LOGZILLA DOCUMENTATION

Syslog Relays

Configure syslog-ng and rsyslog relays to forward events to a LogZilla server over TCP, UDP, TLS, or HTTP, including token-authenticated HTTPS

Administration · Generated May 3, 2026 · logzilla.ai/docs/administration/syslog-relays

Syslog Relays

Syslog relays (sender-side systems) forward logs to LogZilla using syslog-ng or rsyslog over TCP/UDP, TLS, or HTTP/HTTPS. Receiver-side configuration appears in HTTP Receiver Settings (https://www.logzilla.ai/docs/administration/http-receiver-settings).

Architecture overview

```
remote site/devices  ->  relay (syslog-ng/rsyslog)  ->  LogZilla server
                                 |                      (syslog or HTTP)
                                 +-- TLS/JSON/filters
```

Prerequisites

- LogZilla server host and port information.
- For HTTP/HTTPS: an access token and the `/incoming` path on the LogZilla server.
- Optional TLS materials for syslog TLS (certificates/keys) when using port 6514.

syslog-ng (TCP/UDP)

The following example receives on TCP/UDP 514 and forwards to LogZilla over TCP 514. Adjust ports and transports as needed.

```
# filename: /etc/syslog-ng/conf.d/logzilla-relay.conf

options {
  flush_lines(100);
  threaded(yes);
  use_dns(yes);
  use_fqdn(no);
  keep_hostname(yes);
  dns-cache-size(2000);
  dns-cache-expire(87600);
};

source s_network {
  network(transport("tcp") port(514));
  network(transport("udp") so_rcvbuf(1048576) port(514));
};

destination d_logzilla {
  network("<LOGZILLA_HOST>" port(514) transport(tcp));
};
```

```
log {
  # Disable s_src if local events are not needed
  source(s_src);
  source(s_network);
  destination(d_logzilla);
  flags(flow-control);
};
```

syslog-ng with TLS (6514)

Use TLS (RFC 5425) for encrypted forwarding. Ensure certificate paths are valid on the relay host.

```
destination d_logzilla_tls {
  network(
    "<LOGZILLA_HOST>"
    port(6514)
    transport(tls)
    tls(
      ca_dir("/etc/syslog-ng/ca.d")
      key_file("/etc/syslog-ng/key.d/relay-key.pem")
      cert_file("/etc/syslog-ng/cert.d/relay-cert.pem")
    )
  );
};
```

Update the `log {}` path to use `d_logzilla_tls` when forwarding securely.

Notes:

- **Relay trust store (`ca_dir`/`ca-file`)**: Use `ca_dir("/etc/syslog-ng/ca.d")` or `ca-file("/path/to/ca.pem")` to trust the LogZilla server certificate. The server's public cert can be copied to the relay and referenced with `ca-file("/path/to/tls.crt")`, or use a proper CA bundle.
- **Relay client key/cert are optional**: `key_file` and `cert_file` on the relay are only needed for mutual TLS (client authentication). For typical server-auth-only TLS, omit them.
- **Server-side TLS materials**: The LogZilla server's TLS listener uses `SYSLOG_TLS_CERT_FILE` and `SYSLOG_TLS_KEY_FILE` (defaults: `/etc/logzilla/syslog-ng/tls.crt` and `/etc/logzilla/syslog-ng/tls.key`). These files live on the server and are not copied to relays.
- **Client certs not required by default**: The server's syslog TLS listener does not require client certificates by default (peer verification is optional). Enable mutual TLS only if users have configured the server to require and trust client certs.

Minimal server-auth-only example (no client key/cert on the relay):

```
destination d_logzilla_tls {
  network(
```

```
    "<LOGZILLA_HOST>"
    port(6514)
    transport(tls)
    tls(
      # Either trust the issuing CA or the server's public cert
      ca-file("/path/to/logzilla-ca.crt")
      # or: ca_dir("/etc/syslog-ng/ca.d")
    )
  );
};
```

Background on certificates and TLS ports can be found in Using TLS Tunnels (https://www.logzilla.ai/docs/administration/using-tls-tunnels).

syslog-ng over HTTP/HTTPS (to /incoming)

Forward events using HTTP/HTTPS to LogZilla's HTTP Receiver at `/incoming`. Include an access token in headers.

```
destination d_logzilla_http {
  http(
    url("https://<LOGZILLA_HOST>:<PORT>/incoming")
    method("POST")
    user-agent("syslog-ng Relay")
    headers(
      "Content-Type: application/json",
      "Authorization: token <YOUR_TOKEN>"
    )
    body-prefix("{\"events\": [\n")
    delimiter(",\n")
    body('$(format-json
      --pair priority=int($PRI)
      --pair host="$HOST"
      --pair program="$PROGRAM"
      --pair message="$MESSAGE"
    )')
    body-suffix("\n]}")
    batch-lines(10000)
    batch-bytes(10485760)
    batch-timeout(500)
  );
};

log {
  source(s_src);
  destination(d_logzilla_http);
  flags(flow-control);
};
```

- Endpoint reference and minimal tests appear in HTTP Event Receiver (https://www.logzilla.ai/docs/receiving-data/http-event-receiver).

- Interactive API docs are available at `/incoming/docs` on the LogZilla server. Accepted token header forms are documented in the HTTP Event Receiver documentation.

rsyslog with TLS

Use port 6514 for TLS syslog; 443 may be used only in constrained environments.

```
$DefaultNetstreamDriver gtls
$DefaultNetstreamDriverCAFile /etc/rsyslog.d/keys/ca.pem
$DefaultNetstreamDriverCertFile /etc/rsyslog.d/keys/client-cert.pem
$DefaultNetstreamDriverKeyFile /etc/rsyslog.d/keys/client-key.pem

$ActionSendStreamDriverAuthMode x509/name
$ActionSendStreamDriverPermittedPeer <LOGZILLA_HOST>
$ActionSendStreamDriverMode 1

*.* action(type="omfwd" Target="<LOGZILLA_HOST>" Port="6514" Protocol="tcp")
```

Best practices

Load balance high-volume sources across multiple relays.

Enable disk buffering to prevent loss during outages:

```
destination d_logzilla {
  network(
    "<LOGZILLA_HOST>"
    port(514)
    transport(tcp)
    disk-buffer(
      mem-buf-size(10000)
      disk-buf-size(2000000)
      reliable(yes)
    )
  );
};
```

Tag forwarded messages with relay identification (e.g., `relay_id`).

Apply coarse filtering at the relay to reduce traffic.

For WAN links, deploy both local and central relays for resilience.

Verification

- End-to-end checks and packet captures: Syslog Troubleshooting (https://www.logzilla.ai/docs/administration/syslog-troubleshooting).
- HTTP relays: endpoint details and sample requests: HTTP Event Receiver (https://www.logzilla.ai/docs/receiving-data/http-event-receiver).