bcc Hacking Stuff

Map in Map Issue

https://github.com/iovisor/bcc/issues/1318

The Basics

- we need a file descriptor to an inner template map
  - bcc already supports BPF_PINNED_MAP
  - maybe this could be useful?
- possibly should have a way to *free* the template map after it is created

Main Problems

- best way to handle creation of template map?
  - probably want to reuse some existing macro here (maybe BPF_TABLE or BPF_PINNED_MAP)
- how to lookup file descriptor by name?
  - palmtenor said this was easy but then he dropped off the face of the earth without explaining
- how to fix potential rewriter problems?
  - some of the diffs below might be helpful
  - `src/cc/frontends/clang/b_frontend_action.cc`
- palmtenor said the hardest problem was inner map lookup after the fact
- what is the best workflow for testing here?

Creation Flow

1. define inner map template
2. lookup inner map fd by name
3. define outer map using attributes of inner map
4. maybe free inner map fd

Important Files

- `src/python/bcc/table.py`
  - python objects to refer to BPF tables
  - already defines ID for hash_of_maps
  - still need to write the class
- `src/cc/export/helpers.h`
  - define helper macros for BPF programs
  - need to add macros for HASH_OF_MAPS here
- `src/cc/frontends/clang/b_frontend_action.cc`
  - rewriter stuff (see palmtenor's issue above)
- maybe more

Some Diffs That Might Be Useful

- add btf support for maps
- add support for devmap (this seems to be cc API exclusive but could still be useful)
- add lru hash and lru per cpu hash
- add devmap and cpumap (**this is probably the most helpful so far**)

Meeting

1. pin an inner map of file descriptors

2. (maybe somehow call create_map_in_map directly?)

- pin a map
- access bytecode from /sys/fs/bpf
- load it into a data structure
- get assembly code from compiler
- edit the assembly code
- load it in
- send off to the kernel

Useful Links

- https://blogs.oracle.com/linux/notes-on-bpf-3