Rocky Enterprise Linux 9.2 Manual Pages on command 'slirp4netns.1'

$ man slirp4netns.1

```
SLIRP4NETNS(1)                    User Commands                    SLIRP4NETNS(1)

NAME

       slirp4netns - User-mode networking for unprivileged network namespaces

SYNOPSIS

       slirp4netns [OPTION]... PID|PATH TAPNAME

DESCRIPTION

       slirp4netns provides user-mode networking ("slirp") for network namespaces.

       Unlike veth(4), slirp4netns does not require the root privileges on the host.

       Default configuration:

           ? MTU:               1500

           ? CIDR:              10.0.2.0/24

           ? Gateway/Host:      10.0.2.2    (network address + 2)

           ? DNS:               10.0.2.3    (network address + 3)

           ? IPv6 CIDR:         fd00::/64

           ? IPv6 Gateway/Host: fd00::2

           ? IPv6 DNS:          fd00::3

OPTIONS

       -c,  --configure bring up the TAP interface. IP will be set to 10.0.2.100 (network address

       + 100) by default. IPv6 will be set to a random address.  Starting with v0.4.0, the  loop?

       back interface (lo) is brought up as well.

       -e,  --exit-fd=FD  specify  the FD for terminating slirp4netns.  When the FD is specified,

       slirp4netns exits when a poll(2) event happens on the FD.

       -r, --ready-fd=FD specify the FD to write to when the initialization steps  are  finished.
```

When the FD is specified, slirp4netns writes "1" to the FD and close the FD. Prior to

v0.4.0, the FD was written after the network configuration (-c) but before the API socket

configuration (-a).

-m, --mtu=MTU (since v0.2.0) specify MTU (max=65521).

-6, --enable-ipv6 (since v0.2.0, EXPERIMENTAL) enable IPv6

-a, --api-socket (since v0.3.0) API socket path

--cidr (since v0.3.0) specify CIDR, e.g. 10.0.2.0/24

--disable-host-loopback (since v0.3.0) prohibit connecting to 127.0.0.1:* on the host

namespace

--netns-type=TYPE (since v0.4.0) specify network namespace type ([path|pid], default=pid)

--userns-path=PATH (since v0.4.0) specify user namespace path

--enable-sandbox (since v0.4.0) enter the user namespace and create a new mount namespace

where only /etc and /run are mounted from the host.

Requires /etc/resolv.conf not to be a symlink to a file outside /etc and /run.

When running as the root, the process does not enter the user namespace but all the capa?

bilities except CAP_NET_BIND_SERVICE are dropped.

--enable-seccomp (since v0.4.0, EXPERIMENTAL) enable seccomp(2) to limit syscalls.   Typi?

cally used in conjunction with --enable-sandbox.

-h, --help (since v0.2.0) show help and exit

-v, --version (since v0.2.0) show version and exit

EXAMPLE

Terminal 1: Create user/network/mount namespaces

```
$ unshare --user --map-root-user --net --mount
unshared$ echo $$ > /tmp/pid
```

Terminal 2: Start slirp4netns

```
$ slirp4netns --configure --mtu=65520 $(cat /tmp/pid) tap0
starting slirp, MTU=65520
```

Terminal 1: Make sure tap0 is configured and connected to the Internet

```
unshared$ ip a
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: tap0: <BROADCAST,UP,LOWER_UP> mtu 65520 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether c2:28:0c:0e:29:06 brd ff:ff:ff:ff:ff:ff
```

```
           inet 10.0.2.100/24 brd 10.0.2.255 scope global tap0
              valid_lft forever preferred_lft forever
           inet6 fe80::c028:cff:fe0e:2906/64 scope link
              valid_lft forever preferred_lft forever
       unshared$ echo "nameserver 10.0.2.3" > /tmp/resolv.conf
       unshared$ mount --bind /tmp/resolv.conf /etc/resolv.conf
       unshared$ curl https://example.com
```

Bind-mounting /etc/resolv.conf is only needed when /etc/resolv.conf on the host refers to loopback addresses (127.0.0.X, typically because of dnsmasq(8) or systemd-resolved.ser? vice(8)) that cannot be accessed from the namespace.

If your /etc/resolv.conf on the host is managed by networkmanager(8) or systemd-re? solved.service(8), you might need to mount a new filesystem on /etc instead, so as to pre? vent the new /etc/resolv.conf from being unmounted unexpectedly when /etc/resolv.conf on the host is regenerated.

```
       unshared$ mkdir /tmp/a /tmp/b
       unshared$ mount --rbind /etc /tmp/a
       unshared$ mount --rbind /tmp/b /etc
       unshared$ mkdir /etc/.ro
       unshared$ mount --move /tmp/a /etc/.ro
       unshared$ cd /etc
       unshared$ for f in .ro/*; do ln -s $f $(basename $f); done
       unshared$ rm resolv.conf
       unshared$ echo "nameserver 10.0.2.3" > /tmp/resolv.conf
       unshared$ curl https://example.com
```

ROUTING PING PACKETS

To route ping packets, you need to set up net.ipv4.ping_group_range properly as the root.

e.g.

```
       $ sudo sh -c "echo 0   2147483647  > /proc/sys/net/ipv4/ping_group_range"
```

FILTERING CONNECTIONS

By default, ports listening on INADDR_LOOPBACK (127.0.0.1) on the host are accessible from the child namespace via the gateway (default: 10.0.2.2). --disable-host-loopback can be used to prohibit connecting to INADDR_LOOPBACK on the host.

However, a host loopback address might be still accessible via the built-in DNS (default:

10.0.2.3) if /etc/resolv.conf on the host refers to a loopback address. You may want to set up iptables for limiting access to the built-in DNS in such a case.

```
unshared$ iptables -A OUTPUT -d 10.0.2.3 -p udp --dport 53 -j ACCEPT
unshared$ iptables -A OUTPUT -d 10.0.2.3 -j DROP
```

API SOCKET

slirp4netns can provide QMP-like API server over an UNIX socket file:

```
$ slirp4netns --api-socket /tmp/slirp4netns.sock ...
```

add_hostfwd: Expose a port (IPv4 only)

```
$ json='{"execute": "add_hostfwd", "arguments": {"proto": "tcp", "host_addr": "0.0.0.0", "host_port": 8080, "guest_addr": "10.0.2.100", "guest_port": 80}}'
$ echo -n $json | nc -U /tmp/slirp4netns.sock
{ "return": {"id": 42}}
```

If host_addr is not specified, then it defaults to "0.0.0.0".

If guest_addr is not specified, then it will be set to the default address that corre? sponds to --configure.

list_hostfwd: List exposed ports

```
$ json='{"execute": "list_hostfwd"}'
$ echo -n $json | nc -U /tmp/slirp4netns.sock
{ "return": {"entries": [{"id": 42, "proto": "tcp", "host_addr": "0.0.0.0", "host_port": 8080, "guest_addr": "10.0.2.100", "guest_port": 80}]}}
```

remove_hostfwd: Remove an exposed port

```
$ json='{"execute": "remove_hostfwd", "arguments": {"id": 42}}'
$ echo -n $json | nc -U /tmp/slirp4netns.sock
{ "return": {}}
```

Remarks:

? Client needs to shutdown(2) the socket with SHUT_WR after sending every request.
  i.e. No support for keep-alive and timeout.

? slirp4netns "stops the world" during processing API requests.

? A request must be less than 4096 bytes.

? JSON responses may contain error instead of return.

DEFINED NAMESPACE PATHS

A user can define a network namespace path as opposed to the default process ID:

```
$ slirp4netns --netns-type=path ... /path/to/netns tap0
```

Currently, the netns-type=TYPE argument supports path or pid args with the default being pid.

Additionally, a --userns-path=PATH argument can be included to override any user namespace path defaults

```
$ slirp4netns --netns-type=path --userns-path=/path/to/userns /path/to/netns tap0
```

BUGS

Kernel 4.20 bumped up the default value of /proc/sys/net/ipv4/tcp_rmem from 87380 to 131072. This is known to slow down slirp4netns port forwarding: https://github.com/root?less-containers/slirp4netns/issues/128.

As a workaround, you can adjust the value of /proc/sys/net/ipv4/tcp_rmem inside the name?space. No real root privilege is needed to modify the file since kernel 4.15.

```
unshared$ c=$(cat /proc/sys/net/ipv4/tcp_rmem); echo $c | sed -e s/131072/87380/g > /proc/sys/net/ipv4/tcp_rmem
```

SEE ALSO

network_namespaces(7), user_namespaces(7), veth(4)

AVAILABILITY

The slirp4netns command is available from https://github.com/rootless-contain?ers/slirp4netns under GNU GENERAL PUBLIC LICENSE Version 2 (or later).