

Name

bpftool-net – tool for inspection of networking related bpf prog attachments

SYNOPSIS

bpftool [*OPTIONS*] **net** *COMMAND*

OPTIONS := { { **-j** | **--json** } [{ **-p** | **--pretty** }] | { **-d** | **--debug** } }

COMMANDS := { **show** | **list** | **attach** | **detach** | **help** }

NET COMMANDS

bpftool net { **show** | **list** } [**dev** *NAME*]

bpftool net attach *ATTACH_TYPE* *PROG* **dev** *NAME* [**overwrite** | **prepend**]

bpftool net detach *ATTACH_TYPE* **dev** *NAME*

bpftool net help

PROG := { **id** *PROG_ID* | **pinned** *FILE* | **tag** *PROG_TAG* | **name** *PROG_NAME* }

ATTACH_TYPE := { **xdp** | **xdpgeneric** | **xdpdrv** | **xdpoffload** | **tcx_ingress** | **tcx_egress** }

DESCRIPTION

bpftool net { **show** | **list** } [**dev** *NAME*]

List bpf program attachments in the kernel networking subsystem.

Currently, device driver xdp attachments, tcx, netkit and old-style tc classifier/action attachments, flow_dissector as well as netfilter attachments are implemented, i.e., for program types

BPF_PROG_TYPE_XDP, **BPF_PROG_TYPE_SCHED_CLS**,
BPF_PROG_TYPE_SCHED_ACT, **BPF_PROG_TYPE_FLOW_DISSECTOR**,
BPF_PROG_TYPE_NETFILTER.

For programs attached to a particular cgroup, e.g., **BPF_PROG_TYPE_CGROUP_SKB**, **BPF_PROG_TYPE_CGROUP_SOCK**, **BPF_PROG_TYPE_SOCK_OPS** and **BPF_PROG_TYPE_CGROUP_SOCK_ADDR**, users can use **bpftool cgroup** to dump cgroup attachments. For sk_{filter, skb, msg, reuseport} and lwt/seg6 bpf programs, users should consult other tools, e.g., iproute2.

The current output will start with all xdp program attachments, followed by all tcx, netkit, then tc class/qdisc bpf program attachments, then flow_dissector and finally netfilter programs. Both xdp programs and tcx/netkit/tc programs are ordered based on ifindex number. If multiple bpf programs attached to the same networking device through **tc**, the order will be first all bpf programs attached to tcx, netkit, then tc classes, then all bpf programs attached to non clsact qdiscs, and finally all bpf programs attached to root and clsact qdisc.

bpftool net attach *ATTACH_TYPE* *PROG* **dev** *NAME* [**overwrite** | **prepend**]

Attach bpf program *PROG* to network interface *NAME* with type specified by *ATTACH_TYPE*.

ATTACH_TYPE can be of: **xdp** – try native XDP and fallback to generic XDP if NIC driver does not support it; **xdpgeneric** – Generic XDP. runs at generic XDP hook when packet already enters receive path as skb; **xdpdrv** – Native XDP. runs earliest point in driver's receive path; **xdpoffload** – Offload XDP. runs directly on NIC on each packet reception; **tcx_ingress** – Ingress TCX. runs on ingress net traffic; **tcx_egress** – Egress TCX. runs on egress net traffic;

For XDP-related attach types (**xdp**, **xdpgeneric**, **xdpdrv**, **xdpoffload**), the **overwrite** option can be used to replace a previously attached bpf program.

For **tcx_ingress** and **tcx_egress** attach types, the **prepend** option can be used to attach the program at the beginning of the chain instead of at the end.

bpftool net detach *ATTACH_TYPE* **dev** *NAME*

Detach bpf program attached to network interface *NAME* with type specified by *ATTACH_TYPE*. To detach bpf program, same *ATTACH_TYPE* previously used for attach must be specified.

bpftool net help

Print short help message.

OPTIONS**-h, --help**

Print short help message (similar to **bpftool help**).

-V, --version

Print bpftool's version number (similar to **bpftool version**), the number of the libbpf version in use, and optional features that were included when bpftool was compiled. Optional features include linking against LLVM or libbfd to provide the disassembler for JIT-*ted* programs (**bpftool prog dump jited**) and usage of BPF skeletons (some features like **bpftool prog profile** or showing pids associated to BPF objects may rely on it).

-j, --json

Generate JSON output. For commands that cannot produce JSON, this option has no effect.

-p, --pretty

Generate human-readable JSON output. Implies **-j**.

-d, --debug

Print all logs available, even debug-level information. This includes logs from libbpf as well as from the verifier, when attempting to load programs.

EXAMPLES**# bpftool net**

```
xdp:
eth0(2) driver id 198

tc:
eth0(2) htb name prefix_matcher.o:[cls_prefix_matcher_htb] id 111727 act []
eth0(2) clsact/ingress fbflow_icmp id 130246 act []
eth0(2) clsact/egress prefix_matcher.o:[cls_prefix_matcher_clsact] id 111726
eth0(2) clsact/egress cls_fg_dscp id 108619 act []
eth0(2) clsact/egress fbflow_egress id 130245
```

bpftool -jp net

```
[{
  "xdp": [{
    "devname": "eth0",
    "ifindex": 2,
    "mode": "driver",
    "id": 198
  }],
  "tc": [{
    "devname": "eth0",
    "ifindex": 2,
    "kind": "htb",
    "name": "prefix_matcher.o:[cls_prefix_matcher_htb]",
    "id": 111727,
```

```

        "act": []
      }, {
        "devname": "eth0",
        "ifindex": 2,
        "kind": "clsact/ingress",
        "name": "fbflow_icmp",
        "id": 130246,
        "act": []
      }, {
        "devname": "eth0",
        "ifindex": 2,
        "kind": "clsact/egress",
        "name": "prefix_matcher.o:[cls_prefix_matcher_clsact]",
        "id": 111726,
      }, {
        "devname": "eth0",
        "ifindex": 2,
        "kind": "clsact/egress",
        "name": "cls_fg_dscp",
        "id": 108619,
        "act": []
      }, {
        "devname": "eth0",
        "ifindex": 2,
        "kind": "clsact/egress",
        "name": "fbflow_egress",
        "id": 130245,
      }
    ]
  }
]

```

```

# bpftool net attach xdpdrv id 16 dev enp6s0np0
# bpftool net

```

```

xdp:
enp6s0np0(4) driver id 16

```

```

# bpftool net attach xdpdrv id 16 dev enp6s0np0
# bpftool net attach xdpdrv id 20 dev enp6s0np0 overwrite
# bpftool net

```

```

xdp:
enp6s0np0(4) driver id 20

```

```

# bpftool net attach xdpdrv id 16 dev enp6s0np0
# bpftool net detach xdpdrv dev enp6s0np0
# bpftool net

```

```

xdp:

```

```
# bpftool net attach tcx_ingress name tc_prog dev lo
# bpftool net
```

```
tc:
lo(1) tcx/ingress tc_prog prog_id 29
```

```
# bpftool net attach tcx_ingress name tc_prog2 dev lo prepend
# bpftool net
```

```
tc:
lo(1) tcx/ingress tc_prog2 prog_id 30
lo(1) tcx/ingress tc_prog prog_id 29
```

```
# bpftool net attach tcx_ingress name tc_prog dev lo
# bpftool net detach tcx_ingress dev lo
# bpftool net
```

```
tc:
```

SEE ALSO

bpf(2), **bpf-helpers(7)**, **bpftool(8)**, **bpftool-btf(8)**, **bpftool-cgroup(8)**, **bpftool-feature(8)**, **bpftool-gen(8)**, **bpftool-iter(8)**, **bpftool-link(8)**, **bpftool-map(8)**, **bpftool-perf(8)**, **bpftool-prog(8)**, **bpftool-struct_ops(8)**, **bpftool-token(8)**