2020 스마트 공유기 개발

2020. 10. 23

중부대학교 정보보호학과

지도교수 : 김민수 교수님

2 조 박지윤 나지혜
윤성준 전예진
정현성 표상영

목 차

조원 편성

이 름	역 할
박지윤(조장)	네트워크와 연동할 웹서버 구축
나지혜	네트워크와 연동할 웹서버 구축
윤성준	SNI 차단우회 네트워크 프로그래밍
전예진	자료수집 및 정리, 보고서 작성
정현성	프로젝트 관련 자료 분석, PPT 작성
표상영	ARP 스푸핑

주제 선정(1/2)

재택근무 시대, 공유기 노린 악성코드 증가, 예방책은?

추현우 기자 | 승인 2020.03

각종 가정용 유무선 공유기 제

[디지털투데이 추현우 기자] 거리두기 캠페인이 널리 확산 안 취약점을 노리는 악성코드

인터넷 보안 소프트웨어 기업 련 악성코드(멀웨어·malware)

"공공 와이파이 10개 중 4개 보안 허술"

노동균 기자

입력 2017.10.12 14

만2300개소를 설

과기정통부의 '공 WPA2 방식을 따 기는 데이터 암호

연예인 폰 해킹 어떻게? 25초 만에 뚫린 '인터넷 공유기'

[조세일보] 강대경 기자

보도 : 2020.01.20 11:38 수정 : 2020.01.20 11:38

인터넷 공유기 통해 특정 대상 해킹 가능
가정 내 네트워크 중심이나 보안은 취약
'25초'만에 후다닥 뚫리는 단순한 숫자 암호
'복잡한 암호'와 '2단계 인증' 필수!

최근 유명 연예인 J씨는 삼성 스마트폰에 있던 사생활 정보가 노출되면서 크고 작은 파장을 일으키고 있다. 해커는 그의 '아이디와 비밀번호'를 알아냈고 다른 삼성 스마트폰으로 삼성 클라우드에 접속해 2013년 당시 전화번호, 문자, 사진 등을 복원하는데 성공했다

널리 보급되어 모두가 사용하는 공유기. 과연 보안은 완벽할까?

주제 선정(2/2)

- IT 기술의 발전으로 도시(스마트 시티), 주거(스마트 홈), 사물(스마트 IoT) 등 모든 분야가 네트워크화/스마트화
- 그러나 가정집, 카페, 길거리, 교통수단 등 우리가 생활하는 모든 곳에 존재하는 공유기는 10년 전 모습 그대로
- 사이버 위협에 대응하여 우리 모두가 공유하는 공유기의 안전은 ?

보다 효율적이고 더욱 안전한 '2020 스마트 공유기' 개발 추진

구상도(1/4)

기능적 측면

기존 공유기

1. 기본적인 공유기 기능
2. 속도 제한이 걸려있음
3. 하드웨어 성능을 100% 활용하지 못함

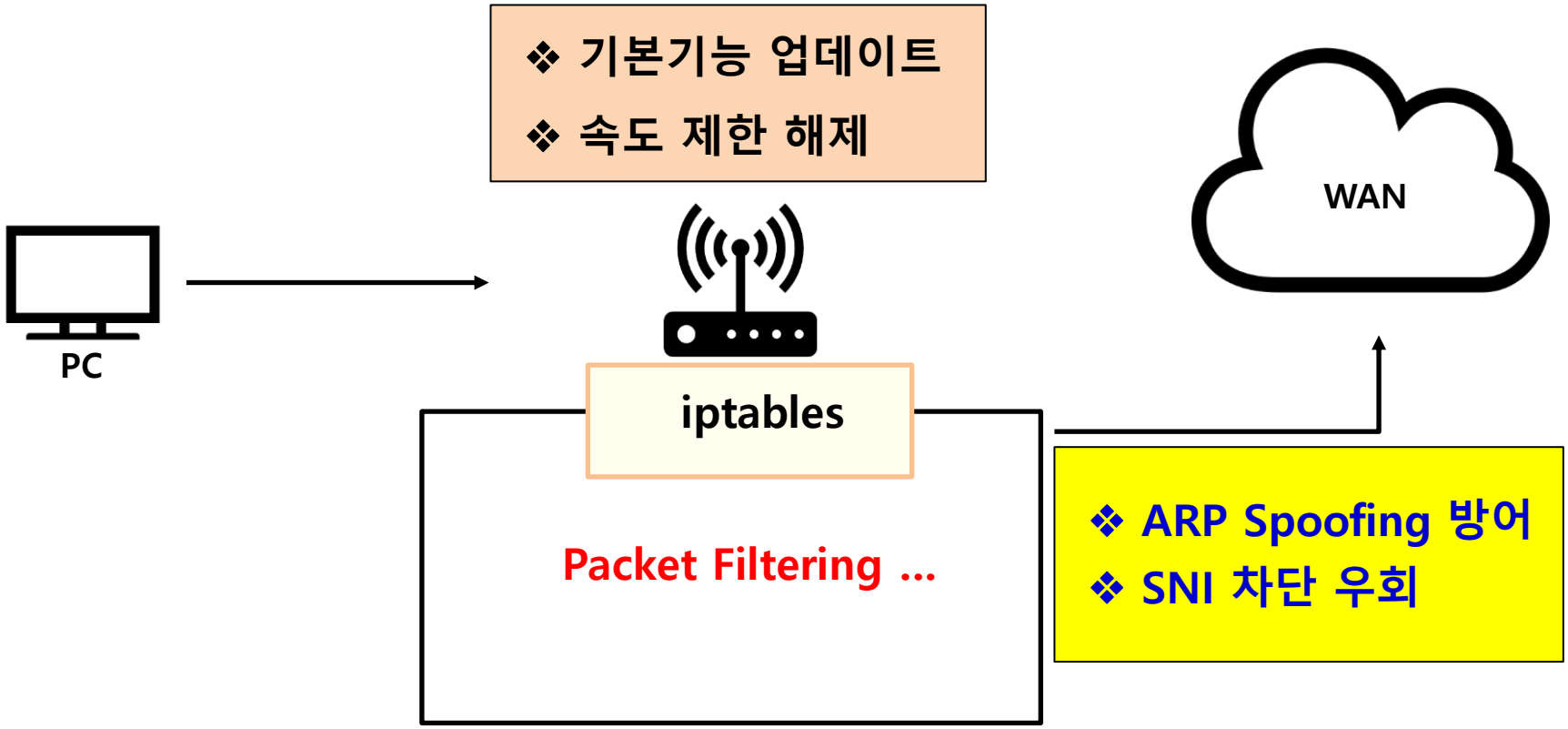
개발 공유기

1. 기본적인 공유기 기능 업데이트
2. ARP Spoofing 방어
3. SNI 차단 우회

공유기의 이점과 한계점을 고려하여 기능 추가 및 성능 향상

구상도(2/4)

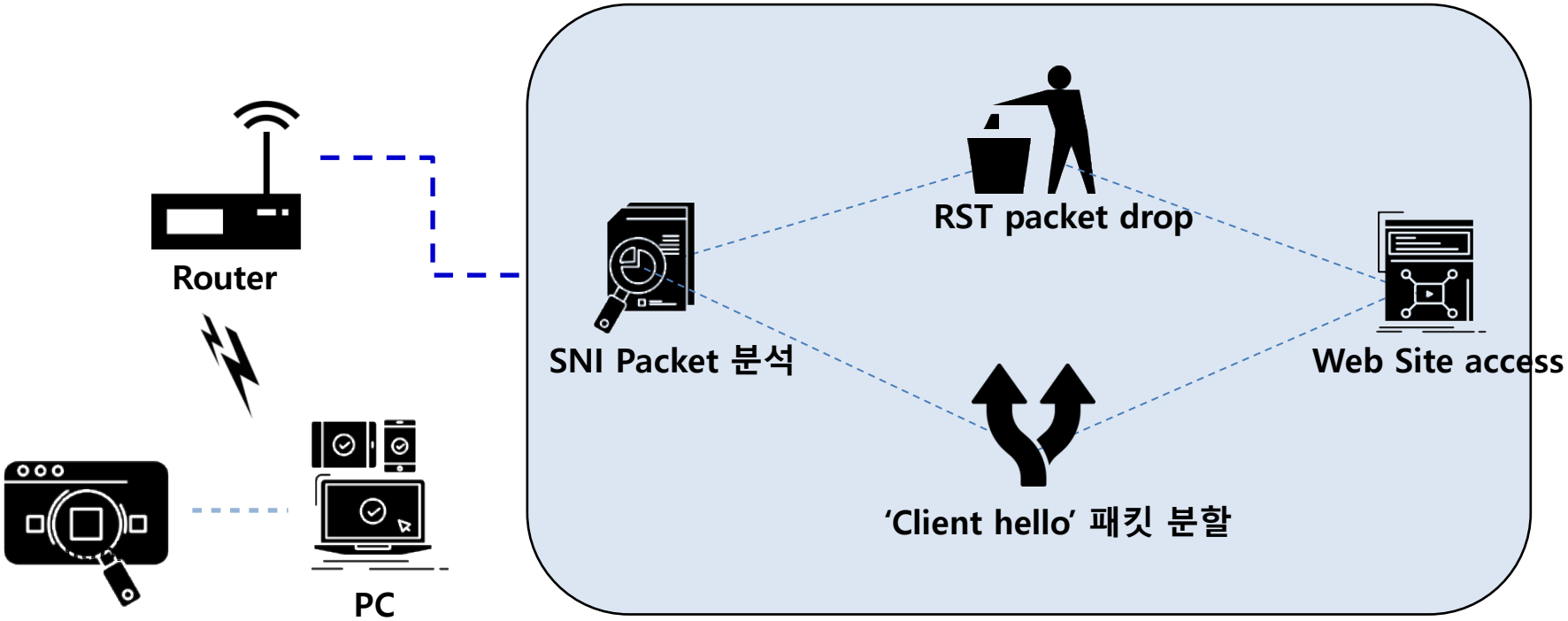
운영 특성(전체 구상도)



공유기 성능 향상 및 보안 기능 보강으로 스마트 공유기 운용

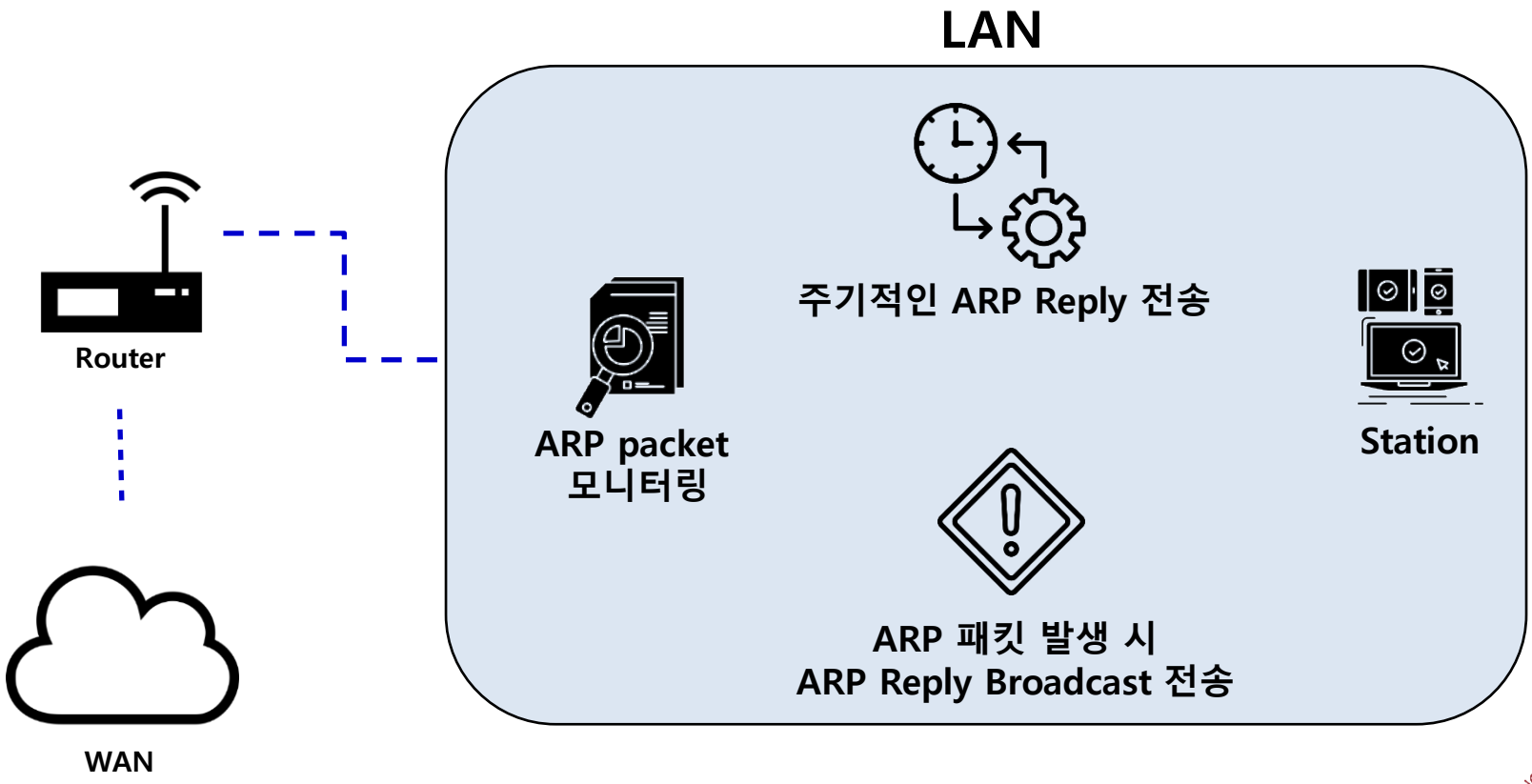
구상도(3/4)

운영 특성(SNI 차단우회)

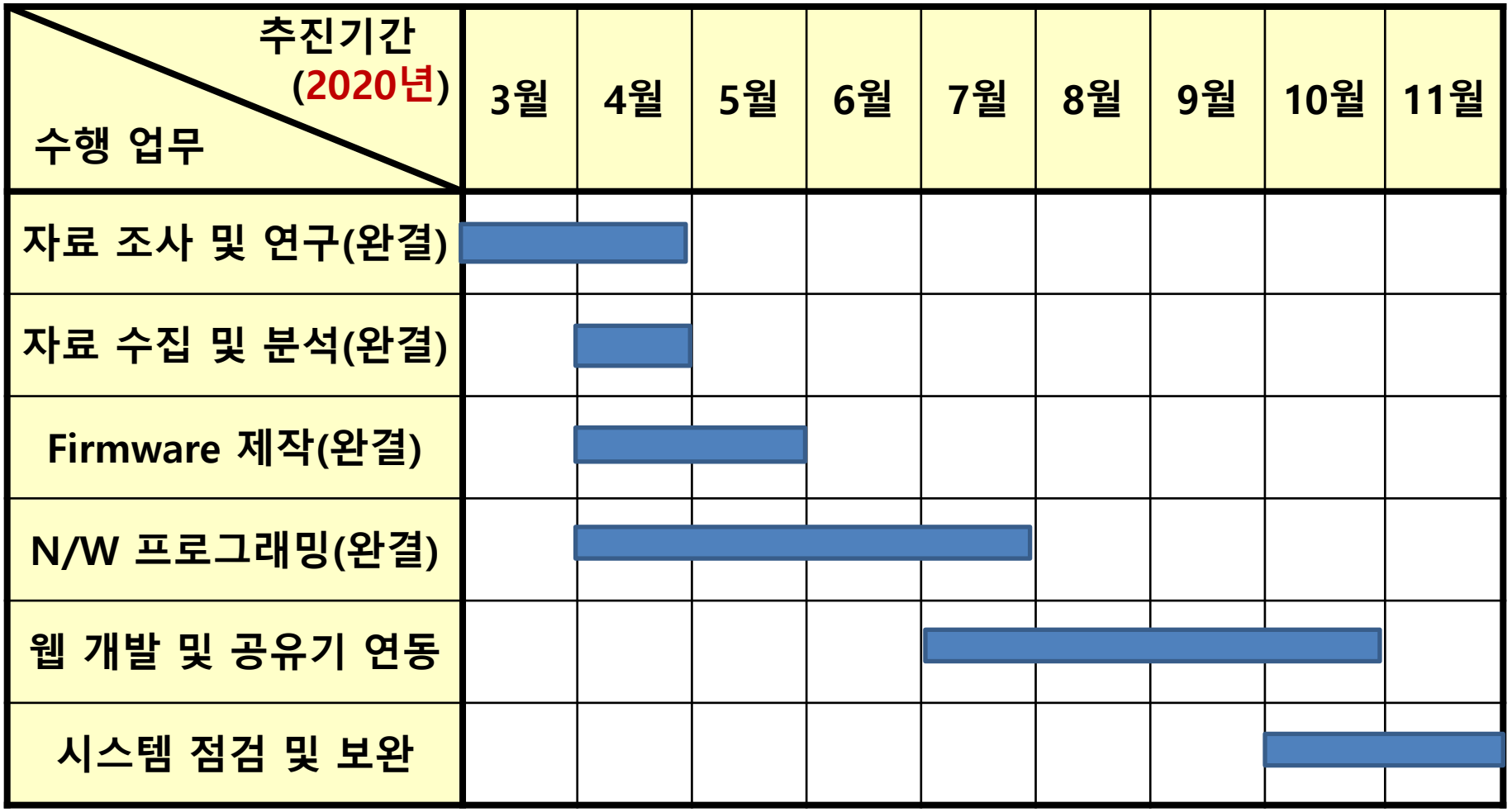


구상도(4/4)

운영 특성(ARP Spoofing 방어)



추진 경과


추진기간
(2020년)
수행 업무
3월
4월
5월
6월
7월
8월
9월
10월
11월
자료 조사 및 연구(완결)
자료 수집 및 분석(완결)
Firmware 제작(완결)
N/W 프로그래밍(완결)
웹 개발 및 공유기 연동
시스템 점검 및 보완

개발 환경 및 개발 내용(1/9)

개발 환경

Web

PHP, Luci, MYSQL

Development Language

Python, C++

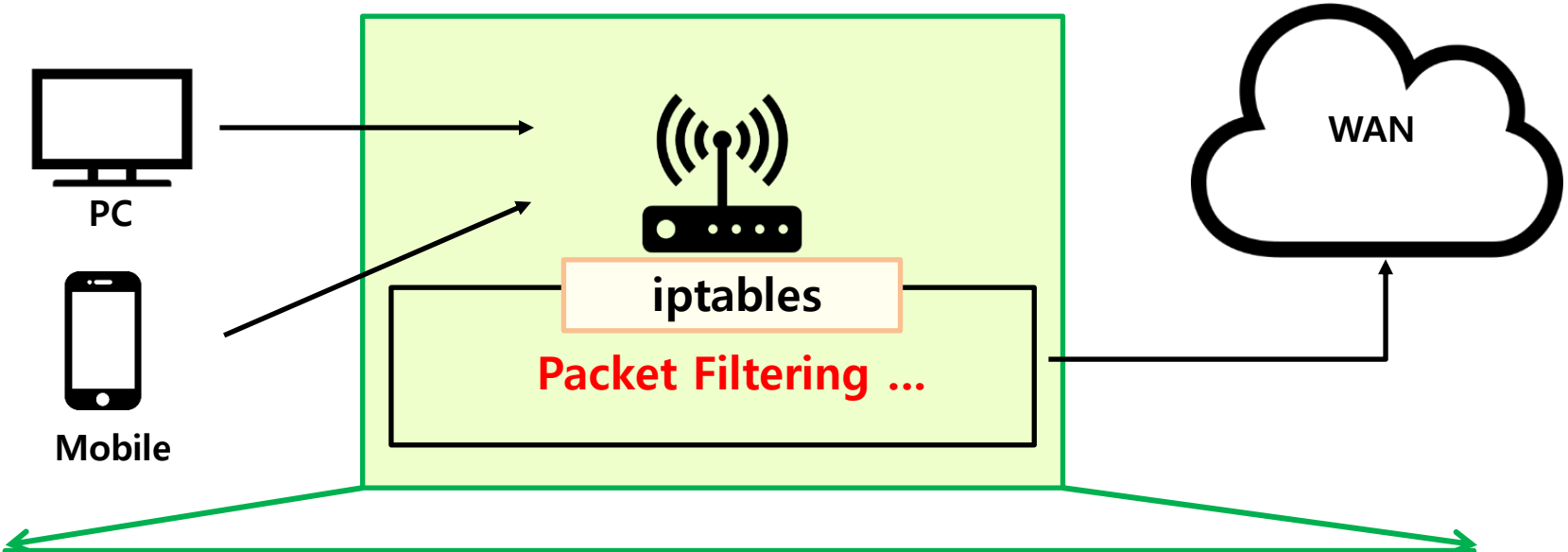
Firmware

LEDE(OpenWrt)

개발 환경 및 개발 내용(2/9)

개발 내용(1/8)

◆ 시스템 및 운영체제



◆ 공유기 개발을 위해 공유기에 로드할 'OS Firmware 개발'(Linux)

- LEDE Project(OpenWRT)를 활용, Firmware Image를 빌드
- 개발 프로그램은 공유기에 SSH로 접속하여 업데이트 진행

개발 환경 및 개발 내용(3/9)

개발 내용(2/8)

◆ 운영체제 구축

OS firmware

OpenWrt in QEMU MIPS

Use QEMU >= 2.2 (earlier versions can have bugs with MIPS16) ticket 16881 - Ubuntu 14.03.x LTS uses QEMU 2.0 which is has this bug.

The "malta" platform is meant for use with QEMU for emulating a MIPS system.

The `malta` target supports both big and little-endian variants, pick the matching files and qemu version (`qemu-system-mips`, or `qemu-system-mipsel`).

```
qemu-system-mipsel -kernel openwrt-malta-le-vmlinux-initramfs.elf -nographic -m 256
```

In recent enough versions one can enable ext4 root filesystem image building, and since r46269 (only in trunk, it's not part of the 15.05 CC release)

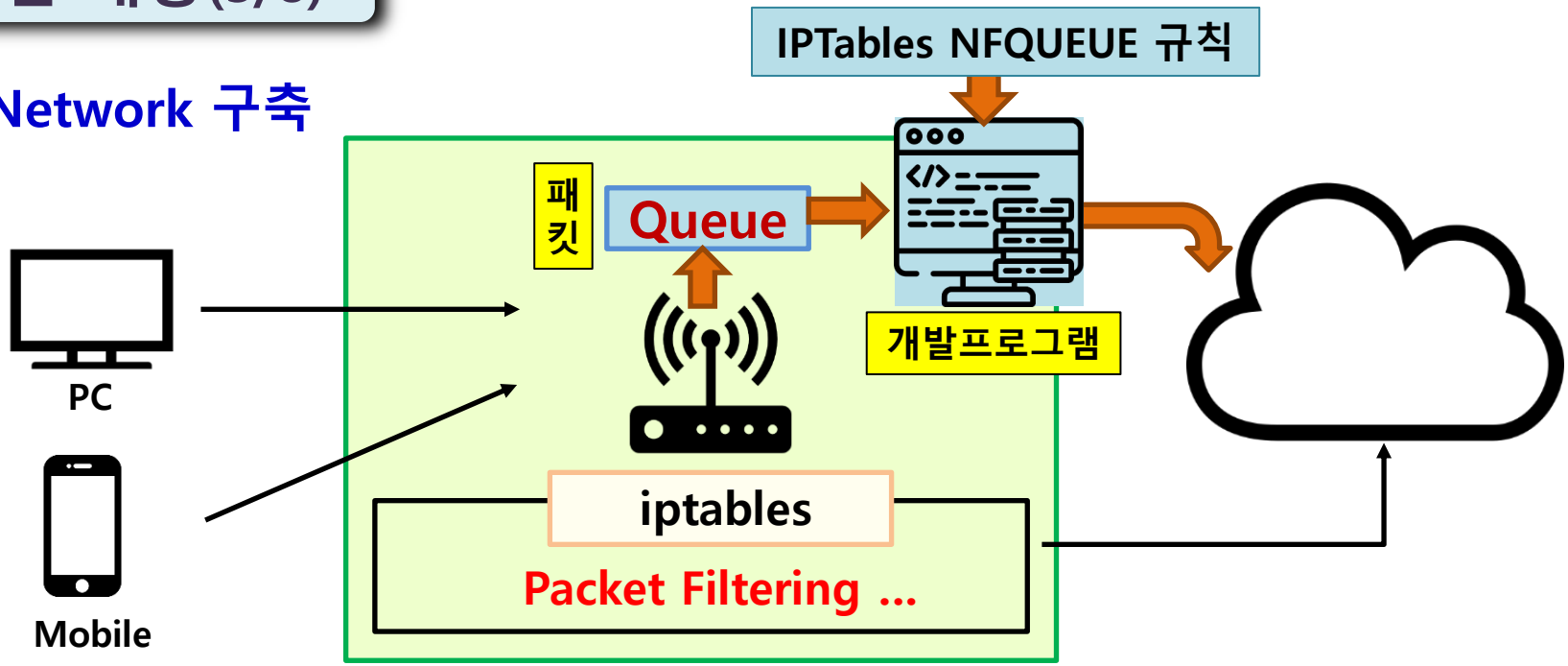
```
qemu-system-mipsel -M malta \
-hda openwrt-malta-le-root.ext4 \
-kernel openwrt-malta-le-vmlinux.elf \
```

공유기에 로드할 OS firmware 구축

개발 환경 및 개발 내용(4/9)

개발 내용(3/8)

◆ Network 구축



◆ IPTables NFQUEUE 규칙을 설정, 외부행 패킷은 Queue에 대기

◆ Netfilter Library를 활용, Queue의 패킷을 필터링(개발 프로그램)

⇨ 필터 결과에 따라 패킷이 DROP, ACCEPT ... 처리

많은 패킷을 처리해야 하기 때문에 적절한 Thread Scheduling이 요구

개발 환경 및 개발 내용(5/9)

개발 내용(4/8)

◆ IPTable 규칙 설정

IPTables NFQUEUE 규칙

```
# iptables -A OUTPUT -p tcp --dport 443 -j NFQUEUE --queue-num 0
# iptables -A OUTPUT -p tcp --dport 80 -j NFQUEUE --queue-num 0
```

IPTables 규칙 리스트

```
root@kali:~/Desktop/Smart_Router/SNI_bypass/netfilt
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
```

패킷 분할 성공

Capturing from nfqueue

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

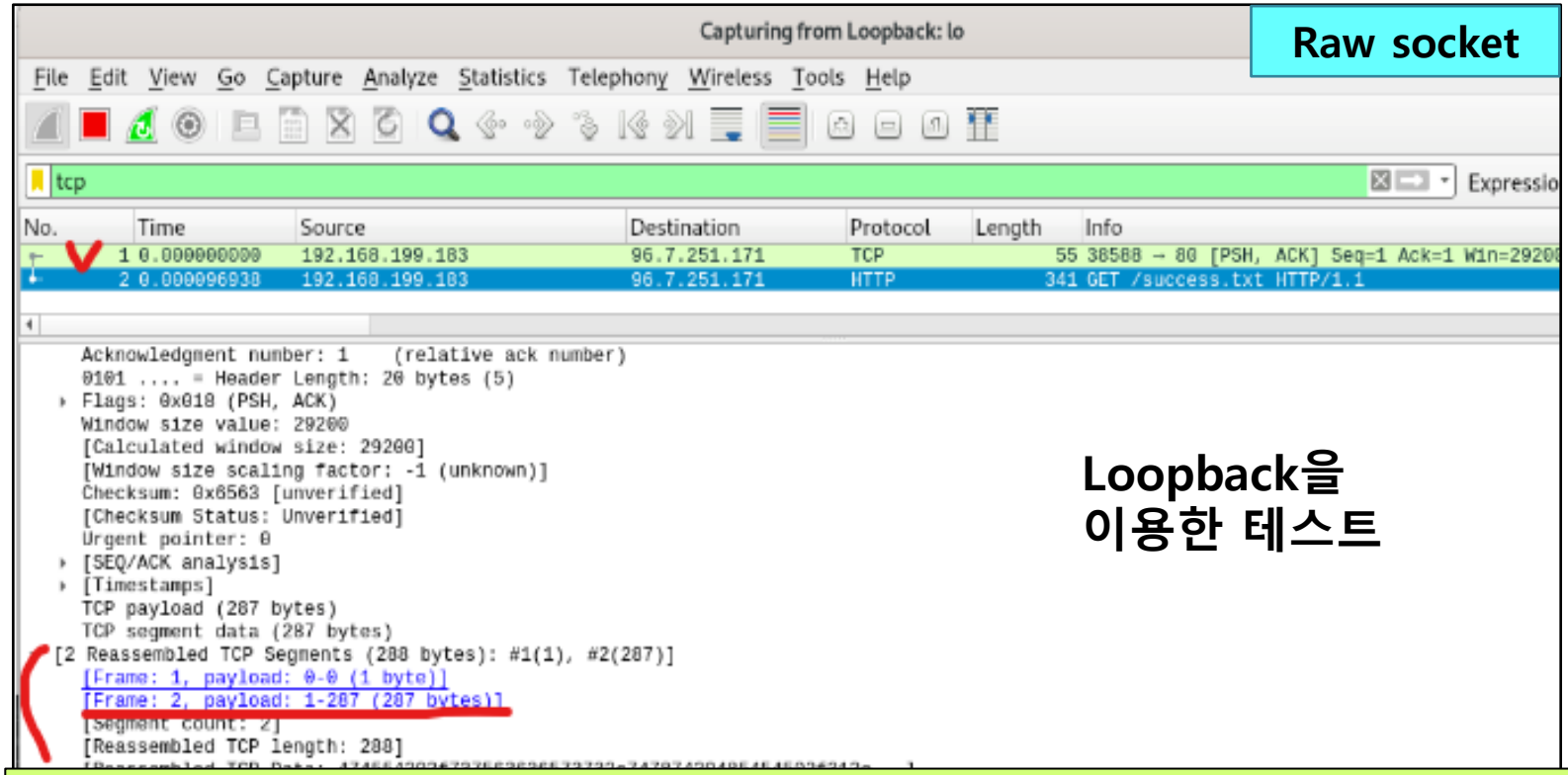
No.	Time	Source	Destination	Protocol	Length	Info
2	0.250907	192.168.199.183	23.46.147.171	TCP	60	33174 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
3	2.052024	192.168.199.183	52.26.114.88	TCP	60	47352 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
4	2.329802	192.168.199.183	52.26.114.88	TCP	60	47354 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
5	2.878556	192.168.199.183	99.86.144.34	TCP	60	40904 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
6	3.128621	192.168.199.183	99.86.144.34	TCP	60	40906 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
7	4.186162	192.168.199.183	172.217.161.170	TCP	60	53056 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
8	4.436263	192.168.199.183	172.217.161.170	TCP	60	53058 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
9	4.956694	192.168.199.183	23.46.147.186	TCP	60	54046 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
10	5.206689	192.168.199.183	23.46.147.186	TCP	60	54048 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460

IPTables의 규칙 설정 및 패킷 분할 구현

개발 환경 및 개발 내용(6/9)

개발 내용(5/8)

◆ 유해사이트 차단(우회) 시도



Raw socket

Loopback을
이용한 테스트

Raw socket을 이용하여 http,https로
송신되는 패킷을 분할하여 차단 우회를 시도

개발 환경 및 개발 내용(7/9)

개발 내용(6/8)

◆ 유해사이트 차단 과정

```
u_char *assemble;
if(fst) //처음 보내는 패킷과 두번째 보내는 패킷의 사이즈가 달라서 if 를 사용함
assemble=static_cast<u_char*>(malloc(static_ca
else
assemble=static_cast<u_cha
```

패킷 분할 사이즈 조절

헤더값 입력

```
else if(tcp_segment_len >0){
  if(fst){ //여기선 1byte만 먼저 보낸다.
```

첫번째 분할 패킷 전송

```
else{ //sequence number 더하기
    if((i-iptcp_len) <remainter_segment)
    assemble[i]=packet[i+1];
    else{
    struct tcphdr * as_tcphdr = reinterpret_cast<struct tcphdr*>(assemble+(iphdr->ihl*4));
    as_tcphdr->seq = htonl(ntohl(as_tcphdr->seq) + 1);
    sendto_packet(assemble,iptcp_len+remainter_segment);
    send=false;
    break;
}
```

두번째 패킷 분할 전송

분할 패킷을 전송하는 절차를 거쳐 유해사이트를 차단

개발 환경 및 개발 내용(8/9)

개발 내용(7/8)

◆ ARP spoofing(1/2)

◈ 차단방법

- 주기적인 ARP패킷 브로드캐스트를 통한 ARP Spoofing공격 탐지/차단

◈ 기대효과

- 공격자가 임의의 ARP패킷으로 공격을 하더라도 정상적인 ARP패킷이 주기적으로 발생되기 때문에 ARP Spoofing 공격 차단이 가능

개발 환경 및 개발 내용(9/9)

개발 내용(8/8)

◆ ARP spoofing(2/2)

각종 오류 처리 함수

```
void err_print(int err_num)
{
```

IP, Mac주소 자동 탐색

```
void find_me(char *dev_name)
{
    FILE *ptr;
    char MAC[20];
    char IP[20] = {0,};
    char cmd[300] = {0,};

    sprintf(cmd, "ifconfig %s | grep HWaddr | awk '{print $5}'", dev_nam
    ptr = popen(cmd, "r");
    fgets(MAC, sizeof(MAC), ptr);
    pclose(ptr);
    ether_aton_r(MAC,&gateway_mac);

    printf("[*] FIND MAC: %s",MAC);


    sprintf(cmd,"ifconfig %s | egrep 'inet addr:'
    ptr = popen(cmd, "r");
    fgets(IP, sizeof(IP), ptr);
    pclose(ptr);
    inet_aton(IP+5,&gateway_ip.sin_addr);

    printf("[*] FIND IP: %s\n",IP+5);

}
```

ARP 브로드캐스트 처리

```
void arp_broadcast()
{
    struct mine *protoType;
    struct mine _protoType;
    struct sockaddr_in target_ip;
    u_int8_t pkt_data[PKT_SIZE] = {0,};

    protoType = (struct mine *)pkt_data;

    memcpy(pkt_data, &_protoType, ARP_SIZE);
    inet_aton("192.168.0.2", &target_ip.sin_addr);

    memcpy(protoType->src_mac, gateway_mac.ether_addr_octet, 6);
    memcpy(protoType->s_mac, gateway_mac.ether_addr_octet, 6);
    protoType->s_ip = gateway_ip.sin_addr;
    protoType->t_ip = target_ip.sin_addr;
```

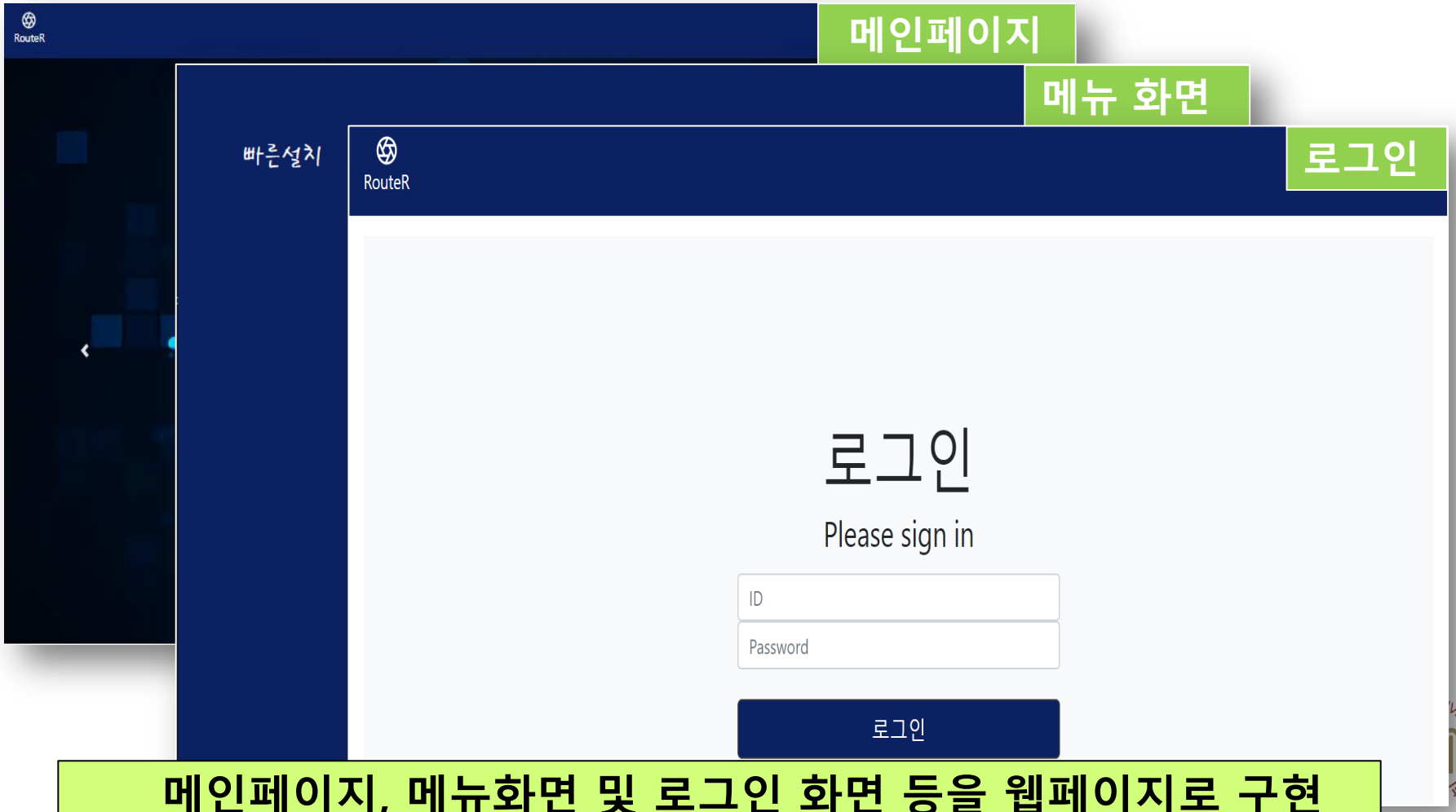
네트워크 인터페이스 설정

```
void init_dev(char *dev_name)
{
    char err_buf[ERRBUF_SIZE];

    use_dev = pcap_open_live(dev_name, SNAPLEN, 1, 1000, err_buf);
    if(use_dev == NULL)
    {
        err_print(1);
        exit(1);
    }

    cout << "[*] DEVICE SETTING SUCCESS" << endl;
```

ARP spoofing 작업을 위한 과정을 프로그래밍

개발 시스템 운영(1/4)

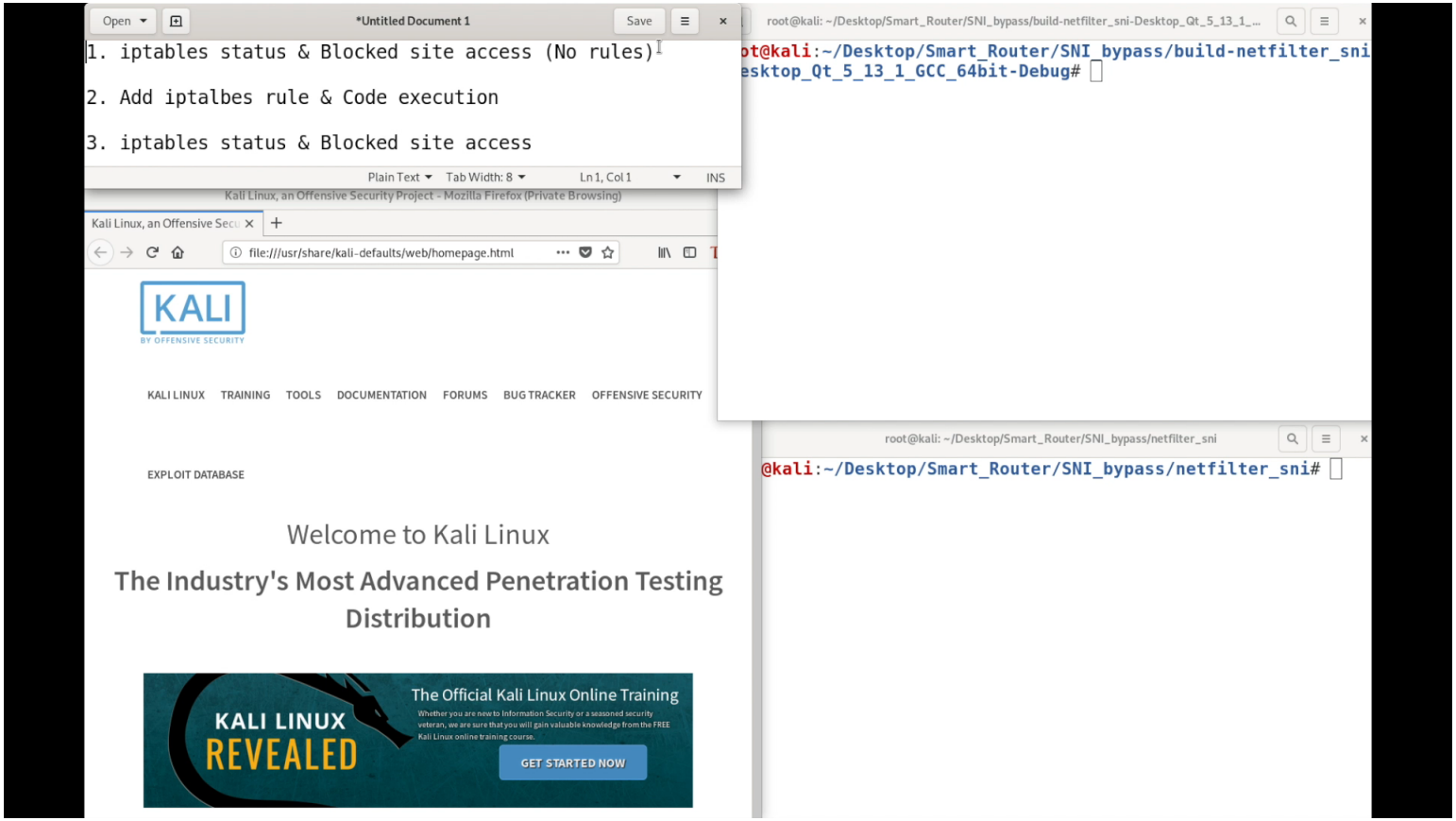
기본 화면



메인페이지, 메뉴화면 및 로그인 화면 등을 웹페이지로 구현

개발 시스템 운영(2/4)

SNI 차단 우회 시연



개발 시스템 운영(3/4)

ARP Spoofing 차단

ARP 패킷 전송 성공

```
pyozzi01-1:~ pyozzi$ ./ARP_Broadcast
[*] FIND MAC: 3c:22:fb:93:9b:60
[*]
[*]
[*]
[*]
[*]
[*]
[*]
[*]
[*]
[*]
[*]
```

프로그램에서 요청한 ARP

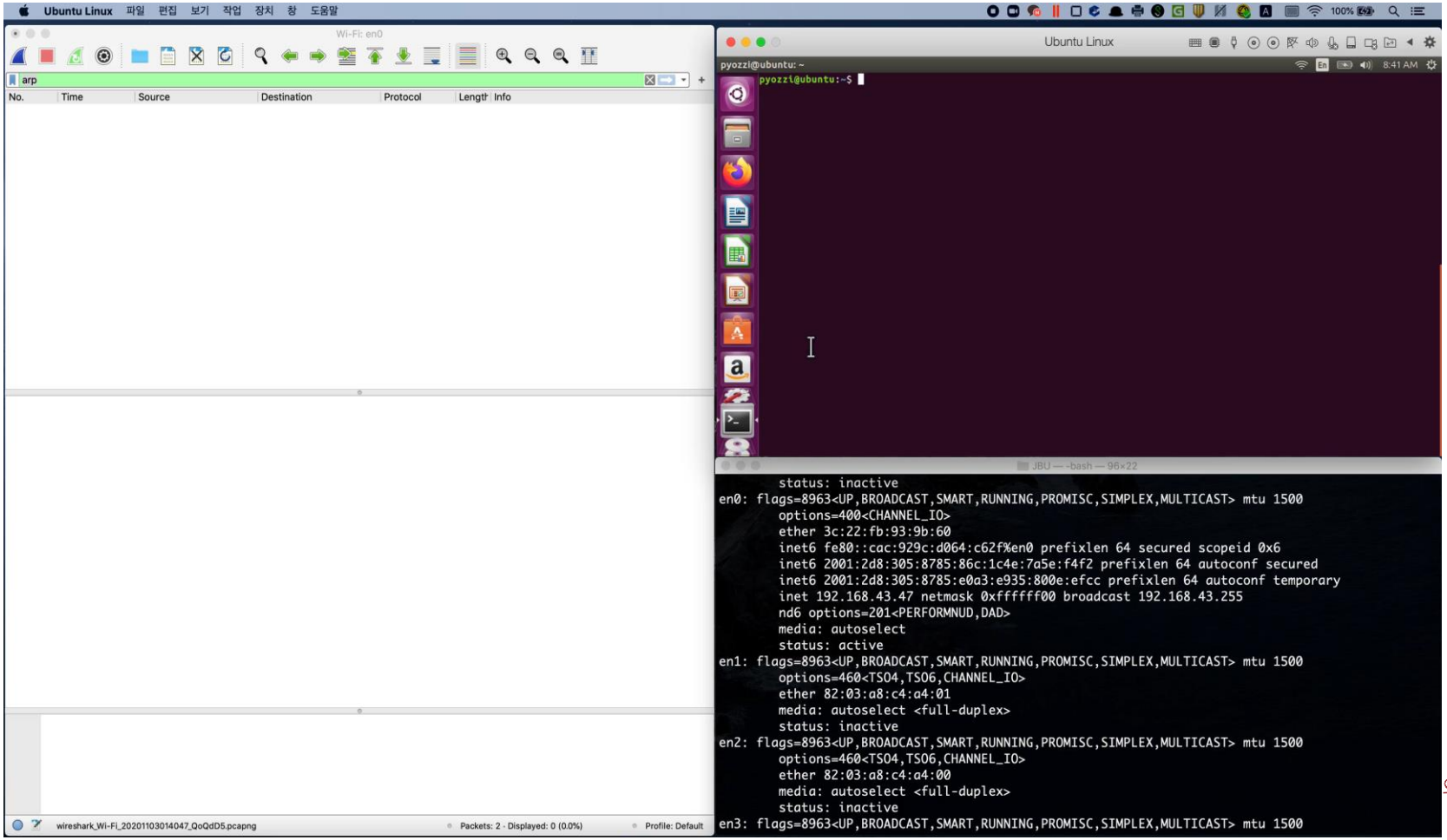
arp

No.	Ti	Source	Destination	Protocol
70	…	IETF-VRRP-VRID_32	Apple_93:9b:60	ARP
71	…	IETF-VRRP-VRID_32	Apple_93:9b:60	ARP
138	…	IETF-VRRP-VRID_32	Apple_93:9b:60	ARP
139	…	IETF-VRRP-VRID_32	Apple_93:9b:60	ARP
235	…	IETF-VRRP-VRID_32	Apple_93:9b:60	ARP
236	…	IETF-VRRP-VRID_32	Apple_93:9b:60	ARP
2139	…	RuckusWi_1f:d8:7f	Apple_93:9b:60	ARP
2140	…	RuckusWi_1f:d8:7f	Apple_93:9b:60	ARP
2752	…	BrocadeC_f6:4a:b4	Apple_93:9b:60	ARP
2753	…	BrocadeC_f6:4a:b4	Apple_93:9b:60	ARP
3085	…	BrocadeC_f6:f0:b4	Apple_93:9b:60	ARP
3289	…	RuckusWi_8e:23:b9	Apple_93:9b:60	ARP
3290	…	RuckusWi_8e:23:b9	Apple_93:9b:60	ARP
5476	…	Apple_9c:17:91	Apple_93:9b:60	ARP
5477	…	Apple_9c:17:91	Apple_93:9b:60	ARP
5879	…	RuckusWi_1f:d8:7f	Apple_93:9b:60	ARP

ARP Spoofing 차단을 성공적으로 완료

개발 시스템 운영(4/4)

ARP Spoofing 차단시연



결론 및 기대 효과

결 론

- 개발한 2020 Smart RouteR는 기존의 공유기보다 업그레이드된 보안성과 편리한 기능을 구현
- 특히 웹 인터페이스를 제공함으로써 웹에서 공유기를 편리하게 제어할 수 있도록 기능을 향상

기대 효과

- 이 프로젝트를 통해 네트워크의 취약점에 대해 연구할 수 있었고 네트워크, 웹 개발 등을 통해 이 부분의 전문지식을 습득하고 기술 역량을 배양
- 전체적인 네트워크 보안이 한 단계 더 발전하는 긍정적인 효과 기대. 끝.

Q & A

감사합니다