Fall ’19 COSE322-00

System Programming

HW2 appendix

2019. 11. 27.

HW appendix

Socket Client Q&A

- **서버 가상 머신은 커넥션 종료까지 패킷을 보냄**
  - 1. 따라서 client를 ctrl+c로 종료해주거나
  - 2. client에 시간제한, 반복 수 제한 등등을 추가해 종료해주시면 됩니다.
  - 넷필터 훅에 로그만 잘 찍히면 되므로 짧은 시간동안만 패킷을 주고받아도 됩니다.
  - 2차과제에서는 <port.txt>를 생성하거나 제출할 필요가 없습니다

Proc Q&A

- **Proc fs는 커널 모듈에 포워딩할 포트 번호를(33333 or 3333) 전달하는데 사용됩니다.**
  - 실습 슬라이드 예시로 echo 부분만 존재하는데, echo 와 proc_write 를 이용해 포트 번호를 커널 모듈로 전달해주시면 됩니다.
  - 어려우시면 먼저 포트 번호를 하드코딩해 과제를 수행하고, 나중에 proc을 구현하셔도 됩니다.

Routing Table

❖ 사용법

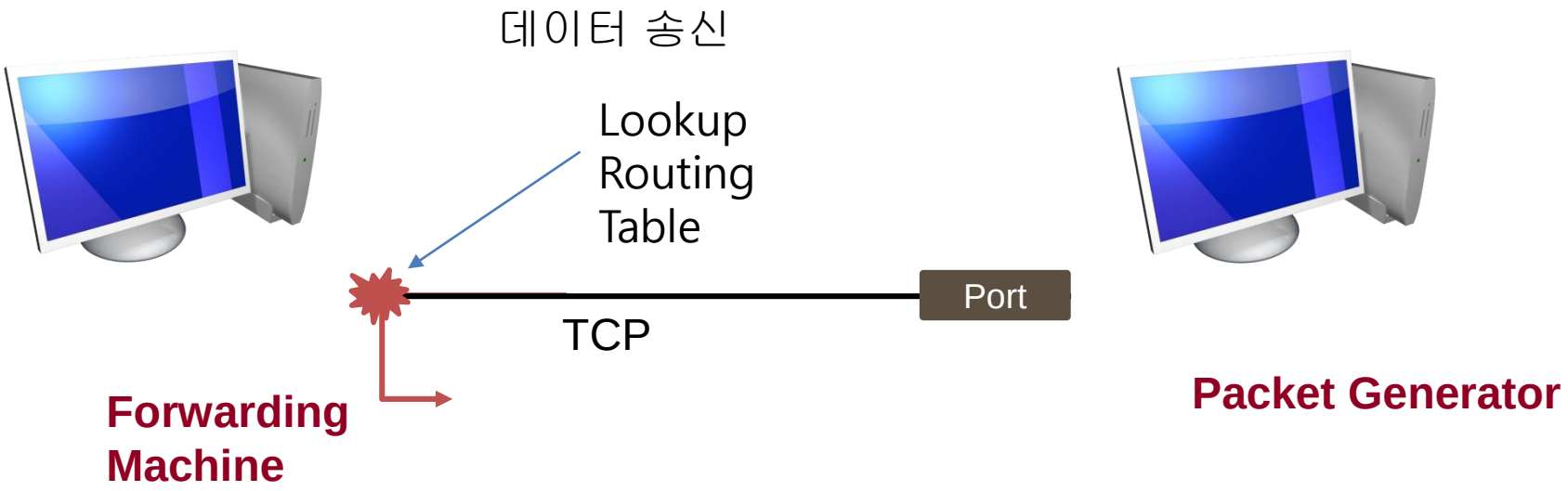
- 라우팅 테이블의 엔트리는 인터페이스에 할당한 IP에 대해선 자동으로 잡힘
- 라우팅 테이블 엔트리를 임의로 추가, 삭제할 수 있음.
  - Route add, del –net 명령어 이용
  - 구글링을 통해 다양한 예시 확인 가능
- 예시 (route add)

```
root@oslab-VirtualBox:/home/oslab# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.2.2        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        *               255.255.255.0   U     100    0        0 enp0s3
link-local      *               255.255.0.0     U     1000   0        0 enp0s8
```

  - route add -net 100.1.1.0 netmask 255.255.255.0 dev enp0s8
  - route add -net 111.1.1.0 netmask 255.255.255.0 dev enp0s8

```
root@oslab-VirtualBox:/home/oslab# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.2.2        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        *               255.255.255.0   U     100    0        0 enp0s3
100.1.1.0       *               255.255.255.0   U     0      0        0 enp0s8
111.1.1.0       *               255.255.255.0   U     0      0        0 enp0s8
link-local      *               255.255.0.0     U     1000   0        0 enp0s8
```

예시


데이터 송신
Lookup
Routing
Table
Port
TCP
Forwarding
Machine
Packet Generator

과제 환경에서는?

- **IP Forwarding의 개념을 반드시 숙지!**
- **Routing Table을 추가**
  - 패킷 송신 interface 입력 주의!
- **Netfilter를 이용해야 한다 → PRE_ROUTING 지점이 핵심**
  - Routing Table의 맞게 Packet 내용 변조
  - 내용 변조시 주의해야할 사항 다음과 같음
    - 교수님 수업과 실습 수업 때 언급한 endian 주의
    - 어느 layer의 header를 고칠지 명확하게 숙지
    - 헤더 구조체 멤버 변수 정확히 파악
  - PRE_ROUTING은 Routing Table을 거치기 전이라는 점 숙지

과제 FAQ

- **Netfilter 지점을 3곳 지정한 이유**
  - Forwarding 동작이 정확히 이루어지는지 확인하기 위해
  - 가장 핵심 지점은 PRE_ROUTING
  - 나머지는 확인하는 지점
- **Port를 변조하는 이유**
  - IP header 뿐만 아니라 TCP header 가져오고 변조할 수 있는지 파악
  - Forwarding 자체와는 연관은 없음

관련 코드

```
/* Network layer. */
if (key->eth.type == htons(ETH_P_IP)) {
        struct iphdr *nh;
        __be16 offset;

        error = check_iphdr(skb);
        if (unlikely(error)) {
                memset(&key->ip, 0, sizeof(key->ip));
                memset(&key->ipv4, 0, sizeof(key->ipv4));
                if (error == -EINVAL) {
                        skb->transport_header = skb->network_header;
                        error = 0;
                }
                return error;
        }

        nh = ip_hdr(skb);
        key->ipv4.addr.src = nh->saddr;
        key->ipv4.addr.dst = nh->daddr;

        key->ip.proto = nh->protocol;
        key->ip.tos = nh->tos;
        key->ip.ttl = nh->ttl;

        offset = nh->frag_off & htons(IP_OFFSET);
        if (offset) {
                key->ip.frag = OVS_FRAG_TYPE_LATER;
                return 0;
        }
        if (nh->frag_off & htons(IP_MF) ||
                skb_shinfo(skb)->gso_type & SKB_GSO_UDP)
                key->ip.frag = OVS_FRAG_TYPE_FIRST;
        else
                key->ip.frag = OVS_FRAG_TYPE_NONE;
```

❖ Network Layer

- Struct iphdr
  - IP Header 정보 포함
  - IP레이어에 맞는 여러 정보 존재
  - Saddr: Source IP
  - Daddr: Destination IP
- ip_hdr(sk_buff skb)
  - IP 헤더 정보를 담은 구조체의 주소 리턴
- /include/uapi/linux/ip.h

관련 코드

```
/* Transport layer. */
if (key->ip.proto == IPPROTO_TCP) {
    if (tcphdr_ok(skb)) {
        struct tcphdr *tcp = tcp_hdr(skb);
        key->tp.src = tcp->source;
        key->tp.dst = tcp->dest;
        key->tp.flags = TCP_FLAGS_BE16(tcp);
    } else {
        memset(&key->tp, 0, sizeof(key->tp));
    }
```

- **Transport Layer**
  - Struct tcp_hdr 존재
    - TCP 헤더 정보 포함
    - TCP에 맞는 여러 정보 포함
    - Source: Source Port
    - Dest: Destination Port
  - tcp_hdr(sk_buff skb)
    - TCP헤더 정보를 담은 tcp_hdr 의 주소 리턴
  - /include/uapi/linux/tcp.h

Endian + α

Macro	Meaning (short; 2bytes, long; 4bytres)
htons	Host-to-network byte order (short)
htonl	Host-to-network byte order (long)
ntohs	Network-to-host byte order (short)
ntohl	Network-to-host byte order (long)

- **이를 활용하여 변조를 해야함**
- **단, IP주소의 경우 어떻게 해야 할지 고민 필요**
  - IP 주소 입력은 String
  - Iphdr→saddr은 String이 아님 → 확인해볼 것
  - 변환해주는 '무언가'가 필요

Print IP in readable format

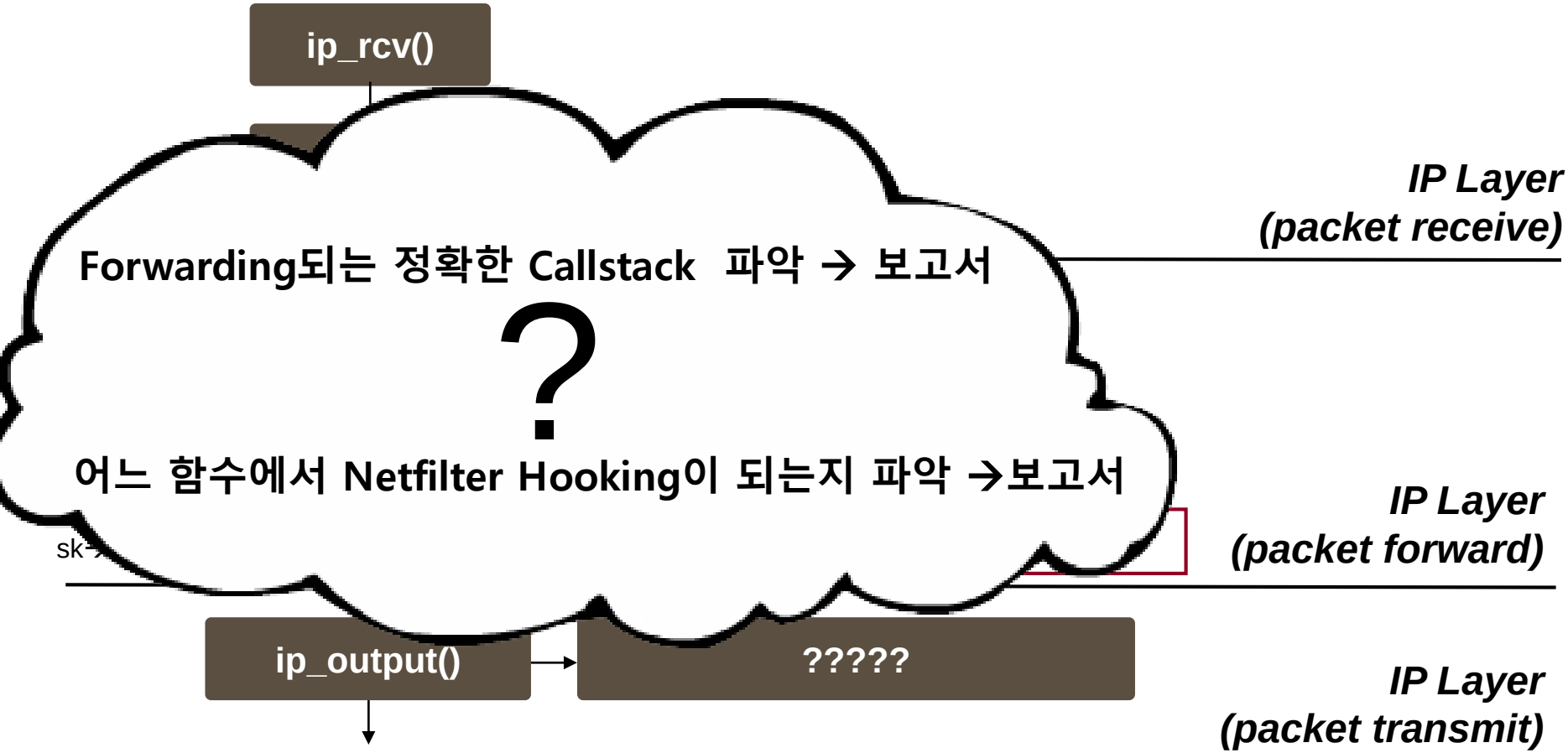
- **IP주소는 String이 아님**
  - Print → 원하는 format 이 아님
- **NIPQUAD 활용**
  - /include/linux/kernel.h
  - 쓸 수 있는 방법은 두가지
    - Include
    - Code를 가져옴

```
printk(KERN_INFO "IP header : original source : %d.%d.%d.%d\n", NIPQUAD(iph->saddr));
```

사용방식은 다음과 같음

Callstack Analysis

❖ Function calls in Kernel layer (PF_INET & TCP)



Q&A