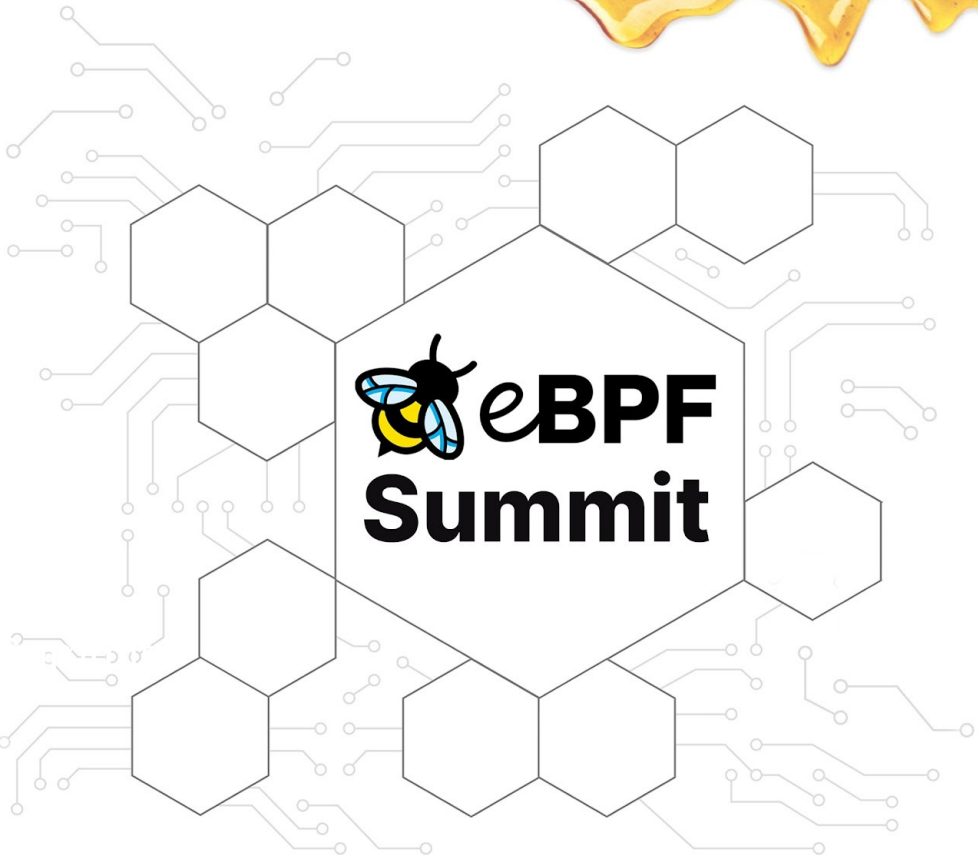
The power of a self-describing kernel



How the BPF Type Format helps applications

Stephen.S.Brennan@oracle.com @brenns10

Alan.Maguire@oracle.com

Kernel Debugging

- KDB, KGDB: built-in to the kernel
- GDB: general purpose debugger
- Crash, Pykdump: kernel specific, built on GDB
- Drgn: independent, scriptable debugger

All except KDB require debug symbols

Self-describing kernel

- Kallsyms: a symbol table
  - CONFIG_KALLSYMS: symbols for functions
  - CONFIG_KALLSYMS_ALL: symbols from all sections
  - Compare to: ELF Symbol Table
- ORC: a stack unwinding information format
  - Alternatively, frame pointers
  - Compare to: DWARF CFI
- BTF: description of C types
  - Struct layouts, membernames, etc
  - Compare to: DWARF .debug_frame

Goal

- Bring these self-describing components together, to allow basic debugging tasks without DWARF
- Enable debuggers (like drgn) to debug a running kernel or core dump without requiring any external resources
- Non-goal: replace all features of DWARF

Roadblocks

- Kallsyms symbol table can’t be found in core dumps
  - This is resolved in Linux 6.0 - location of symbol table is now included in vmcoreinfo note
- BTF currently contains only function & percpu variable types
  - Work-in-progress series for pahole to address this
- Debuggers don’t yet understand BTF or kallsyms
  - Currently reviewing branches for drgn to add support

BTF and libpcap/wiresharp/tcpdump

We can use wireshark, tcpdump to capture network traffic at specific tap points in the kernel.

With eBPF, we can be more flexible – what about capturing traffic at arbitrary function entry points too?

And using BTF, we can identify functions that have an sk_buff argument.

The user just needs to specify a function name as “device”.

BTF and libpcap/wiresharp/tcpdump (2)

For example:

```
$ tcpdump -p -vvv -y ieee802_11 -i fentry/ieee80211_scan_rx
tcpdump: data link type ieee802_11
tcpdump: listening on fentry/ieee80211_scan_rx, link-type
IEEE802_11 (802.11), capture size 262144 bytes
16:54:12.538455 Probe Response (WifiNetwork1) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0
54.0 Mbit] CH: 2, PRIVACY
16:54:12.538477 Probe Response (WifiNetwork2) [1.0* 2.0* 5.5* 11.0* 18.0 24.0 36.0
54.0 Mbit] CH: 2, PRIVACY
```

It is important to specify a linktype. Another example:

```
$ tcpdump -i fentry/ip_output -y raw
tcpdump: data link type raw
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on fentry/ip_output, link-type RAW (Raw IP), capture size 262144 bytes
16:57:40.206053 IP localhost.localdomain.43242 > ec2-63-33-185-197.eu-west-
1.compute.amazonaws.com.https: Flags [P.], seq 1476971937:1476971991, ack 1968800920,
win 501, options [nop,nop,TS val 2735046249 ecr 3135936950], length 54
```

See https://github.com/alan-maguire/libpcap/tree/pcap-ebpf for proof-of-concept.

Summary

Kernel debugging and packet capture are activities we have been doing for a long time.

With BTF availability, we no longer need to rely on expensive external type information for simple debugging.

And other tasks can be simplified (packet capture at arbitrary functions) by having function signature information.

References

Work-in-progress series for pahole

https://lore.kernel.org/bpf/20220826184911.168442-1-stephen.s.brennan@oracle.com/

drgn support for BTF

https://github.com/osandov/drgn/pull/177

libpcap support

https://github.com/alan-maguire/libpcap/tree/pcap-ebpf