Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'bpftool-feature.8' command

$ man bpftool-feature.8

BPFTOOL-FEATURE(8) BPFTOOL-FEATURE(8)

NAME

bpftool-feature - tool for inspection of eBPF-related parameters for Linux kernel or net device

SYNOPSIS

bpftool [OPTIONS] feature COMMAND

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

COMMANDS := { probe | help }

FEATURE COMMANDS

bpftool feature probe [COMPONENT] [full] [unprivileged] [macros [prefix PREFIX]]

bpftool feature list_builtins GROUP

bpftool feature help

COMPONENT := { kernel | dev NAME }

GROUP := { prog_types | map_types | attach_types | link_types | helpers }

DESCRIPTION

bpftool feature probe [kernel] [full] [macros [prefix PREFIX]]

Probe the running kernel and dump a number of eBPF-related parameters, such as availability of the bpf() system call, JIT status, eBPF program types availability, eBPF helper functions availability, and more.

By default, bpftool does not run probes for bpf_probe_write_user() and bpf_trace_printk() helpers which

print warnings to kernel logs. To enable them and run all probes, the full keyword should be used.

If the macros keyword (but not the -j option) is passed, a subset of the output is dumped as a list of #define macros that are ready to be included in a C header file, for exam? ple. If, additionally, prefix is used to define a PREFIX, the provided string will be used as a prefix to the names of the macros: this can be used to avoid conflicts on macro names when including the output of this command as a header file.

Keyword kernel can be omitted. If no probe target is speci? fied, probing the kernel is the default behaviour.

When the unprivileged keyword is used, bpftool will dump only the features available to a user who does not have the CAP_SYS_ADMIN capability set. The features available in that case usually represent a small subset of the parameters sup? ported by the system. Unprivileged users MUST use the unpriv? ileged keyword: This is to avoid misdetection if bpftool is inadvertently run as non-root, for example. This keyword is unavailable if bpftool was compiled without libcap.

bpftool feature probe dev NAME [full] [macros [prefix PREFIX]]

Probe network device for supported eBPF features and dump re? sults to the console.

The keywords full, macros and prefix have the same role as when probing the kernel.

bpftool feature list_builtins GROUP

List items known to bpftool. These can be BPF program types (prog_types), BPF map types (map_types), attach types (at? tach_types), link types (link_types), or BPF helper functions (helpers). The command does not probe the system, but simply lists the elements that bpftool knows from compilation time, as provided from libbpf (for all object types) or from the BPF UAPI header (list of helpers). This can be used in scripts to iterate over BPF types or helpers.

bpftool feature 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 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.

-l, --legacy

Use legacy libbpf mode which has more relaxed BPF program requirements. By default, bpftool has more strict requirements about section names, changes pinning logic and doesn't support some of the older non-BTF map declarations.

See https://github.com/libbpf/libbpf/wiki/Libbpf:-the-road-to-v1.0 for details.

SEE ALSO

bpf(2), bpf-helpers(7), bpftool(8), bpftool-btf(8),

bpftool-cgroup(8), bpftool-gen(8), bpftool-iter(8), bpftool-link(8), bpftool-map(8), bpftool-net(8), bpftool-perf(8), bpftool-prog(8), bpftool-struct_ops(8)

BPFTOOL-FEATURE(8)