eBPF maps

'maps' is a generic storage of different types for sharing data between kernel and userspace.

The maps are accessed from user space via BPF syscall, which has commands:

- create a map with given type and attributes `map_fd = bpf(BPF_MAP_CREATE, union bpf_attr *attr, u32 size)` using attr->map_type, attr->key_size, attr->value_size, attr->max_entries returns process-local file descriptor or negative error
- lookup key in a given map `err = bpf(BPF_MAP_LOOKUP_ELEM, union bpf_attr *attr, u32 size)` using attr->map_fd, attr->key, attr->value returns zero and stores found elem into value or negative error
- create or update key/value pair in a given map `err = bpf(BPF_MAP_UPDATE_ELEM, union bpf_attr *attr, u32 size)` using attr->map_fd, attr->key, attr->value returns zero or negative error
- find and delete element by key in a given map `err = bpf(BPF_MAP_DELETE_ELEM, union bpf_attr *attr, u32 size)` using attr->map_fd, attr->key
- to delete map: close(fd) Exiting process will delete maps automatically

userspace programs use this syscall to create/access maps that eBPF programs are concurrently updating.

maps can have different types: hash, array, bloom filter, radix-tree, etc.

The map is defined by:

- type
- max number of elements
- key size in bytes
- value size in bytes

Map Types

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\bpf\(linux-master)(Documentation)(bpf)maps.rst, line 48)

Unknown directive type "toctree".

```
.. toctree::
   :maxdepth: 1
   :glob:

   map_*
```