syslog-ng Premium Edition 7.0.30

Collecting log messages from UDP sources
Copyright 2022 One Identity LLC.
ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this
guide is furnished under a software license or nondisclosure agreement. This software may be used
or copied only in accordance with the terms of the applicable agreement. No part of this guide may
be reproduced or transmitted in any form or by any means, electronic or mechanical, including
photocopying and recording for any purpose other than the purchaser’s personal use without the
written permission of One Identity LLC.

The information in this document is provided in connection with One Identity products. No license,
express or implied, by estoppel or otherwise, to any intellectual property right is granted by this
document or in connection with the sale of One Identity LLC products. EXCEPT AS SET FORTH IN THE
TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT,
ONE IDENTITY ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED
OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-
INFRINGEMENT. IN NO EVENT SHALL ONE IDENTITY BE LIABLE FOR ANY DIRECT, INDIRECT,
CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT
LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF
INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF
ONE IDENTITY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. One Identity makes
no representations or warranties with respect to the accuracy or completeness of the contents of this
document and reserves the right to make changes to specifications and product descriptions at any
time without notice. One Identity does not make any commitment to update the information
contained in this document.

If you have any questions regarding your potential use of this material, contact:

One Identity LLC.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (http://www.OneIdentity.com) for regional and international office
information.

Patents
One Identity is proud of our advanced technology. Patents and pending patents may apply to this
product. For the most current information about applicable patents for this product, please visit our

Trademarks
One Identity and the One Identity logo are trademarks and registered trademarks of One Identity
LLC. in the U.S.A. and other countries. For a complete list of One Identity trademarks, please visit
our website at www.OneIdentity.com/legal. All other trademarks are the property of their
respective owners.

Legend

⚠️ WARNING: A WARNING icon highlights a potential risk of bodily injury or property
damage, for which industry-standard safety precautions are advised. This icon is
often associated with electrical hazards related to hardware.

⚠️ CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data
if instructions are not followed.

dsyslog-ng PE Collecting log messages from UDP sources
Updated - 11 May 2022, 14:23
Version - 7.0.30
## Contents

<table>
<thead>
<tr>
<th>Problems</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolving UDP message loss problems</td>
<td>5</td>
</tr>
<tr>
<td>About us</td>
<td>8</td>
</tr>
<tr>
<td>Contacting us</td>
<td>9</td>
</tr>
<tr>
<td>Technical support resources</td>
<td>10</td>
</tr>
</tbody>
</table>
Problems

Central logging is based on TCP connections in most cases (or even encrypted TCP connections) as it provides several reliability features that are simply not available using UDP. Still, there are certain situations, when you have to use UDP. One use case is when the company standard syslog configuration for servers contains only a single, common UDP destination, and it cannot be altered because of company IT policies. The other, more common use case is, that network devices, for example, routers, switches or firewalls send their logs using UDP. Most of the time the TCP implementation of syslog is completely missing from these devices. In some cases it is there, but badly broken and therefore avoided by the users.

When compared to TCP, UDP is more lightweight, and it consumes less computing resources. However, this is probably the only positive feature of UDP when discussing log message transfer. On the sender side UDP sends packets with a “fire and forget” method, meaning that it does not ensure that the packet was received, there is no error handling, acknowledgment, retransmission or timeout and therefore it is prone to losing messages.

There are several points during the transmission where messages can be lost when using UDP:

- When there are multiple hops, a burst of log messages can be lost.
- Kernel network card driver level: virtualization might cause problems, also certain drivers and cards are unable to keep up with high-traffic and therefore log messages can be lost.
- Kernel buffer size: when syslog-ng Premium Edition (syslog-ng PE) cannot read fast enough from the kernel, buffers are filled and log messages can be lost.
- In syslog-ng PE when the destination side is not fast enough and buffers are filled. For example, when the disk I/O is held by some other application, therefore the messages cannot get written by syslog-ng PE. Even on a dedicated server, just browsing the stored logs can be a reason for that.

The effects of these problems can be reduced, but keep in mind that UDP is not a reliable protocol.
**Resolving UDP message loss problems**

Even if UDP is not a reliable protocol, and message loss cannot be completely avoided, there are many ways to improve the situation. If you only have a few incoming UDP messages, you might not need any or all of these safeguards. However, as soon as you expect higher UDP traffic or burst of UDP packages, it is definitely worth implementing them.

**Using the udp-balancer() source**

The `udp-balancer()` source allows you to use multiple CPU cores to process the incoming UDP messages at a very high message rate, depending on the available hardware resources, incoming message size, and your syslog-ng Premium Edition (syslog-ng PE) configuration.

**NOTE:** This feature requires a Linux kernel that supports the `SO_REUSEPORT` kernel option, so it is only supported on selected platforms.

For more information about the `udp-balancer()` source, its limitations and recommended use cases, declaration and configuration examples, and the `udp-balancer()` source options, see "udp-balancer: Receiving UDP messages at very high rate" in the Administration Guide.

**Collect messages close to the source**

UDP packages easily get lost if they have to travel through switches and routers. This kind of loss is difficult to detect, unless you can count the number of messages both on the sending and receiving side. To avoid message loss, install a syslog-ng PE relay collecting UDP messages as close as possible to the log source, ideally on the same switch or at least on the same subnetwork. Forward the logs to your central log server using a more reliable protocols, such as TCP or ALTP.

**Use the right network card and network driver**

Some network cards handle high load better than others. In high-traffic environments it is worth using server class network cards that can off-load many of the tasks. The same hardware can sometimes be used with different drivers, so select the driver that suits your needs best.
To generate syslog traffic for stress testing your environment, use loggen (a tool bundled with syslog-ng PE). To check for dropped packages at hardware level, use ifconfig or ethtool.

**Use physical machines instead of virtual ones**

Full hardware virtualization, for example, VMware can degrade performance, especially during hardware access like high speed networking. In some situations, using the right network driver inside the virtual machine can resolve this problem. On a loaded host machine even the right network driver will not solve these issues, therefore it is better to move log collection to a physical machine.

**Use large buffers in kernel**

If syslog-ng PE cannot read the messages fast enough from the UDP socket, the kernel receive buffers will start to fill and after the configured limit has been reached, the kernel will start discarding messages. In this case, it is necessary to adjust the buffer size accordingly. To raise the size of the kernel receive buffers, use the sysctl command to tune the net.core.rmem_max parameter. Next, raise the size of the so-rcvbuf option of the syslog-ng source definition as well, so that syslog-ng PE is capable of utilizing the larger kernel receive buffers. In a high-traffic environment as high as 256MB might be necessary:

```
sysctl -w net.core.rmem_max=268435456
```

Enter the value in bytes. In the example above, 256*1024*1024=268435456 bytes. As a rule of thumb, this buffer size should be enough to accommodate incoming peak message rate for at least one second.

⚠️ **CAUTION:**

*Any issue causing the loss of buffer content can result in higher message loss because of the higher buffer size. To minimize risk, it is important to determine the required buffer size instead of using a buffer that is higher than what is absolutely necessary.*

To monitor packet loss, use the following command:

```
netstat -su | grep "receive errors"
```

**Tuning syslog-ng PE for UDP**

There are several configuration possibilities in syslog-ng PE that are related to using UDP protocol. To be able to handle message bursts, increase the value of the log_fifo_size() option. To match the value configured for net.core.rmem_max in the previous step, increase the value of the so-rcvbuf() accordingly.

Flow control in syslog-ng PE slows down message reception if the destination slowed down for some reason. This works fine when messages are coming from TCP sources, as the sender side will notice slower reception and send messages at a lower rate. UDP, as a stateless protocol cannot handle this situation, the sender side will never notice that the receiving end slowed down and this will result in immediate message loss. Therefore in case of UDP sources, do not enable flow control for the destination.
When you use a syslog-ng Store Box as the central logserver, flow-control is always applied to the log paths where messages are written into logspaces in order to avoid message loss. Therefore, log processing may slow down if logs are massively searched by dozens of users (this is similar to the example described in the “Problems” section above). Contrary to the highly configurable syslog-ng PE, you do not have the ability to fine-tune the low level OS level buffers underneath the syslog-ng Store Box log management application. Therefore, you may turn the UDP-based log collection into TCP-based already on a lower message rate.

The UDP source driver in syslog-ng PE is not multi-threaded. A single UDP source runs only in a single thread on a single CPU core. If you have a high message rate and multiple CPU cores, define multiple UDP sources in your syslog-ng PE configuration. This way load is distributed among multiple CPU cores, ensuring that CPU does not become a bottleneck. Still in this case, if the messages received from several UDP sources are written into the very same file, the destination side can become a bottleneck.

The consequence is that UDP-based collection always comes with a risk of message loss. The possibility of losing messages is different in each environment, therefore the situation might be examined in all cases in order to select the right safe guards that are sufficient to eliminate the risk of losing messages.
One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.
Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit https://www.oneidentity.com/company/contact-us.aspx.
Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product