Introduction to xdp-tools

Toke Høiland-Jørgensen
Principal Kernel Engineer, Red Hat

NS team tech talk
June 8th, 2022

Reminder: What is XDP?

XDP (eXpress Data Path) is a Linux in-kernel fast-path

- Programmable layer in-front of traditional network stack
  - Read, modify, drop, redirect or pass
  - For L2-L3 use-cases: seeing x10 performance improvements!
- Avoiding memory allocations
  - No SKB allocations and no-init (SKB zeroes 4 cache-lines per pkt)
- Adaptive bulk processing of frames
- Very early access to frame (in driver code after DMA sync)
- Ability to skip (large parts) of kernel code
  - Evolve XDP via BPF-helpers

What is xdp-tools?

Upstream repo: https://github.com/xdp-project/xdp-tools

Contains:

- libxdp
- xdp-loader
- xdp-filter
- xdpdump
- xdp-traffigen (WiP)

Packaged in RHEL 8 & 9

libxdp and the multi-prog dispatcher

libxdp

libxdp is a small library built on top of libbpf; it provides:

- User space support code for using `AF_XDP` (moved from libbpf)
- Multi-prog dispatcher for running multiple XDP programs on a single interface

The multi-prog dispatcher

The kernel only allows executing a single XDP program per interface!

```
static volatile const struct xdp_dispatcher_config conf = {};
__attribute__ ((noinline)) int prog0(struct xdp_md *ctx) {
        volatile int ret = XDP_DISPATCHER_RETVAL;

        return ret;  /* this function is replaced on load using freplace */
}
SEC("xdp")
int xdp_dispatcher(struct xdp_md *ctx)
{
        __u8 num_progs_enabled = conf.num_progs_enabled;
        int ret;

        if (num_progs_enabled < 1)
                goto out;
        ret = prog0(ctx); /* repeat for up to 10 progs */
        if (!((1U << ret) & conf.chain_call_actions[0]))
                return ret;
out:
        return XDP_PASS;
}
```

The tools in xdp-tools

xdp-loader

General-purpose program loader based on libxdp (multi-prog support).

```
# xdp-loader load testns xdp_drop.o
# xdp-loader load testns xdp_pass_kern.o
# xdp-loader status
CURRENT XDP PROGRAM STATUS:

Interface        Prio  Program name      Mode     ID   Tag               Chain actions
--------------------------------------------------------------------------------------
lo                     <No XDP program loaded!>
wg0                    <No XDP program loaded!>
redhat0                <No XDP program loaded!>
eth0                   <No XDP program loaded!>
testns                 xdp_dispatcher    native   357  94d5f00c20184d17
 =>              50     xdp_drop                  352  57cd311f2e27366b  XDP_PASS
 =>              50     xdp_prog_simple           364  3b185187f1855c4c  XDP_PASS
```

xdp-filter

Simplistic packet filter demo (not a full-fledged firewall!)

```
# xdp-filter load testns  -f ipv6
# xdp-filter ip fc00:dead:cafe:1::1
# xdp-filter status
CURRENT XDP-FILTER STATUS:

Aggregate per-action statistics:
  XDP_ABORTED                                0 pkts           0 KiB
  XDP_DROP                                   5 pkts           0 KiB
  XDP_PASS                                   3 pkts           0 KiB

Loaded on interfaces:
                                        Enabled features
xdpfilt_alw_ip
  testns (native mode)                  ipv6,ipv4,allow

Filtered IP addresses:
                                        Mode             Hit counter
  fc00:dead:cafe:1::1                   dst              5
```

xdpdump

tcpdump-like utility, but attaches to existing XDP programs:

```
# xdpdump -i testns --rx-capture entry,exit
listening on testns, ingress XDP program ID 400 func xdpfilt_alw_ip, capture mode entry/exit, capture size 2621
1654692789.575086848: xdpfilt_alw_ip()@entry: packet size 118 bytes on if_index 19, rx queue 0, id 1
1654692789.575092034: xdpfilt_alw_ip()@exit[DROP]: packet size 118 bytes on if_index 19, rx queue 0, id 1
1654692790.590477081: xdpfilt_alw_ip()@entry: packet size 118 bytes on if_index 19, rx queue 0, id 2
1654692790.590493371: xdpfilt_alw_ip()@exit[DROP]: packet size 118 bytes on if_index 19, rx queue 0, id 2

# xdpdump -i testns  -w - | tcpdump -nr -
listening on testns, ingress XDP program ID 400 func xdpfilt_alw_ip, capture mode entry, capture size 262144 byt
reading from file -, link-type EN10MB (Ethernet), snapshot length 262144
14:54:06.190611 IP6 fc00:dead:cafe:1::2 > fc00:dead:cafe:1::1: ICMP6, echo request, id 59320, seq 1, length 64
14:54:07.216884 IP6 fc00:dead:cafe:1::2 > fc00:dead:cafe:1::1: ICMP6, echo request, id 59320, seq 2, length 64
```

These packets were filtered, but we can still see them!

xdp-trafficgen

XDP-based programmable traffic generator (not in RHEL yet):

```
# xdp-trafficgen udp ens3f1 # single core
[..]
XDP_REDIRECT    11150720 pkts (   8919659 pps)      696920 KiB (  4567 Mbits/s)

# xdp-trafficgen udp ens3f1 -t 6 # 6 cores
[..]
XDP_REDIRECT    65123603 pkts (  52095122 pps)     4070225 KiB ( 26673 Mbits/s)

# xdp-trafficgen udp ens3f1 -t 6 -d 100 # spraying over 100 dst-ports
[..]
XDP_REDIRECT     8226576 pkts (  32896120 pps)      514161 KiB ( 16843 Mbits/s)

# xdp-trafficgen tcp -i ens3f1 fe80::ee0d:9aff:fedb:11cd -p 1234
Connected to fe80::ee0d:9aff:fedb:11cd port 1234 from fe80::ee0d:9aff:fed8:f5d3 port 39500
[...]
XDP_DROP         3249504 pkts (     23878 pps)     4760015 KiB (   287 Mbits/s)
XDP_PASS               0 pkts (         0 pps)           0 KiB (     0 Mbits/s)
XDP_TX         516392331 pkts (   6217106 pps)   756434078 KiB ( 74605 Mbits/s) <--- retransmissions!
XDP_REDIRECT   545096864 pkts (   6534151 pps)   798481734 KiB ( 78410 Mbits/s)
```

Kernel samples tools

WiP: moving the tools from kernel `samples/bpf` into xdp-tools:

https://github.com/xdp-project/xdp-tools/pull/158

End: Questions?