LOGZILLA DOCUMENTATION

Syslog Basics

Syslog protocol fundamentals covering RFC 3164 BSD format, RFC 5424 structured format, and RFC 6587 TCP and TLS transport for reliable log delivery

Administration · Generated April 27, 2026 · logzilla.ai/docs/administration/syslog-basics

Syslog is the standard protocol for operational telemetry across servers, network infrastructure, security appliances, and applications. Originally developed in the 1980s by Eric Allman as part of the Sendmail project, syslog enables centralized logging for faster incident triage, auditability, and long-term analysis.

Modern syslog implementations support multiple transport protocols and security options to meet current operational and compliance requirements.

Syslog Standards Overview

Syslog has evolved through multiple RFC specifications:

- **RFC 3164 (2001)**: The traditional BSD syslog format, widely supported but with limitations in timestamp precision and message structure
- **RFC 5424 (2009)**: The modern syslog standard with enhanced structure, precise timestamps, and structured data support
- **RFC 6587**: Defines transmission protocols (TCP/TLS) for reliable delivery

RFC 3164 Message Structure (Legacy Format)

Traditional syslog messages consist of three main components:

PRI (Priority) Encodes both facility and severity in a single numeric value. The facility identifies the originating subsystem (e.g., auth, daemon, local0-local7), while severity indicates the importance level.

Header Contains the timestamp and hostname of the sending device. Proper time synchronization using NTP is essential for accurate log correlation.

Message The payload containing the program name and event details.

RFC 3164 Format Example

```
<34>Oct 11 22:14:15 myhost su[1234]: 'su root' failed for jdoe on /dev/pts/2
```

RFC 5424 Message Structure (Modern Format)

RFC 5424 provides enhanced structure and capabilities:

PRI (Priority) Same as RFC 3164 - encodes facility and severity.

VERSION Identifies the syslog protocol version (always "1" for RFC 5424).

TIMESTAMP High-precision timestamp in ISO 8601 format with timezone information and optional fractional seconds.

HOSTNAME Fully qualified domain name or IP address of the originating device.

APP-NAME Name of the application or process generating the message.

PROCID Process ID or thread ID of the generating process.

MSGID Message type identifier for categorizing similar events.

STRUCTURED-DATA Optional key-value pairs in standardized format for machine parsing and analysis.

MSG Free-form message text, typically human-readable.

Facilities and Severities

Severity Levels (0-7)

Code	Level	Description
0	Emergency	System unusable
1	Alert	Immediate action required
2	Critical	Critical conditions
3	Error	Error conditions
4	Warning	Warning conditions
5	Notice	Normal but significant condition
6	Informational	Informational messages
7	Debug	Debug-level messages

Common Facilities

Facilities identify the source system or application type:

- **kern**: Kernel messages
- **user**: User-level messages

- **mail**: Mail system messages
- **daemon**: System daemon messages
- **auth**: Security/authorization messages
- **local0-local7**: Custom facility codes for specific device types

Many organizations map device types (firewalls, routers, servers) to specific local facilities for easier routing and filtering.

Transport Protocols

UDP (Port 514)

- **Advantages**: Lightweight, ubiquitous support, minimal overhead
- **Disadvantages**: No delivery guarantees, potential message loss
- **Use Cases**: Legacy devices, high-volume environments where occasional loss is acceptable

TCP (Port 514)

- **Advantages**: Reliable delivery, ordered messages, flow control
- **Disadvantages**: Higher overhead, connection management complexity
- **Use Cases**: Critical systems requiring guaranteed delivery

TLS (Port 6514)

- **Advantages**: Encrypted transport, peer authentication, data integrity
- **Disadvantages**: Certificate management overhead, higher resource usage
- **Use Cases**: Sensitive data, cross-network boundaries, compliance requirements

Transport Selection Guidelines

Use TLS when:

- Transmitting sensitive or regulated data
- Crossing network boundaries or untrusted networks
- Compliance requirements mandate encryption
- Peer authentication is required

Use TCP when:

- Reliability and ordered delivery are critical

- Message loss is unacceptable
- TLS overhead is not justified

Use UDP when:

- Working with legacy or constrained devices
- High-volume environments where occasional loss is tolerable
- Minimal overhead is required

LogZilla Syslog Implementation

LogZilla utilizes the industry-standard syslog-ng (https://syslog-ng.org) daemon to receive syslog messages and forward them to LogZilla's processing architecture. This implementation supports all standard syslog transports and provides advanced parsing and filtering capabilities.

Supported Protocols

- **UDP 514**: Standard syslog reception
- **TCP 514**: Reliable syslog reception
- **TLS 6514**: Encrypted syslog reception with certificate validation
- **Custom ports**: Configurable for specific network requirements

Implementation Best Practices

Time Synchronization

Ensure all syslog sources maintain accurate time synchronization using NTP. Inconsistent timestamps complicate log correlation and incident analysis. Standardize on UTC across all systems where possible.

Facility Mapping

Establish consistent facility usage across the organization:

- Map device types to specific local facilities (local0-local7)
- Document facility assignments for operational teams
- Use facilities for automated routing and filtering rules

Message Formatting

Modern syslog implementations benefit from structured data:

- **Prefer RFC 5424**: Use RFC 5424 format when supported by devices for enhanced parsing and analysis capabilities
- **Structured Data**: Leverage RFC 5424 structured data elements for machine-readable context
- **Consistent Timestamps**: Use high-precision ISO 8601 timestamps with timezone information
- **Meaningful MSG-IDs**: Implement consistent message type identifiers for automated processing
- **Security**: Avoid embedding sensitive data in log messages

Security Considerations

- Use TLS transport for sensitive or regulated data
- Implement certificate-based authentication where required
- Configure appropriate firewall rules for syslog traffic
- Monitor for unusual syslog traffic patterns

Common Implementation Patterns

Enterprise Network Infrastructure

Large networks typically implement hierarchical syslog collection:

- Edge devices forward to regional collectors via UDP for simplicity
- Regional collectors aggregate and forward to central systems via TLS
- Central systems apply parsing, enrichment, and routing rules
- Archive systems maintain long-term retention for compliance

Security-Focused Deployments

Security-sensitive environments prioritize integrity and confidentiality:

- All syslog transport uses TLS with mutual authentication
- Certificate-based device identity validation
- Structured data formats for consistent parsing
- Real-time forwarding to SIEM with preprocessing to reduce noise

Troubleshooting Common Issues

Message Truncation

Syslog messages have practical size limits:

- UDP: typically 1024 bytes to avoid fragmentation
- TCP/TLS: larger messages supported but check receiver limits
- Use structured data to organize information within constraints

Timestamp Problems

Inconsistent timestamps complicate log correlation:

- Synchronize device clocks using NTP
- Configure consistent timestamp formats (prefer RFC 3339)
- Document systems that cannot be standardized

Certificate Management

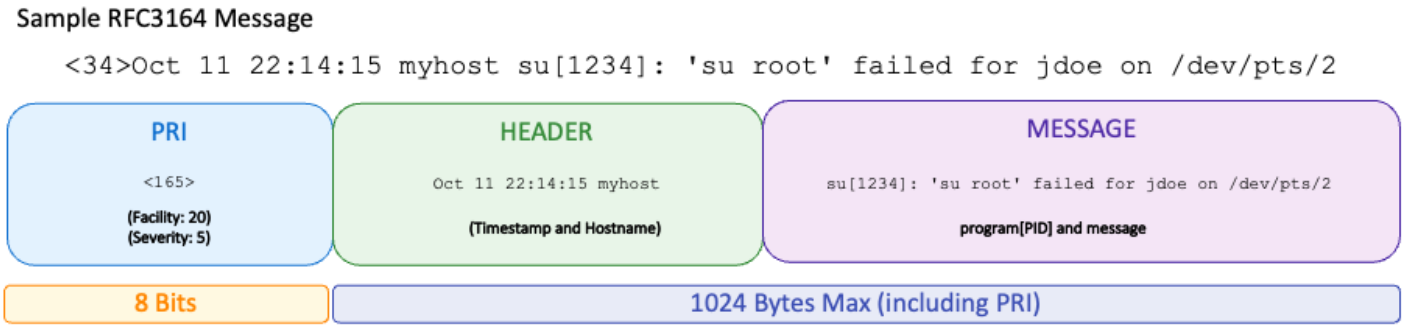
TLS deployments require ongoing certificate lifecycle management:

- Automate certificate renewal before expiration
- Monitor certificate validity across all endpoints
- Test certificate validation in staging environments.

Relays are used to forward logs from local networks to remote networks. This is the most reliable and common way to ensure message reception on the primary server when utilizing a wide-area network. For help configuring a relay, refer to the Syslog Relays (https://www.logzilla.ai/docs/administration/syslog-relays) section.

Syslog Message Format and Contents

RFC 3164 Format (Traditional)

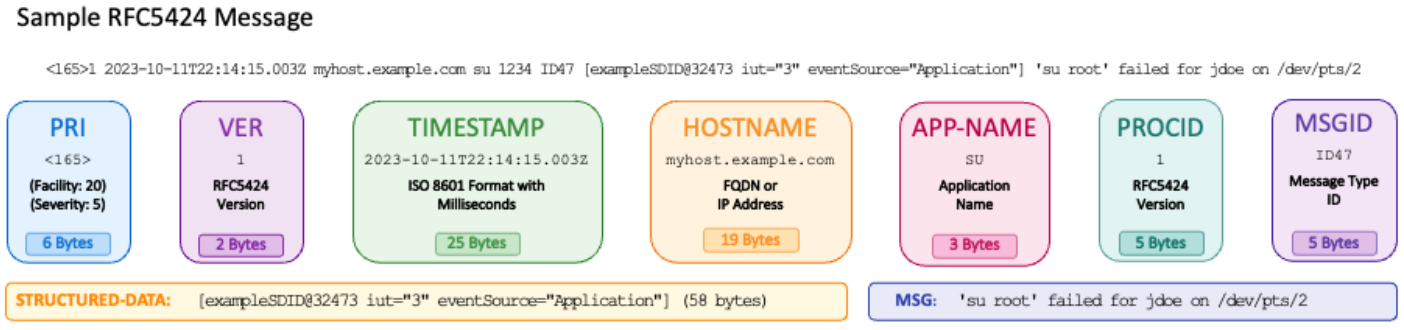


The traditional RFC 3164 syslog message has three distinct parts:

- **PRI** (priority)
- **HEADER** (timestamp and hostname)
- **MSG** (message text)

For RFC 3164 compliant events, the total length cannot exceed 1024 bytes.

RFC 5424 Format (Modern)



RFC 5424 messages provide enhanced structure with these components:

- **PRI** (priority) - Same as RFC 3164
- **VERSION** - Protocol version ("1")
- **TIMESTAMP** - ISO 8601 format with high precision
- **HOSTNAME** - FQDN or IP address

- **APP-NAME** - Application identifier
- **PROCID** - Process/thread identifier
- **MSGID** - Message type identifier
- **STRUCTURED-DATA** - Machine-readable key-value pairs
- **MSG** - Human-readable message text

RFC 5424 messages can be significantly larger than the 1024-byte RFC 3164 limit, with practical limits determined by transport protocol and implementation.

Structured Data in RFC 5424

Structured data provides machine-readable context using standardized syntax:

Structured Data Element Format

```
[SD-ID param1="value1" param2="value2"]
```

Multiple Elements Example

```
[timeQuality tzKnown="1" isSynced="1"][origin software="rsyslogd" swVersion="8.24.0"]
```

Common Structured Data Elements

timeQuality: Timestamp reliability information

- `tzKnown`: "1" if timezone is known, "0" if unknown
- `isSynced`: "1" if clock is synchronized, "0" if not

origin: Message origin information

- `software`: Name of originating software
- `swVersion`: Software version
- `ip`: IP address of originating device

meta: Additional metadata

- `sequenceId`: Message sequence number
- `sysUpTime`: System uptime when message was generated

Enterprise-Specific Elements

Organizations can define custom structured data elements using Private Enterprise Numbers (PEN):

```
[myCompany@32473 eventType="login" userId="jdoe" result="failure"]
```

Message Size Considerations

RFC 3164 Limitations

- Maximum message size: 1024 bytes (including PRI)
- No standardized timestamp format
- Limited structure for automated parsing
- Hostname length restrictions

RFC 5424 Advantages

- No fixed message size limit (transport-dependent)
- Standardized ISO 8601 timestamps with microsecond precision
- Structured data for machine parsing
- Clear field separation and encoding rules
- Support for UTF-8 encoding

Transport-Specific Limits

UDP Transport:

- Practical limit: ~1400 bytes to avoid IP fragmentation
- Consider network MTU when sizing messages
- Fragmented packets may be dropped by firewalls

TCP/TLS Transport:

- Much larger messages supported (typically 64KB+)
- Reliable delivery ensures message integrity
- Flow control prevents receiver overload

Syslog PRI Code

The Priority field is an 8-bit number that represents both the `Facility` and `Severity` of the message. The three least significant bits represent the Severity of the message (with three bits you can represent eight different Severities), and the other five bits represent the Facility of the message.

Note: Syslog Daemons (running on the syslog receiver) do not generate these Priority and Facility values. The values are created by the syslog sender (applications or hardware) from which the event is generated.

Syslog Facilities

Syslog messages are broadly categorized on the basis of the sources that generate them. These categories, referred to as `Facilities`, are represented by integers in the syslog packet. The `local` facilities are not reserved; the processes and applications that do not have pre-assigned Facility values may choose any of the eight local use facilities.

Integer	Facility
0	Kernel messages
1	User-level messages
2	Mail system
3	System daemons
4	Security/authorization messages
5	Messages generated internally by Syslogd
6	Line printer subsystem
7	Network news subsystem
8	UUCP subsystem
9	Clock daemon
10	Security/authorization messages
11	FTP daemon
12	NTP subsystem

Integer	Facility
13	Log audit
14	Log alert
15	Clock daemon
16	Local use 0 (local0)
17	Local use 1 (local1)
18	Local use 2 (local2)
19	Local use 3 (local3)
20	Local use 4 (local4)
21	Local use 5 (local5)
22	Local use 6 (local6)
23	Local use 7 (local7)

Syslog Severities

The log sender (device or software generating the message) specifies the severity of that message using single-digit integers 0-7

Note: When configuring the sending device or application, the recommended logging levels are 0-6 under normal operation, level 7 (debug) should only be used for local troubleshooting on that system.

Integer	Facility
0	Emergency: System is unusable
1	Alert: Action must be taken immediately
2	Critical: Critical conditions
3	Error: Error conditions
4	Warning: Warning conditions
5	Notice: Normal but significant condition

Integer	Facility
6	Informational: Informational messages
7	Debug: Debug-level messages

RFC 5424 Implementation Guidelines

Timestamp Best Practices

Use High-Precision Timestamps:

```
2023-10-11T22:14:15.003Z        # Millisecond precision
2023-10-11T22:14:15.003456Z     # Microsecond precision
```

Include Timezone Information:

```
2023-10-11T22:14:15.003+00:00   # UTC with offset
2023-10-11T22:14:15.003-05:00   # EST with offset
```

Application Name Guidelines

- Use consistent, descriptive application names
- Avoid spaces and special characters
- Consider using service names for system services
- Examples: `sshd`, `httpd`, `myapp`, `backup-service`

Process ID Usage

- Include actual process ID when available
- Use thread ID for multi-threaded applications
- Use "-" (NILVALUE) if not applicable
- Helps with process-specific log correlation

Message ID Strategies

Consistent Categorization:

```
ID01    # Authentication events
ID02    # Authorization events
ID03    # System errors
ID04    # Performance alerts
```

Event-Specific IDs:

```
LOGIN_SUCCESS    # Successful login
LOGIN_FAILURE    # Failed login attempt
CONFIG_CHANGE    # Configuration modification
```

Structured Data Design

Security Events:

```
[auth@32473 user="jdoe" src_ip="192.168.1.100" method="password"]
```

Performance Metrics:

```
[perf@32473 cpu_usage="85.2" memory_mb="2048" response_time_ms="150"]
```

Application Context:

```
[app@32473 module="payment" transaction_id="tx_12345" customer_id="cust_67890"]
```

Migration from RFC 3164 to RFC 5424

Assessment Phase

Device Inventory: Catalog all syslog sources and their RFC 5424 support
Parser Compatibility: Verify log management systems support RFC 5424
Network Capacity: Assess bandwidth impact of larger messages
Integration Testing: Test RFC 5424 with existing SIEM/analysis tools

Gradual Migration Strategy

Pilot Group: Start with modern devices supporting RFC 5424
Dual Format: Run both formats during transition period
Parser Updates: Update log processing rules for new format

Legacy Accommodation: Maintain RFC 3164 support for older devices
Monitoring: Track message delivery and parsing success rates

Common Migration Challenges

Increased Message Size:

- Monitor network utilization
- Adjust UDP buffer sizes if needed
- Consider TCP transport for larger messages

Parser Compatibility:

- Update regular expressions for new format
- Handle NILVALUE ("-") in empty fields
- Parse structured data elements correctly

Timestamp Parsing:

- Update timestamp parsing for ISO 8601 format
- Handle timezone information correctly
- Preserve microsecond precision where supported

Custom Syslog Configurations

For LogZilla-specific syslog configurations and advanced receiving options, refer to the Receiving Syslog Events (https://www.logzilla.ai/docs/receiving-data/receiving-syslog-events) section.

LogZilla RFC 5424 Support

LogZilla fully supports both RFC 3164 and RFC 5424 formats:

- **Automatic Detection**: Identifies message format automatically
- **Structured Data Parsing**: Extracts structured data elements into searchable fields
- **High-Precision Timestamps**: Preserves microsecond timestamp precision
- **Unicode Support**: Handles UTF-8 encoded messages correctly
- **Large Message Support**: Processes messages beyond 1024-byte limit via TCP/TLS transport