

eBPF



eBPF is a pure Go library that provides utilities for loading, compiling, and debugging eBPF programs. It has minimal external dependencies and is intended to be used in long running processes.

The library is maintained by [Cloudflare](#) and [Cilium](#).

See [ebpf.io](#) for other projects from the eBPF ecosystem.

Getting Started

A small collection of Go and eBPF programs that serve as examples for building your own tools can be found under [examples/](#).

Contributions are highly encouraged, as they highlight certain use cases of eBPF and the library, and help shape the future of the project.

Getting Help

Please [join](#) the [#ebpf-go](#) channel on Slack if you have questions regarding the library.

Packages

This library includes the following packages:

- [asm](#) contains a basic assembler, allowing you to write eBPF assembly instructions directly within your Go code. (You don't need to use this if you prefer to write your eBPF program in C.)
- [cmd/bpf2go](#) allows compiling and embedding eBPF programs written in C within Go code. As well as compiling the C code, it auto-generates Go code for loading and manipulating the eBPF program and map objects.
- [link](#) allows attaching eBPF to various hooks
- [perf](#) allows reading from a `PERF_EVENT_ARRAY`
- [ringbuf](#) allows reading from a `BPF_MAP_TYPE_RINGBUF` map

Requirements

- A version of Go that is [supported by upstream](#)
- Linux >= 4.9. CI is run against LTS releases.

Regenerating Testdata

Run `make` in the root of this repository to rebuild testdata in all subpackages. This requires Docker, as it relies on a standardized build environment to keep the build output stable.

The toolchain image build files are kept in [testdata/docker/](#).

License

MIT

eBPF Gopher

The eBPF honeygopher is based on the Go gopher designed by Renee French.