

Zabbix: Monitor Linux rsyslog configured central log server is reachable with check_log_server_status.sh userparameter script

Author : admin



On modern Linux OS servers on Redhat / CentOS / Fedora and Debian based distros log server service is usually running on the system such as rsyslog (rsyslogd) to make sure the logging from services is properly logged in separate logs under **/var/log**.

A very common practice on critical server machines in terms of data security, where logs produced by rsyslog daemon needs to be copied over network via TCP or UDP protocol immediately is to copy over the /var/log produced logs to another configured **central logging server**. Then later every piece of bit generated by rsyslogd could be overseen by a third party auditor person and useful for any investigation in case of logs integrity is required or at worse case if there is a suspicion that system in question is **hacked** by a malicious hax0r and logs have been "cleaned" up from any traces leading to the intruder (things usually done locally by hackers) or by any automated script exploit tools since yesr.

This *doubled logging of system events to external log server* ipmentioned is *very common practice by companies to protect their log data* and quite useful for logs to be recovered easily later on from the **central logging server** machine that could be also setup for example to use **rsyslogd** to receive logs from other Linux machines in circumstances where some log disappears just like that (things i've seen happen) for any strange reason or gets destroyed by the admins mistake locally on machine / or by any other mean such as filesystem gets damaged. a *very common practice by companies to protect their log data*.

Monitor remote logging server is reachable with userparameter script

Assuming that you already have setup a logging from the *server hostname A* towards the **Central**

logging server log storepool and everything works as expected the next logical step is to have at least some basic way to monitor remote logging server configured is still reachable all the time and respectively rsyslog */var/log/*.** logs gets properly produced on remote side for example with something like a **simple TCP remote server port check and reported in case of troubles in zabbix**.

To solve that simple task for company where I'm employed, I've developed below **check_log_server_status.sh**:

```
#!/bin/bash
# @@ for TCP @ for UDP
# check_log_server_status.sh Script to check if configured TCP / UDP logging server in
/etc/rsyslog.conf is reachable
# report to zabbix
DELIMITER='@@';
GREP_PORT='5145';
CONNECT_TIMEOUT=5;

PORT=$(grep -Ei '".* $DELIMITER.*:$GREP_PORT"' /etc/rsyslog.conf | awk -F : '{ print $2
}' | sort -rn | uniq);

# for i in $(grep -Ei '".* $DELIMITER.*:$GREP_PORT"' /etc/rsyslog.conf | grep -v '\#' | awk
-F "$DELIMITER" '{ print $2 }' | awk -F ':' '{ print $1 }' | sort -rn); do
HOST=$(grep -Ei '".* $DELIMITER.*:$GREP_PORT"' /etc/rsyslog.conf | grep -v '\#' | awk
-F "$DELIMITER" '{ print $2 }' | awk -F ':' '{ print $1 }' | sort -rn)

# echo $PORT

if [[ ! -z $PORT ]] && [[ ! -z $HOST ]]; then
SSH_RETURN=$(/bin/ssh $HOST -p $PORT -o ConnectTimeout=$CONNECT_TIMEOUT
2>&1);
else
echo "PROBLEM Port $GREP_PORT not defined in /etc/rsyslog.conf";
fi

##echo SSH_RETURN $SSH_RETURN;
#exit 1;
if [[ $(echo $SSH_RETURN | grep -i 'Connection timed out during banner exchange' | wc -l)
-eq '1' ]]; then
echo "rsyslogd $HOST:$PORT OK";
fi

if [[ $(echo $SSH_RETURN | grep -i 'Connection refused' | wc -l) -eq '1' ]]; then
echo "rsyslogd $HOST:$PORT PROBLEM";
```

fi

```
#sleep 2;  
#done
```

You can [download a copy of the script check_log_server_status.sh here](#)

Depending on the port the remote rsyslogd central logging server is using configure it in the script with respective port through the **DELIMITER='@@', GREP_PORT='5145', CONNECT_TIMEOUT=5 values.**

The delimiter is setup as usually in **/etc/rsyslog.conf** this the remote logging server for TCP IP is configured with @@ prefix to indicated TCP mode should be used.

Below is example from **/etc/rsyslog.conf** of how the rsyslogd server is configured:

```
[root@Server-hostA /root]# grep -i @@ /etc/rsyslogd.conf  
# central remote Log server IP / port  
*. * @@10.10.10.1:5145
```

To use the script on a machine, where you have a properly configured **zabbix-agentd** service host connected and reporting data to a **zabbix-server** monitoring server.

1. Set up the script under **/usr/local/bin/check_log_server_status.sh**

```
[root@Server-hostA /root]# vim /usr/local/bin/check_log_server_status.sh  
...  
[root@Server-hostA /root]# chmod +x /usr/local/bin/check_log_server_status.sh
```

2. Prepare **userparameter_check_log_server.conf** with **log_server.check**

Item key

```
[root@Server-hostA zabbix_agentd.d]# cat userparameter_check_log_server.conf  
UserParameter=log_server.check, /usr/local/bin/check_log_server_status.sh
```

3. Set in Zabbix some Item such as on below screenshot

Item
Preprocessing

Parent items

OS Linux

*

Name

Check log server status

Type

Zabbix agent (active)

*

Key

log_server.check

Type of information

Character

*

Update interval

1h

Custom intervals

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00:00-24:00

Add

*

History storage period

Do not keep history
Storage period
90d

i

Show value

As is

New application

Logserver
Maintenance
Memory
Network interfaces
OS
Pacemaker
Performance
Processes
rkhunter
Security

Populates host inventory field

None-

Description

Enabled

Update

Clone

Execute now

Test

Clear history and trends

Delete

Cancel

4. Create a Zabbix trigger

5 / 6

Trigger
Tags
Dependencies

Parent triggers

OS Linux

* Name

Logserver is not reachable

Operational data

Severity

Not classified

NORMAL

WARNING

MINOR

MAJOR

CRITIC

* Expression

:dependent.regex (PROBLEM) =1

Add

Expression constructor

OK event generation

Expression

Recovery expression

None

PROBLEM event generation mode

Single

Multiple

OK event closes

All problems

All problems if tag values match

Allow manual close
☒

URL

Description

Enabled
☒

Update

Clone

Delete

Cancel

The redted hided field in **Expression** field should be substituted with your actual hostname on which the monitor script will run.