Rocky Enterprise Linux 9.2 Manual Pages on command 'tc-nat.8'

$ man tc-nat.8

NAT action in tc(8) Linux NAT action in tc(8)

NAME

nat - stateless native address translation action

SYNOPSIS

tc ... action nat DIRECTION OLD NEW

DIRECTION := { ingress | egress }

OLD := IPV4_ADDR_SPEC

NEW := IPV4_ADDR_SPEC

IPV4_ADDR_SPEC := { default | any | all | in_addr[/{prefix|netmask}]

DESCRIPTION

The nat action allows one to perform NAT without the overhead of con?

ntrack, which is desirable if the number of flows or addresses to per?
form NAT on is large. This action is best used in combination with the
u32 filter to allow for efficient lookups of a large number of state?
less NAT rules in constant time.

OPTIONS

ingress

Translate destination addresses, i.e. perform DNAT.

egress Translate source addresses, i.e. perform SNAT.

OLD Specifies addresses which should be translated.

NEW Specifies addresses which OLD should be translated into.

NOTES

The accepted address format in OLD and NEW is quite flexible. It may
either consist of one of the keywords default, any or all, representing
the all-zero IP address or a combination of IP address and netmask or
prefix length separated by a slash (/) sign. In any case, the mask (or
prefix length) value of OLD is used for NEW as well so that a one-to-
one mapping of addresses is assured.

Address translation is done using a combination of binary operations.
First, the original (source or destination) address is matched against
the value of OLD. If the original address fits, the new address is
created by taking the leading bits from NEW (defined by the netmask of
OLD) and taking the remaining bits from the original address.

There is rudimental support for upper layer protocols, namely TCP, UDP
and ICMP. While for the first two only checksum recalculation is per?
formed, the action also takes care of embedded IP headers in ICMP pack?
ets by translating the respective address therein, too.

SEE ALSO

tc(8)