NAME

tcpdrop - drop TCP connections

SYNOPSIS

tcpdrop *local-address local-port foreign-address foreign-port*
tcpdrop [**-l**] **-a**
tcpdrop [**-l**] **-C** *cc-algo* [**-S** *stack*] [**-s** *state*]
tcpdrop [**-l**] [**-C** *cc-algo*] **-S** *stack* [**-s** *state*]
tcpdrop [**-l**] [**-C** *cc-algo*] [**-S** *stack*] **-s** *state*

DESCRIPTION

The **tcpdrop** command may be used to drop TCP connections from the command line.

If **-a** is specified then **tcpdrop** will attempt to drop all TCP connections.

If **-C** *cc-algo* is specified then **tcpdrop** will attempt to drop all connections using the TCP congestion control algorithm *cc-algo*.

If **-S** *stack* is specified then **tcpdrop** will attempt to drop all connections using the TCP stack *stack*.

If **-s** *state* is specified then **tcpdrop** will attempt to drop all TCP connections being in the state *state*. *state* is one of SYN_SENT, SYN_RCVD, ESTABLISHED, CLOSE_WAIT, FIN_WAIT_1, CLOSING, LAST_ACK, FIN_WAIT_2, or TIME_WAIT.

If multiple of **-C** *cc-algo*, **-S** *stack*, and **-s** *state* are specified, **tcpdrop** will attempt to drop all TCP connections using the congestion control algorithm *cc-algo*, being in the state *state*, and using the TCP stack *stack*, if specified. Since TCP connections in the TIME_WAIT state are not tied to any TCP stack, using the option **-s** TIME_WAIT in combination with the **-S** *stack* option results in **tcpdrop** not dropping any TCP connection.

The **-l** flag may be given in addition to the **-a**, **-C**, **-S**, or **-s** options to list the tcpdrop invocation to drop all corresponding TCP connections one at a time.

If none of the **-a**, **-C**, **-S**, or **-s** options are specified then only the connection between the given local address *local-address*, port *local-port*, and the foreign address *foreign-address*, port *foreign-port*, will be dropped.

Addresses and ports may be specified by name or numeric value. Both IPv4 and IPv6 address formats are supported.

The addresses and ports may be separated by periods or colons instead of spaces.

EXIT STATUS

The **tcpdrop** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

If a connection to httpd(8) is causing congestion on a network link, one can drop the TCP session in charge:

```
# sockstat -c | grep httpd
www      httpd      16525 3  tcp4 \
                    192.168.5.41:80      192.168.5.1:26747
```

The following command will drop the connection:

```
# tcpdrop 192.168.5.41 80 192.168.5.1 26747
```

The following command will drop all connections but those to or from port 22, the port used by sshd(8):

```
# tcpdrop -l -a | grep -vw 22 | sh
```

To drop all TCP connections using the new-reno congestion control algorithm use:

```
# tcpdrop -C new-reno
```

The following command will drop all connections using the TCP stack rack:

```
# tcpdrop -S rack
```

To drop all TCP connections in the LAST_ACK state use:

```
# tcpdrop -s LAST_ACK
```

To drop all TCP connections using the congestion control algorithm new-reno and the TCP stack rack and being in the LAST_ACK state use:

```
# tcpdrop -C new-reno -S rack -s LAST_ACK
```

SEE ALSO

netstat(1), sockstat(1), tcp(4), tcp_functions(9)

AUTHORS

Markus Friedl <*markus@openbsd.org*>
Juli Mallett <*jmallett@FreeBSD.org*>