LOGZILLA DOCUMENTATION

Syslog Troubleshooting

Diagnose syslog reception issues on a LogZilla host using logzilla.log inspection, tcpdump packet capture, and syslog-ng debug output

Administration · Generated October 30, 2026 · logzilla.ai/docs/administration/syslog-troubleshooting

© LogZilla Corporation · logzilla.ai

Syslog Troubleshooting

Practical checks and tools for diagnosing syslog reception issues in LogZilla. The focus is on verifying that events arrive at the host, that syslog-ng is processing them, and that debug outputs are available.

Check the LogZilla log

Use the primary LogZilla log to look for errors and pipeline state:

```
sudo tail -f /var/log/logzilla/logzilla.log
```

Verify that the source is sending

Confirm that traffic from the sender reaches the LogZilla server. Adjust the port and interface as needed for the environment.

```
# Listen on the default gateway interface for UDP 514 (RFC 3164)
sudo tcpdump -vvv -i $(awk '$2 == 00000000 { print $1 }' /proc/net/route) \
  udp port 514
```

If a different interface is required:

```
sudo tcpdump -vvv -i p1p1 udp port 514
```

Enable syslog-ng debug mode (advanced)

After confirming traffic arrives, enable syslog-ng debugging in the `lz_syslog` container to inspect parsing behavior.

```
sudo docker exec -it lz_syslog bash -c 'syslog-ng-ctl debug --set=on'
sudo docker logs lz_syslog --tail 100 -f
```

Disable debug mode after checking the output:

```
sudo docker exec -it lz_syslog bash -c 'syslog-ng-ctl debug --set=off'
```

Syslog debugging controls

Use the UI (recommended)

Troubleshooting toggles are available in the web interface:

Open `Settings` → `System Settings` → `Syslog Daemon`.
Toggle `Syslog Debug` to `On` to write to `/var/log/logzilla/syslog/debug.log`.
Toggle `Syslog Debug Json` to `On` to write to `/var/log/logzilla/syslog/debug-json.log`.
Disable these toggles after troubleshooting to avoid extra load and disk usage.
View logs using:

```
sudo tail -F /var/log/logzilla/syslog/debug.log
sudo tail -F /var/log/logzilla/syslog/debug-json.log
```

Use the console (fallback)
When the UI is unavailable, the same settings can be changed via the CLI:

```
sudo logzilla settings update SYSLOG_DEBUG=true
sudo logzilla settings update SYSLOG_DEBUG_JSON=true
```

Disable after troubleshooting:

```
sudo logzilla settings update SYSLOG_DEBUG=false
sudo logzilla settings update SYSLOG_DEBUG_JSON=false
```

Generate local test messages

Use `logger` to generate local test events and verify they appear in LogZilla:

```
sudo logger -T -P 514 --rfc3164 -n localhost -p local0.emerg \
  -t "test" "rfc3164 event test on TCP Port 514 from $(hostname)"

sudo logger -d -P 514 --rfc3164 -n localhost -p local0.emerg \
  -t "test" "rfc3164 event test on UDP Port 514 from $(hostname)"
```

Raw capture for support

If events still do not appear, capture a sample for support analysis. Adjust duration (`-G`) based on volume and reproduction window.

```
# Example: capture 3 hours
# 86400 would be 24 hours
# Include fragmented packets and compress the capture
```

```
sudo tcpdump -i $(awk '$2 == 00000000 { print $1 }' /proc/net/route) \
  "udp port 514 or (ip[6:2] & 0x1fff) != 0" \
  -nnvvXSs 0 -G 10800 -W 1 -z gzip -w /tmp/$(hostname).pcap
```

Include LogZilla version information in support tickets:

```
sudo logzilla version
```