XDP-Simple Practice

제목	Docker 환경에서 간단한 eBPF program을 실행하자.
작성	[18반] 라민우

작업 순서

명령어 순서 소개 ▶ 컨테이너 환경 세팅 ▶ 실습 ▶ 정리

☆ [A->B->C] 명령어 순서 소개

A. 사용할 폴더 생성

(최상단 디렉토리 추천)

B. 컴파일[최적화, 타겟 설정]

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

C. 오브젝트 파일 삽입

ip link set dev <device> <타겟 기능> obj <name>.o sec <section>

else

ip link show <dev> : device에 포함된 XDP 확인 가능

ip link set dev <device> <타겟 기능> off : 위 명령어로 삽입한 기능 off

☆ 컨테이너 환경 세팅

ubuntu 22.04 / kernel version 5.0 이상

> docker를 통해 ubuntu 컨테이너 실행

apt update

apt upgrade

[install list]

vim,clang,llvm,gcc-multilib,libbpf-dev,libxdp-dev,sudo,iproute2,iputils-ping

gcc-multilib,iproute2

★ gcc-multilib <- 미설치시 컴파일 난이도 ▲

★ 실습 1 (just drop)

ubuntu 컨테이너 실행 > 환경 세팅(install) > 소스코드 작성 > 컴파일 > NIC 삽입 > 기능 off

```
#include <linux/bpf.h>
#include <bpf/bpf_helpers.h>

SEC("xdp")
int xdp_prog(struct xdp_md *ctx) {
    return XDP_DROP;
}

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

목표 : 들어오는 모든 패킷을 drop

```
root@ee92cb370413:/xdp# 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 ---
6 packets transmitted, 0 received, 100% packet loss, time 51
```

결과 : 들어오는 모든 패킷 dropped

★ 실습 2 (TCP drop)

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

SEC("prog")
int block_udp(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 udphdr *udp;

    if ((void *)(eth + 1) > data_end)  // 수정된 부분
        return XDP_PASS;

    if (eth->h_proto != htons(ETH_P_IP))
        return XDP_PASS;

    ip = (struct iphdr *)(eth + 1);
    if ((void *)(ip + 1) > data_end)  // 수정된 부분
        return XDP_PASS;

    if (ip->protocol != IPPROTO_UDP)
        return XDP_PASS;

    udp = (struct udphdr *)(ip + 1);
    if ((void *)(udp + 1) > data_end)  // 수정된 부분
        return XDP_PASS;

    if (udp->dest != htons(80))
        return XDP_PASS;

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

```
root@ee92cb370413:/xdp# 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=45.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=39.2 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=36.7 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 36.650/40.405/45.384/3.668 ms
root@ee92cb370413:/xdp# ping www.naver.com
PING www.naver.com.nheos.com (223.130.200.219) 56(84) bytes of dat
^C
--- www.naver.com.nheos.com ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1077ms

root@ee92cb370413:/xdp#
```

결과 : UDP(DNS)의 ping은 정상적으로 수행되지만 TCP 연결(네이버 홈페이지)은 안된다.