XDP PRACTICE

제목	Docker 환경에서 간단한 eBPF/XDP program 을 실행하자.
작성	[18반] 라민우

- Docker 환경에서 간단한 eBPF/XDP program을 실행하자.

A. 활동 목표

기본적인 eBPF/XDP 프로그램의 동작 방식을 이해하기 위해,
XDP 프로그램을 디바이스 드라이버에 설치해서 정상적으로 작동하는 것을 확인하자.

B. 실행 흐름

a. 사용할 XDP 프로그램 작성

- prog1 : xdp1.c / Drop All

```c
#include <linux/bpf.h>
#include <bpf/bpf_helpers.h>
SEC("xdp1")
int drop (struct xdp_md *ctx){
        return XDP_DROP;
}
```

- prog2 : xdp2.c / Drop TCP

```c
#include <arpa/inet.h>
#include <linux/bpf.h>
#include <linux/if_ether.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <bpf/bpf_helpers.h>

SEC("xdp2")
int drop_tcp(struct xdp_md *ctx) {
    void *data_end = (void *)(long)ctx->data_end;
    void *data = (void *)(long)ctx->data;

    struct ethhdr *eth = data;
    struct iphdr *ip;
    struct tcphdr *tcp;

    if ((void *)eth + sizeof(struct ethhdr) > data_end)
        return XDP_PASS;

    ip = data + sizeof(struct ethhdr);
    if ((void *)ip + sizeof(struct iphdr) > data_end)
        return XDP_PASS;

    if (ip->protocol == IPPROTO_TCP) {
        return XDP_DROP;
    }

    return XDP_PASS;
}

char _license[] SEC("license") = "GPL";
```

b. 컴파일(ELF 오브젝트 파일로 전환)

clang -O2 -target bpf -c <name>.c -o <name>.o

c. 커널로 오브젝트 파일 전달 & 디바이스 드라이버에 설치

ip link set dev <device> -target <target> obj <name>.o sec <section>

- 디바이스 드라이버에 설치

ip link show <dev>

- 디바이스에 포함된 XDP 확인 가능

ip link set dev <device> <target> off

- 설치된 xdp 프로그램 제거

C. 실습 환경 구축

```
root@ee92cb370413:/xdpf# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 24.04 LTS
Release:        24.04
Codename:       noble
root@ee92cb370413:/xdpf# uname -r
6.5.0-28-generic
```

도커 컨테이너 > 우분투 24.04 LTS 환경에서 실습

명령어&설치항목 > apt update, apt upgrade
- vim,clang,llvm,gcc-multilib,libbpf-dev,libxdp-dev,sudo,iproute2,iputils-ping,gcc-multilib,iproute2

D. 분석 및 정리

a. xdp1.c

ICMP PING을 이용했으며, 들어오는 모든 패킷이 XDP_DROP 되는 것을 볼 수 있다.
Network Interface Card 로 패킷이 들어오면, 디바이스 드라이버에 설치된 XDP 프로그램이 실행된다.

```
root@ee92cb370413:/xdpf# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=36.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=34.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=33.8 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 33.805/34.845/36.122/0.960 ms
root@ee92cb370413:/xdpf# ip link set dev eth0 xdp obj xdp1.o sec xdp1
root@ee92cb370413:/xdpf# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6127ms
```

b. xdp2.c

TCP (SYN FLAG) 패킷 1개를 보냈으며,
XDP를 적용하지 않은 경우 정상적인 패킷 전달을 수행했으며,
반대로 XDP를 적용했을 때는 정상적인 패킷 전달을 실패하게 된다.

- DOCKER SPACE(No XDP)

```
root@ee92cb370413:/xdpf# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
09:24:24.588000 IP 172.17.0.1.2722 > ee92cb370413.0: Flags [none], win 512, length 0
09:24:24.588043 IP ee92cb370413.0 > 172.17.0.1.2722: Flags [R.], seq 0, ack 829909612, win 0, length 0
```

- DOCKER SPACE(XDP enabled)

```
root@ee92cb370413:/xdpf# ip link set dev eth0 xdp obj xdp2.o sec xdp2
root@ee92cb370413:/xdpf# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
09:24:43.192264 IP 172.17.0.1.1225 > ee92cb370413.0: UDP, length 0
09:24:43.192285 IP ee92cb370413 > 172.17.0.1: ICMP ee92cb370413 udp port 0 unreachable, length 36
```

- HOST SPACE

```
user@user-VirtualBox:~$ sudo hping3 -c 1 172.17.0.2
HPING 172.17.0.2 (docker0 172.17.0.2): NO FLAGS are set, 40 headers + 0 data bytes
len=40 ip=172.17.0.2 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=12.2 ms

--- 172.17.0.2 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 12.2/12.2/12.2 ms
user@user-VirtualBox:~$ sudo hping3 -c 1 172.17.0.2
HPING 172.17.0.2 (docker0 172.17.0.2): NO FLAGS are set, 40 headers + 0 data bytes

--- 172.17.0.2 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
user@user-VirtualBox:~$ sudo hping3 -2 -c 1 172.17.0.2
HPING 172.17.0.2 (docker0 172.17.0.2): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=172.17.0.2 name=UNKNOWN
status=0 port=1225 seq=0

--- 172.17.0.2 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 20.6/20.6/20.6 ms
```

c. 결론

XDP (TCP DROP) 프로그램을 dev 에 설치할 경우, udp 프로토콜 패킷은 tcpdump 프로그램을 통해 들어오는 것을 확인할 수 있으나, tcp 프로토콜 패킷은 디바이스 드라이버에서 이미 Drop 처리 됐기 때문에 탐지되지 않는다.
이를 통해 XDP 프로그램 코드와 XDP 프로그램의 실제 작동 결과를 알 수 있었다.