coolbpf Documentation

Shuyi Cheng

Feb 27, 2023

CONTENTS

This is the top level of the coolbpf's documentation tree. Coolbpf documentation, like the coolbpf itself, is very much a work in progress; that is especially true as we work to integrate our many scattered documents into a coherent whole. Please note that improvements to the documentation are welcome; join coolbpf at https://gitee.com/anolis/coolbpf.git if you want to help out.

CHAPTER

ONE

COOLBPF API DOCUMENTATION

This is documentation for coolbpf, a userspace library for quickly build and develop eBPF applications.

CHAPTER

TWO

LIGHTWEIGHT COOLBPF DOCUMENTATION

LightWeight CoolBPF eBPF eBPF API eBPF

2.1 LightWeight CoolBPF (LWCB)

2.1.1 LightWeight CoolBPF introduction

lwcb(LightWeight CoolBPF) eBPF(Berkeley Packet Filter) lwcb

2.1.2 LightWeight CoolBPF tutorial

lwcb lwcb

- & `kprobe:tcp_drop` kprobe `tcp_drop`
- lwcb `C` lwcb

`tcphdrbswapiphdrntoptimestrnstcpflagskstack` lwcb API API reference

lwcb `./lwcb tcpdrop.cb`

```
kprobe:tcp_drop {
    th = tcphdr(skb);
    sport = bswap(th->source);
    dport = bswap(th->dest);

    ih = iphdr(skb);
    sip = ntop(bswap(ih->saddr));
    dip = ntop(bswap(ih->daddr));

    state = tcpstate(sk->__sk_common.skc_state);
    print("%s ip: %s:%d -> %s:%d state: %s flags:%s %s\n", timestr(ns()), sip, sport,␣
↪dip, dport, state, tcpflags(((u8 *)th)[13]), kstack());
}
```

2.1.3 LightWeight CoolBPF building guide

1.

- rust `curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh`
- coolbpf `git clone https://gitee.com/anolis/coolbpf.git`
- lwcb `cd coolbpf/lwcb`

2.

`cargo build --release` lwcb `target/release/lwcb`

2.1.4 LightWeight CoolBPF reference

lwcb

1.

`lwcb -t` eBPF

```
lwcb -t 'kprobe:tcp_rcv_established {print("%s :triggerred by tcp_rcv_established\n",␣
↪timestr(ns()));}'
```

lwcb

`lwcb <>` lwcb

2.

lwcb c

- { }
- //
- “string”
- -> .
- `` if { } else { } ``if
- `return`
- [] eBPF map

3.

- k(ret)probe BPF_PROG_TYPE_KPROBE

- tracepoint BPF_PROG_TYPE_TRACEPOINT
- raw tracepointBPF_PROG_TYPE_RAW_TRACEPOINT
- syscall BPF_PROG_TYPE_SYSCALL
- perf eventBPF_PROG_TYPE_PERF_EVENT

4. MAP

eBPF map

- BPF_MAP_TYPE_PERF_EVENT_ARRAY
- BPF_MAP_TYPE_HASH

eBPF map

- BPF_MAP_TYPE_ARRAY
- BPF_MAP_TYPE_PERCPU_HASH
- BPF_MAP_TYPE_PERCPU_ARRAY
- BPF_MAP_TYPE_RINGBUF ringbuffer ringbuffer perf buffer

5.

print

```
print(fmt, args)
```

iphdr

```
iphdr(skb)
```

tcphdr

```
tcphdr(skb)
```

ntop

```
ntop(i32 addr)
```

bswap

```
bswap(u8 | i8 | u16 | i16 | u32 | i32 | u64 | i64)
```

kstack

```
kstack(i32 depth) | kstack()
```

ns

```
ns()
```

pid

```
pid()
```

tcpstate

```
tcpstate(i32 tcpstate)
```

tcpflags

```
tcpstate(i32 tcpflags)
```

timestr

```
timestr(u64 ts)
```

ksym

```
ksym(u64 kernel_address)
```

reg

```
reg(string)
```

6.

linux `#define IPPROTO_TCP 6` lwcb lwcb

- IPPROTO_IP
- IPPROTO_TCP
- IPPROTO_ICMP
- IPPROTO_UDP

`lwcb/src/cmacro.rs`

2.1.5 LightWeight CoolBPF todo features

lwcb

1. **tracepoint**
   - `/sys/kernel/debug/tracing/events/<category>/<name>/format`
   -
   - `TracepointProgram loadattach`
2. `array map hash map lwcb/src/bpf/map/hash.rs`
3. `uprobe`
4. `beginend`
5.
6. `tuple`
7. `for`
8. `btf id btf id btf id`

2.2 LightWeight CoolBPF (LWCB) for Python

2.2.1 pylwcb introduction

pylwcb lwcb python pylwcb lwcb tracing
python

2.2.2 pylwcb tutorial

pylwcb lwcb python eBPF python

1. hello world

`import pylwcb` pylwcb `say_hello` `hello` `from pylwcb modules`

```
import pylwcb
print(pylwcb.say_hello())
```

2. pylwcb

lwcb pylwcb

```
import pylwcb
lwcb_program = """
kprobe:tcp_rcv_established {
    th = tcphdr(skb);
    ih = iphdr(skb);
    print(ntop(bswap(ih->saddr)), ntop(bswap(ih->daddr)), bswap(th->source), bswap(th->
↪dest));
}
"""

lwcb = pylwcb.Pylwcb(lwcb_program)
lwcb.attach()

events = lwcb.read_events()
for event in events:
    print(event)
```

- **`lwcb_program` lwcb `tcp_rcv_established` tcp**
- `lwcb = pylwcb.Pylwcb(lwcb_program)` Pylwcb
- `lwcb.attach()` eBPF
- `lwcb.read_events()` eBPF

2.2.3 pylwcb developing guide

pylwcb pylwcb pylwcb

1.

pylwcb

1. python `python -m venv .env`
2. `source .env/bin/activate`
3. **pylwcb python**
   - `pip install tomli`
   - `pip install setuptools_rust`
   - `pip install maturin`

pylwcb

2. pylwcb

`maturin develop` pylwcb python python `.env/lib64/python3.6/site-packages/`

pylwcb devbuild.sh

3. pylwcb

2 pylwcb python python

```
$ ./devbuild.sh
$ python
>>> import pylwcb
>>> pylwcb.say_hello()
'hello from pylwcb modules'
```

4. pylwcb

BUG python rust rust

crash crash

- python `rust-gdb --args python test.py`
- r `r`
- `bt` crash

pyo3 debugging guide

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search