BTF function resolve

jiri olsa / isovalent

PROBLEM

- **resolve function by NAME to ADDRESS**
- **static functions with same name**
- **trampoline attachment**

BTF

```
# bpftool btf dump file /sys/kernel/btf/vmlinux  | grep "FUNC 'type_show"
[15268] FUNC 'type_show' type_id=1692 linkage=static
[55008] FUNC 'type_show' type_id=5390 linkage=static
[58659] FUNC 'type_show' type_id=58639 linkage=static
…
```

libbpf

```
__s32 btf__find_by_name_kind_own(const struct btf *btf,
                                 const char *type_name, __u32 kind);
```

kernel

```
bpf_check
  check_attach_btf_id
    bpf_check_attach_target

        t = btf_type_by_id(btf, btf_id);
        tname = btf_name_by_offset(btf, t->name_off);
        addr = kallsyms_lookup_name(tname);
```

kallsyms

```
# cat /proc/kallsyms | egrep 't type_show$'
ffffffffb4034860 t type_show
ffffffffb4121e50 t type_show
ffffffffb416ea40 t type_show
ffffffffb42abe50 t type_show
ffffffffb47ca240 t type_show
ffffffffb47f1a80 t type_show
ffffffffb4827f20 t type_show
ffffffffb485fe70 t type_show
ffffffffb48645f0 t type_show
ffffffffb4892af0 t type_show
ffffffffb49343f0 t type_show
ffffffffb49c5060 t type_show
ffffffffb49ea500 t type_show
ffffffffb4a369d0 t type_show
ffffffffb4a7d190 t type_show
ffffffffb4acef00 t type_show
ffffffffb4ad04a0 t type_show
ffffffffb4b0aa60 t type_show
ffffffffb4b73650 t type_show
```

FIX THE BLEEDING

- **pahole fix by Alan Maguire**
- **ensure functions with same name have same prototype**

PROPOSAL

- use PATH / FUNCTION to identify function
- store PATH in BTF using DECL_TAG
- libbpf path/function search
- kallsyms path/function search

PROPOSAL

- **BTF**

  easy pahole change

  ~1M size increase

```
BTF

[62802] FUNC 'ksys_read' type_id=62801 linkage=static
[62803] DECL_TAG 'path:fs/read_write.c' type_id=62802 component_idx=-1
```

PROPOSAL

- **libbpf**

  lookup of DECL_TAG with func BTF ID

libbpf

```
__s32 btf__find_by_name_func(const struct btf *btf,
                             const char *func, const char *path);
```

PROPOSAL

- **kallsyms change**

```
kernel

bpf_check
  check_attach_btf_id
    bpf_check_attach_target

        resolve_func_path(btf_id, &func, &path);

        addr = kallsyms_lookup_path(func, path);
```

```
  NM      .tmp_vmlinux.kallsyms1.syms
  KSYMS   .tmp_vmlinux.kallsyms1.S
  AS      .tmp_vmlinux.kallsyms1.S
```

```
kallsyms_offsets:
        .long   0
        .long   0
        .long   0x1000
        .long   0x2000
        .long   0x6000

kallsyms_names:
        .byte 0x0c, 0x41, 0xaf, 0x78, 0x65, 0xea, ...
        .byte 0x09, 0x41, 0xff, 0x70, 0xf5, 0xe5, ...
        .byte 0x08, 0x41, 0x63, 0x9e, 0x5f, 0x2a, ...
```

```
  NM      .tmp_vmlinux.kallsyms1.syms
  KSYMS   .tmp_vmlinux.kallsyms1.S
  AS      .tmp_vmlinux.kallsyms1.S
```

```
kallsyms_offsets:
        .long   0
        .long   0
        .long   0x1000
        .long   0x2000
        .long   0x6000

kallsyms_names:
        .byte idx1, 0x0c, 0x41, 0xaf, 0x78, 0x65, 0xea, ...
        .byte idx2, 0x09, 0x41, 0xff, 0x70, 0xf5, 0xe5, ...
        .byte idx3, 0x08, 0x41, 0x63, 0x9e, 0x5f, 0x2a, ...

kallsyms_paths:
        .byte 'path1'
        .byte 'path2'
        .byte 'path3'
```

PROPOSAL

- **store address directly in BTF?**

thanks, questions..