Rsyslogserver Remotelogging

Du möchtest dich gerne für unsere Hilfe erkenntlich zeigen . Gerne. Wir bedanken uns bei dir für deine Spende!

Spenden

Zum frei verfügbaren Apt-Repository

GITLAB:

Mit Rsyslog (Default in vielen Distributionen) ist es ein leiches in 15 Minuten einen zentralen voll funktionsfähigen Loggingserver zu bauen. Nachdem Rsyslog schon in **Ubuntu 18.04** vorinstalliert ist, muss man genau garnichts nach installieren.

Am Server passt man das Konfigurationsfile `/etc/rsyslog.conf` wie folgt an. Die folgende Sektion muss einkommentiert werden:

```
...
module(load="imudp")
input(type="imudp" port="514")
...
```

Nun noch ein Template wir denn gerne unsere Logs gerne abgelegt hätten. Das Ganze unter dem gleichen File, gleich darunter:

```
...
$template remote-incoming-logs,"/var/log/remote-
logging/%HOSTNAME%/%PROGRAMNAME%.log"
*.* ?remote-incoming-logs
& ~
...
```

Den Zugriff könnte man noch mit `$AllowedSender TCP, 127.0.0.1, 192.168.10.0/24, *.example.com` einschränken. Jetzt noch das Verzeichnis erstellen und die richtigen Berechtigungen vergeben.

```
mkdri /var/log/remote-logging
chown syslog:syslog /var/log/remote-logging
```

Nun startet man den Server neu:

```
systemctl restart rsyslog.service
```

Somit ist der Serverpart fertig. Also nächstes kommt die Clientkonfiguration.

Rsyslog Clientkonfiguration

Diese besteht aus einer Datei: `/etc/rsyslog.d/51-remote.conf` Der Inhalt ist simpel. Nach dem Anlegen dieser Datei starten wir auch auf unserem Client Rsyslog neu.

```
$PreserveFQDN on

$ActionQueueFileName queue
$ActionQueueMaxDiskSpace 1g
$ActionQueueSaveOnShutdown on
$ActionQueueType LinkedList
$ActionResumeRetryCount -1

*.* @meinserver.supertux.lan:514;RSYSLOG_SyslogProtocol23Format
```

```
systemctl restart rsyslog.service
```

Ab nun loggt unser Client bereits brav zentral im FQDN mit Unterfiles pro Pogramm.

Verschlüsselte Übertragung

Hierfür sind Zertifikate erforderlich (Zertifikat/Key/CA). Dies wird hier bereits vorausgesetzt. Die `rsyslogserver.conf` würde dann so aussehen:

Kommuniziert wird dann zusätzlich über TCP 6514.

```
# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="6514")

$template remote-incoming-logs,"/var/log/remote-
logging/%HOSTNAME%/%PROGRAMNAME%.log"
*.* ?remote-incoming-logs
& ~

###------
$DefaultNetstreamDriver gtls

# certificate files
$DefaultNetstreamDriverCAFile /usr/local/share/ca-certificates/CA.crt
$DefaultNetstreamDriverCertFile /etc/rsyslog.d/cert.crt
$DefaultNetstreamDriverKeyFile /etc/rsyslog.d/cert.key
```

```
#$ModLoad imtcp  # TCP listener
$InputTCPServerStreamDriverMode 1  # run driver in TLS-only mode
$InputTCPServerStreamDriverAuthMode anon
#$InputTCPServerRun 6514  # start up listener at port 10514
```

Die Clientconf würde damit so aussehen: `51-remote.conf`

```
$PreserveFQDN on

$ActionQueueFileName queue
$ActionQueueMaxDiskSpace 1g
$ActionQueueSaveOnShutdown on
$ActionQueueType LinkedList
$ActionResumeRetryCount -1


*.*     @@(o)meinserver.supertux.lan:6514;RSYSLOG_SyslogProtocol23Format

$DefaultNetStreamDriverCAFile /usr/local/share/CA.crt


# make gtls driver the default
$DefaultNetStreamDriver gtls
$ActionSendStreamDriverMode 1   # run driver in TLS-only mode
$ActionSendStreamDriverAuthMode anon
```

Hierfür muss noch ein Paket nach installiert werden:

```
apt install rsyslog-gnutls
```

Quelle Verschlüsselung Rsyslog:
https://www.golinuxcloud.com/secure-remote-logging-rsyslog-tls-certificate/