TCP-Relay (legacy, no longer maintained)

Marc Huber

COLLABORATORS

	TITLE : TCP-Relay (legacy, no longer maintained)	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Marc Huber	November 17, 2026	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Introduction **1**

1.1 Download . . . 1

2 Operation **1**

2.1 Command line syntax . . . 1

2.2 Signals . . . 1

2.3 Event mechanism selection . . . 1

2.4 Configuration Syntax . . . 2

2.4.1 Railroad Diagram . . . 3

2.5 Sample configuration . . . 3

3 Bugs **4**

4 Copyrights and Acknowledgements **4**

1 Introduction

tcprelay is a TCP connection forwarder with load balancing capabilities. If compiled with TLS support, it may be used as SSL encryption wrapper.

1.1 Download

You can download the source code from the GitHub repository at https://github.com/tacplus-ng/event-driven-servers/. On-line documentation is available via https://tacplus-ng.github.io/event-driven-servers/doc/index.html, too.

2 Operation

This section gives a brief and basic overview on how to run **tcprelay**.

In earlier versions, **tcprelay** wasn't a standalone program but had to be invoked by **spawnd**. This has changed, as **spawnd** is now part of the **tcprelay** binary. However, using a dedicated **spawnd** process is still possible and, more importantly, the **spawnd** configuration options and documentation remain valid.

tcprelay may use auxilliary **MAVIS** backend modules for authentication and authorization.

2.1 Command line syntax

The only mandatory argument is the path to the configuration file:

```
tcprelay [ -P ] [ -d level ]  [ -i child_id ] configuration-file [ id ]
```

If the program was compiled with CURL support, *configuration-file* may be an URL.

Keep the `-P` option in mind - it is imperative that the configuration file supplied is syntactically correct, as the daemon won't start if there are any parsing errors at start-up.

The `-d` switch enables debugging. You most likely don't want to use this. Read the source if you need to.

The `-i` option is only honoured if the build-in **spawnd** functionality is used. In that case, it selects the configuration ID for **tcprelay**, while the optional last argument *id* sets the ID of the **spawnd** configuration section.

2.2 Signals

Both the master (that's the process running the **spawnd** code) and the child processes (running the **tcprelay** code) intercept the `SIGHUP` signal:

- The master process will restart upon reception of `SIGHUP`, re-reading the configuration file. The child processes will recognize that the master process is no longer available. It will continue to serve the existing connections and terminate when idle.
- If `SIGHUP` is sent to a child process it will stop accepting new connections from its master process. It will continue to serve the existing connections and terminate when idle.

2.3 Event mechanism selection

Several level-triggered event mechanisms are supported. By default, the one best suited for your operating system will be used. However, you may use the environment variable `IO_POLL_MECHANISM` to select a specific one.

The following event mechanisms are supported (in order of preference):

- port (Sun Solaris 10 and higher only, `IO_POLL_MECHANISM=32`)

- kqueue (*BSD and Darwin only, `IO_POLL_MECHANISM=1`)
- /dev/poll (Sun Solaris only, `IO_POLL_MECHANISM=2`)
- epoll (Linux only, `IO_POLL_MECHANISM=4`)
- poll (`IO_POLL_MECHANISM=8`)
- select (`IO_POLL_MECHANISM=16`)

Environment variables can be set in the configuration file at top-level:

```
setenv IO_POLL_MECHANISM = 4
```

2.4 Configuration Syntax

A single configuration file is sufficient for configuring both **spawnd** and **tcprelay**. The basic format for this file is:

```
id = spawnd {
    # spawnd configuration directives
}

id = tcprelay {
    # tcprelay configuration directives
}
```

For example, the `spawnd` section could look similar to:

```
    listen = { port = 80}
    spawn = { exec /usr/local/libexec/tcprelay }
```

This tells spawnd to accept connections on the port given, and feed them to a **tcprelay** process. Please see the **spawnd** documentation for more configuration details.

tcprelay has its own set of configuration directives:

- `local address` = *addr*

  Specifies the local address used for outgoing connections.

- `rebalance` = *n*

  Re-balances peers after *n* requests. May be used to reactivate dead peers. Use with care. Default: unset.

- `remote = { ... }`

  The `remote` sections tell **tcprelay** where to relay connections to. Valid configuration directives inside the curly brackets are:

  - `address` = *IPAddress*
  - `port` = *TCPPort*
  - `protocol` = (`TCP` | `SCTP`)
  - `weight` = *Weight*

  Both the `address` and `port` directives are mandatory. The load balancing factor *weight* is optional and defaults to 1. Its value should somehow correspond to the load a destination can handle.

- `retire` = *count*

  If set, the daemon will terminate after processing *count* sessions, what may be useful to remedy the effects of memory leaks. By default, this is not set.

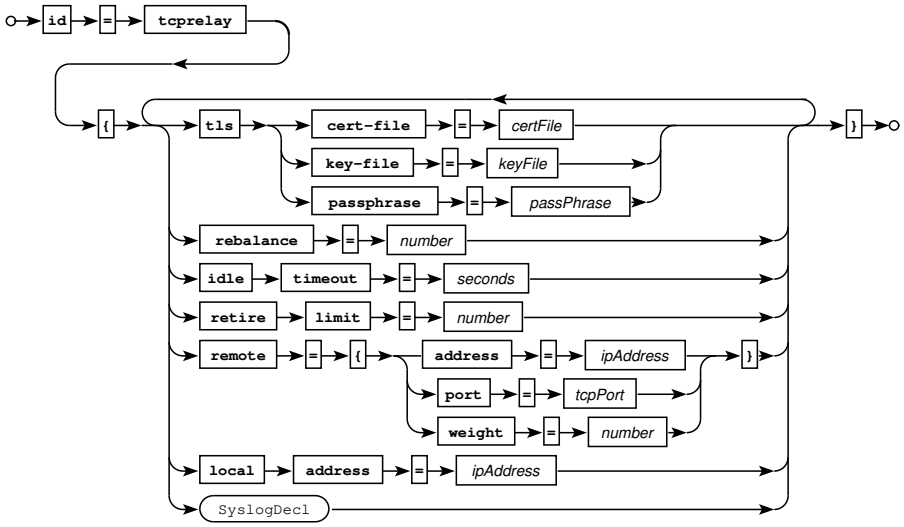
- `syslog` ((`ident` = Ident) | (`severity` = Level) | (`facility` = Facility))

  Selects syslog *ident*, *severity* and *facility*. Defaults to:

```
syslog ident = program-name
syslog facility = UUCP
syslog severity = INFO
```

- `idle timeout` = *Seconds*
  Set session timeout (default: `0`).
- `tls cert-file` = *CertFile*
  `tls key-file` = *KeyFile*
  `tls passphrase` = *PassPhrase*
  If compiled with TLS/SSL, *PassPhrase*, *CertFile* and *KeyFile* may be specified using this option.
  KeyFile may be omitted, it defaults to *CertFile*.

2.4.1 Railroad Diagram



Railroad diagram: TcprelayConfig

2.5 Sample configuration

```
#!/usr/local/sbin/spawnd
id = spawnd {
    listen = { port = 2222 }
    listen = { address = ::0 port = 2222 }
    listen = { ::0 port = 2224 }
    listen = { port = 2225 tls = yes }
    spawn {
        users max = 4000
        users min = 10
        servers min = 1
        servers max = 20
    }
}

id = tcprelay {
  remote = { address = 169.254.1.2 port = 22 }
  ssl cert = /some/where/sample.pem
  ssl passphrase =  12345
}
```

3 Bugs

- TLS re-negotiation is currently untested and may or may not work.

4 Copyrights and Acknowledgements

Please see the source for copyright and licensing information of individual files.

- **The following applies if the software was compiled with TLS support:**

  This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

  This product includes cryptographic software written by Eric Young (eay@cryptsoft.com").

- **Portions of the parsing code are taken from Cisco's tac_plus developers kit which is distributed under the following license:**

  Copyright (c) 1995-1998 by Cisco systems, Inc.

  Permission to use, copy, modify, and distribute this software for any purpose and without fee is hereby granted, provided that this copyright and permission notice appear on all copies of the software and supporting documentation, the name of Cisco Systems, Inc. not be used in advertising or publicity pertaining to distribution of the program without specific prior permission, and notice be given in supporting documentation that modification, copying and distribution is by permission of Cisco Systems, Inc.

  Cisco Systems, Inc. makes no representations about the suitability of this software for any purpose. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

- **The code written by Marc Huber is distributed under the following license:**

  Copyright (C) 1999-2022 Marc Huber (Marc.Huber@web.de). All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

     This product includes software developed by Marc Huber (Marc.Huber@web.de).

     Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

  THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ITS AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.