaam>@shaw.com of <mailnaam>@shaw.net
- Voor de <mailnaam> geldt:
 - o Minimaal 2 karakters (karakter= cijfer, letter en speciale tekens), maximaal 99
 - o Op laatste positie staat of een letter ("a" t/m "z" , "A" t/m "Z"), of een cijfer("0" t/m "9")
 - o Op andere posities geen cijfers toegestaan, wel letters of de speciale tekens ".", "_", "-"
 - o Andere speciale tekens geheel niet toegestaan

De regex kun je als volgt controleren:

```
grep -E -o "<YOUR_REGEX>" access_log
```

"^[a-zA-Z0-9._-]{2,99}@shaw\.(com|net)\$"

Opdracht 2: monitoring en logging

a) Zet een Linux (Ubuntu)server op die als monitor/log server kan dienen.

Als monitor tool kun je de volgende tools gebruiken:

Munin

Nagios

Prometheus

(andere mogen ook na overleg met docent).

Voor het opzetten van een centrale syslog server:

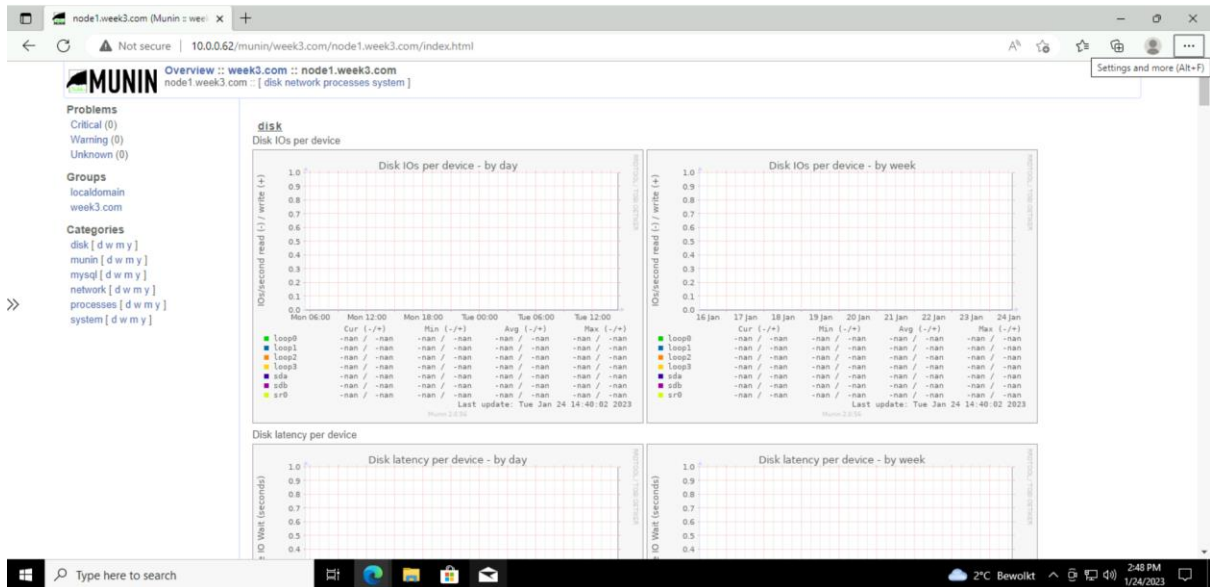
Syslog

Syslog NG

Elastic Stack

(andere mogen ook na overleg met docent).

b) Zet zelf een Ubuntu Linux server op die door de vorige server (in opdracht a) gemonitord wordt.



```

GNU nano 4.8
#
#max_processes 16

# RRD updates are per default
# To reduce IO and enable the
# the location of the socket
#
#rrdcached_socket /var/run/rrd

# Drop someuser@fnord.comm a
# something changes (OK -> WA
#contact.someuser.command mai
#contact.anotheruser.command
#
# For those with Nagios, the
# the services must be define
#contact.nagios.command /usr/

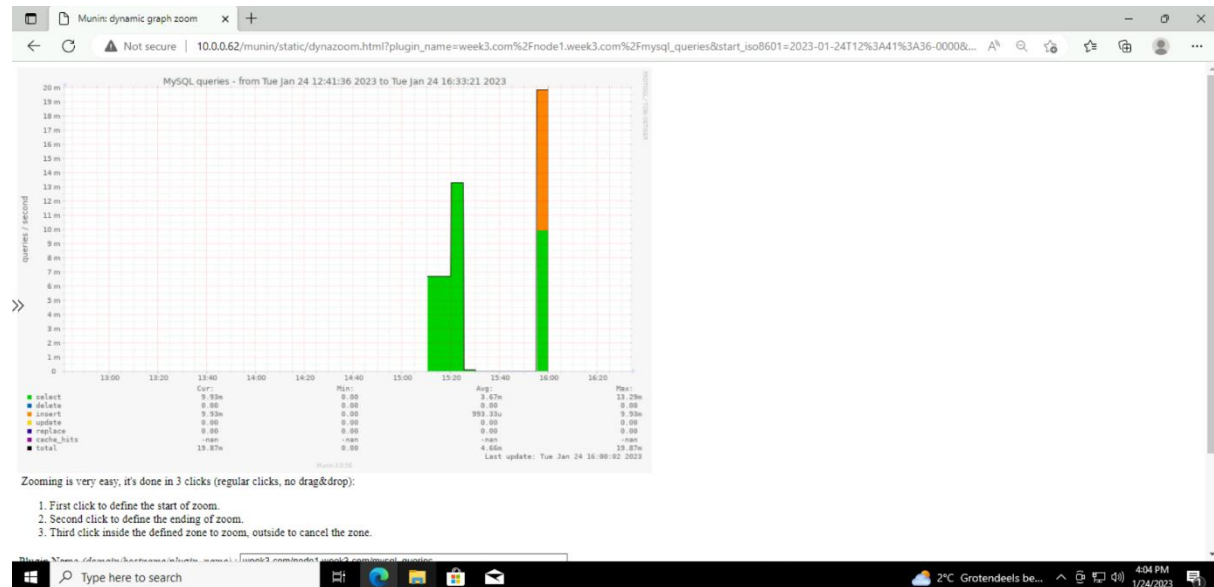
>> a simple host tree
[localhost.localdomain]
    address 127.0.0.1
    use_node_name yes

[node1.week3.com]
    address 10.0.0.63
    use_node_name yes
#
>>
allow ^127\.0\.0\.1$
allow ^::1$
allow ^10\.0\.0\.62$

```

c) Installeer op de server, die bij opdracht b gecreëerd is, Apache en/of NGNIX samen met PHP en MySQL/MariaDB. De logs van de hiervoor genoemde applicaties worden op de log

server verzameld.



Hier kun je zien dat de logging server de SQL Queries van de Node bijhoud.

d) Zorg voor hardening script die de Apache Server beter beveiligd dan de standaard instellingen, bijv. via deze handleiding:

<https://geekflare.com/apache-server-hardening-security/>

Schakeld niet gebruikte apache2 modules uit

```
sudo a2dismod status
```

```
sudo a2dismod autoindex
```

Enable mod_security

```
sudo a2enmod security2
```

Enable mod_headers

```
sudo a2enmod headers
```

Set ServerTokens to Prod

```
sudo sed -i "s/ServerTokens OS/ServerTokens Prod/" /etc/apache2/conf-available/security.conf
```

Set ServerSignature to Off

```
sudo sed -i "s/ServerSignature On/ServerSignature Off/" /etc/apache2/conf-available/security.conf
```

Disable register_globals in PHP

```
sudo sed -i "s/register_globals = On/register_globals = Off/" /etc/php/7.4/apache2/php.ini
```

Disable expose_php in PHP

```
sudo sed -i "s/expose_php = On/expose_php = Off/" /etc/php/7.4/apache2/php.ini
```

```
sudo systemctl restart apache2
```

Bronnen gebruikt:

<https://geekflare.com/apache-web-server-hardening-security/>

https://github.com/ortega-io/apache-hardening/blob/master/http_hard.sh (Gebruikt om het beter te beschrijven, geen code ervan gebruikt)

<https://www.octopus.com/docs/runbooks/runbook-examples/routine/hardening-apache>