신원기반 네트워크 패킷 접근제어 시스템 개발

김현진, 조재현, 허송이

01

프로젝트 개요

1. 주제
2. 시스템 구성도

02

프로젝트 내용

1. 설계
2. 개발
3. 시연

03

프로젝트 결과

1. 논문
2. 수상

01 프로젝트 개요

1. 주제

신원기반 네트워크 패킷 접근제어 시스템 개발

지문인식을 통한
사용자 인증 기능

+

인증 실패 로그 확인,
사용자 등록 등
패킷 및 사용자 관리 기능

01 프로젝트 소개

2. 시스템 구성도

외부망

내부망

02 프로젝트 내용

1. 설계

1. DB 설계

2. C/R 인증 자체 프로토콜 설계

3. 보안 요소

02 프로젝트 내용

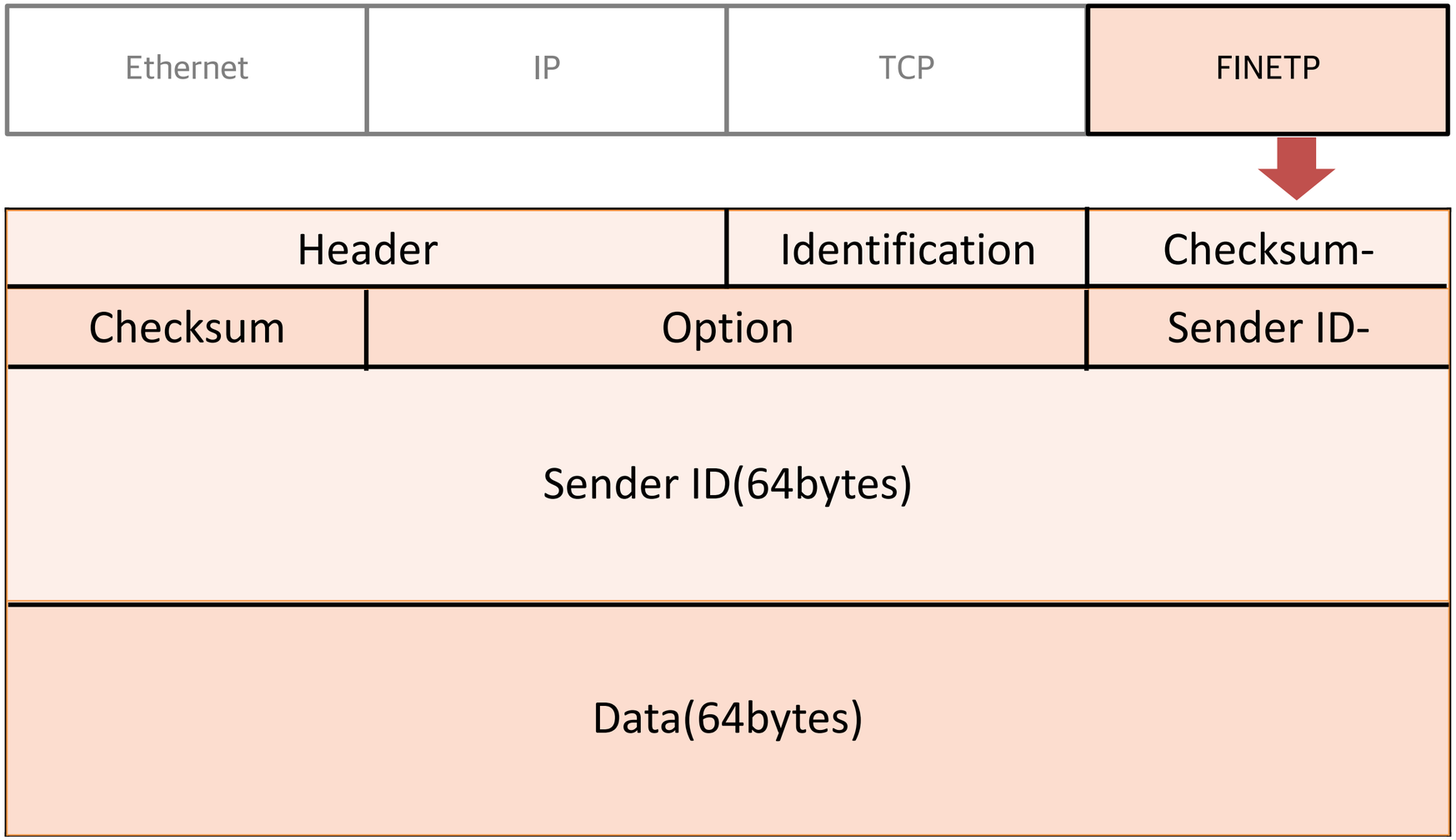
1. 설계

DB 설계

```
MariaDB [finetdb]> select * from member_tb;
+------------+------------------------------------------------------------------+------------------------------------------------------------------+-------------+------------------------------------------------------------------+
| member_idx | member_id                                                        | member_guid                                                      | member_subid | member_subpw                                                     |
+------------+------------------------------------------------------------------+------------------------------------------------------------------+-------------+------------------------------------------------------------------+
|          1 | 62492fe30f8558566b6a2288fbae2f82cfde8d4a9d814125c3081f44f691724f | 1b89ef6d0da66e0d222547c223a2023efe58f76a1b7686acf426585684982da3 | finet        | 404872c562b48cc94b4db5f9c7bdaf4260ac98316672284d8d4d7c04befade82 |
|          2 | c526a6ab42912dbdfc8edf54a399b1c92767141da8802ee9237e7e8c7a2c2fb1 | c6a39891197a579aef09a0fe72f669d2bab9807203a7fdeda7c16423182b9a8e | jaehyeon     | 63403912b98b029748d39be2c1c2ba87f3942dd4f0b5ae6377215fd7f592874e |
+------------+------------------------------------------------------------------+------------------------------------------------------------------+-------------+------------------------------------------------------------------+
```

		AUTO_INCREMENT	
member_id	varchar(64)	UNIQ, NOT NULL	Client MAC Addr (SHA-256)
member_guid	varchar(64)	UNIQ, NOT NULL	Client GUID (SHA-256)
member_subid	varchar(20)	UNIQ, NOT NULL	Client ID (SHA-256)
member_subpw	varchar(64)	UNIQ, NOT NULL	Client PW (SHA-256)

C/R 인증 프로토콜 설계


Ethernet
IP
TCP
FINETP
Header
Identification
Checksum-
Checksum
Option
Sender ID-
Sender ID(64bytes)
Data(64bytes)

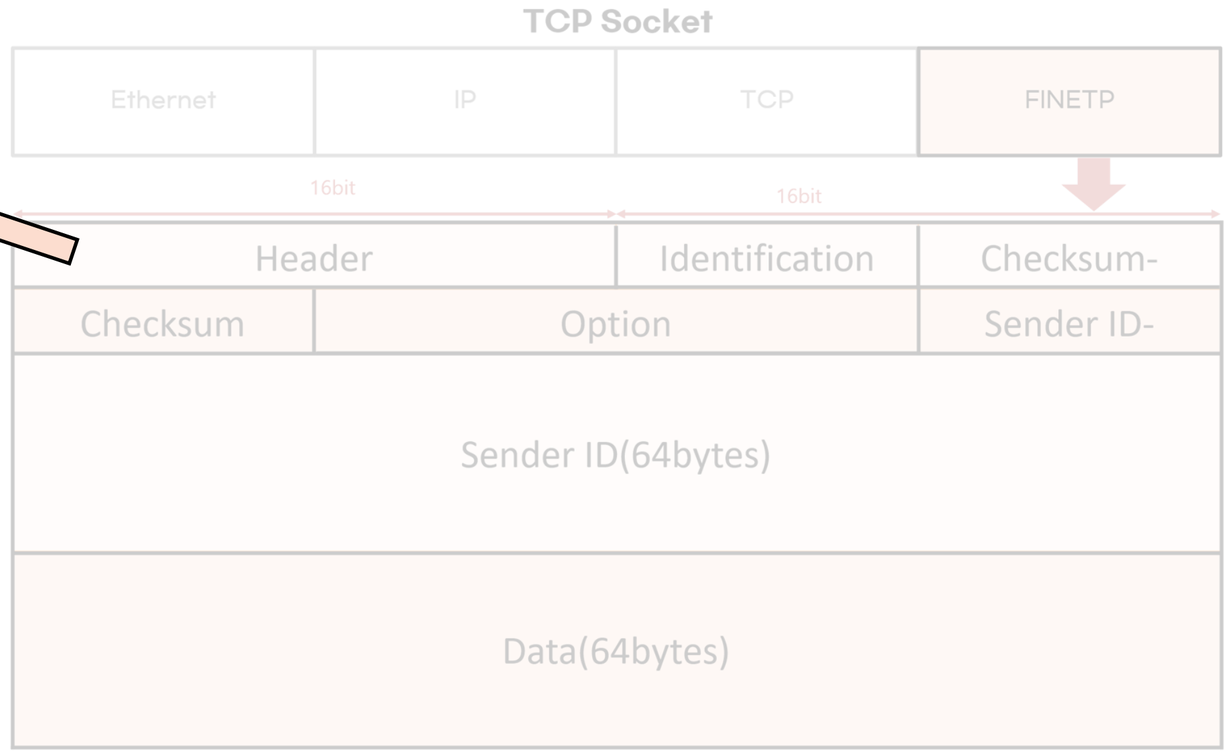
C/R 인증 프로토콜 설계

Header (2 bytes)

- 0xF1E7

Identification (1 bytes)

- 0x01 (Authentication Request)
- 0x02 (Challenge-Response Challenge)
- 0x03 (Challenge-Response Response)
- 0x04 (Authentication Response)



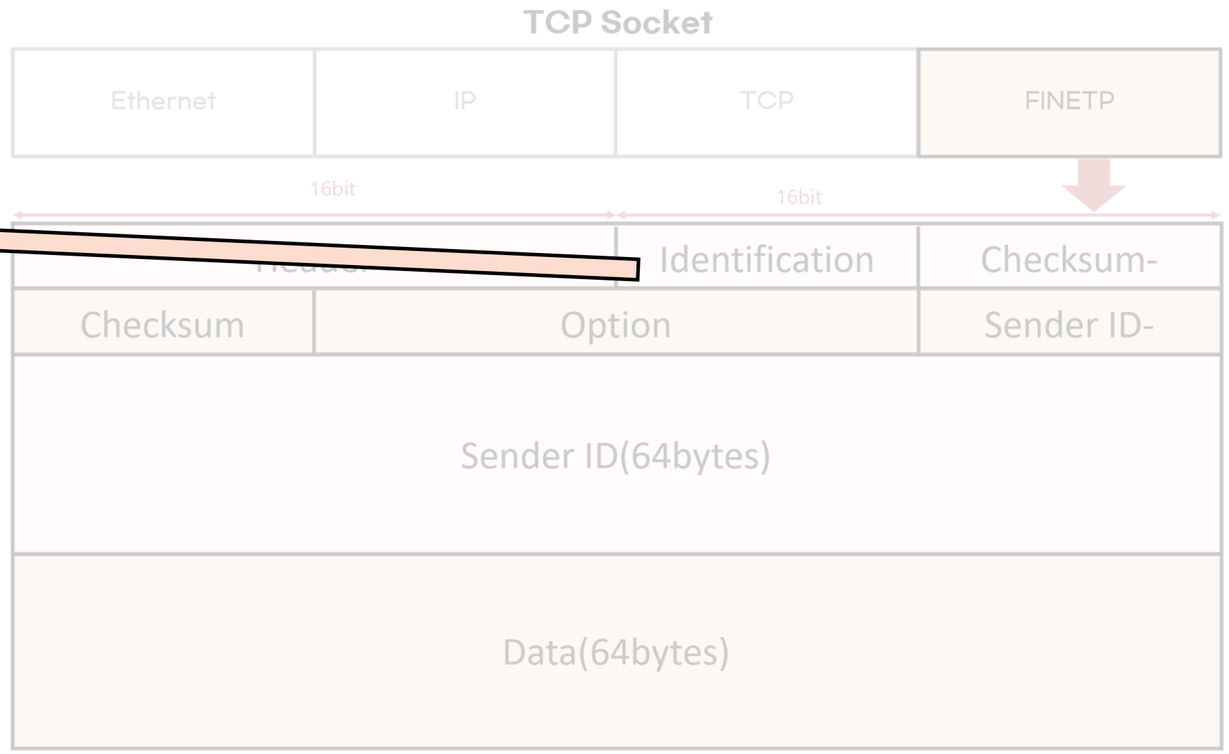
C/R 인증 프로토콜 설계

Header (2 bytes)

- 0xF1E7

Identification (1 bytes)

- 0x01 (Authentication Request)
- 0x02 (Challenge-Response Challenge)
- 0x03 (Challenge-Response Response)
- 0x04 (Authentication Response)



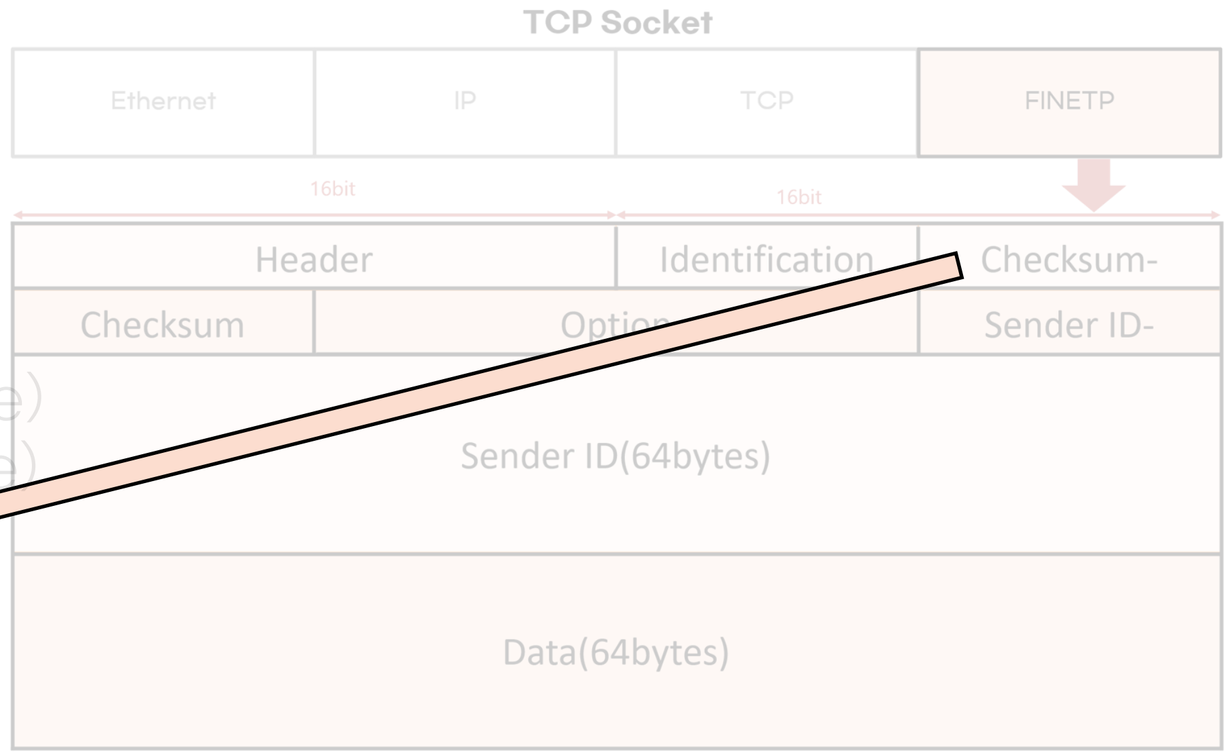
C/R 인증 프로토콜 설계

Header (2 bytes)

- 0xF1E7

Identification (1 bytes)

- 0x01 (Authentication Request)
- 0x02 (Challenge-Response Challenge)
- 0x03 (Challenge-Response Response)
- 0x04 (Authentication Response)



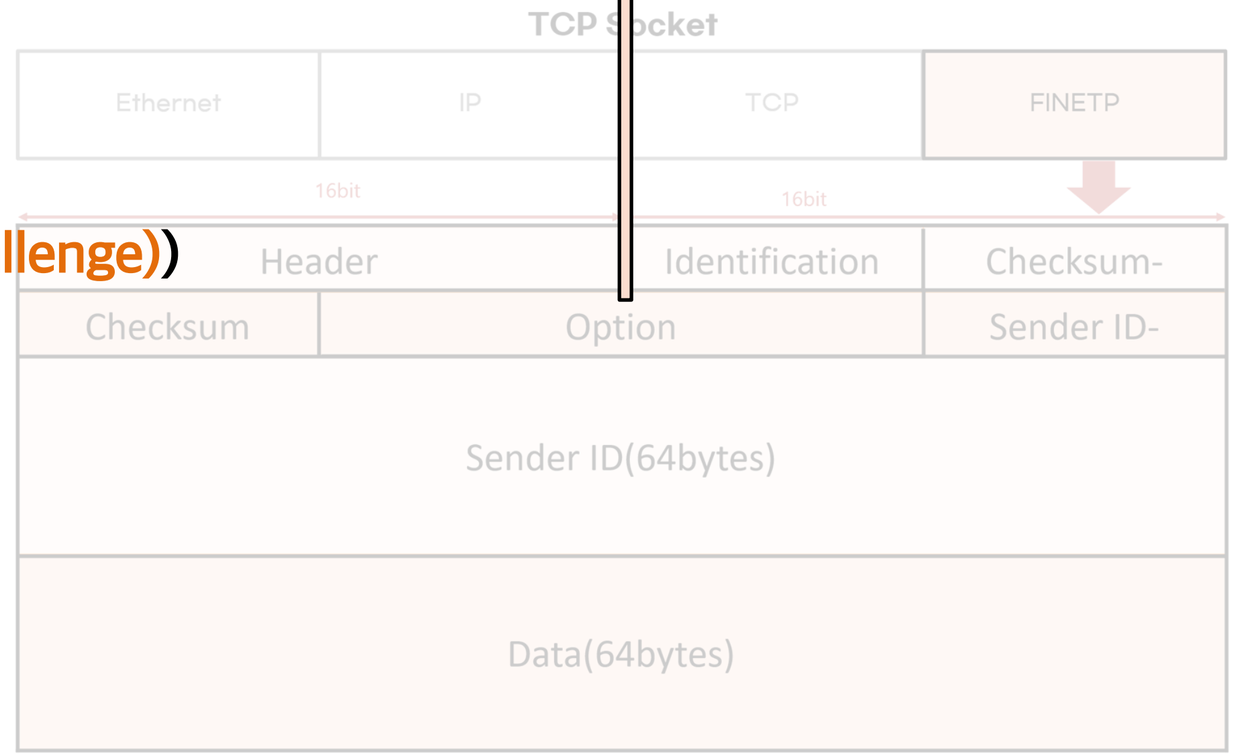
Checksum (2 bytes)

- ~(FINETP[0]+…+FINETP[134](Except [3],[4]))&0x0000FFFF

C/R 인증 프로토콜 설계

Option (2 bytes)

- 0x0001 (Success, When ID(0x03) FIDO Auth)
- 0x0002 (Fail, When ID(0x03) ID/PW Auth)
- 0xXXXX (When ID(0x02) Random Number(Challenge))



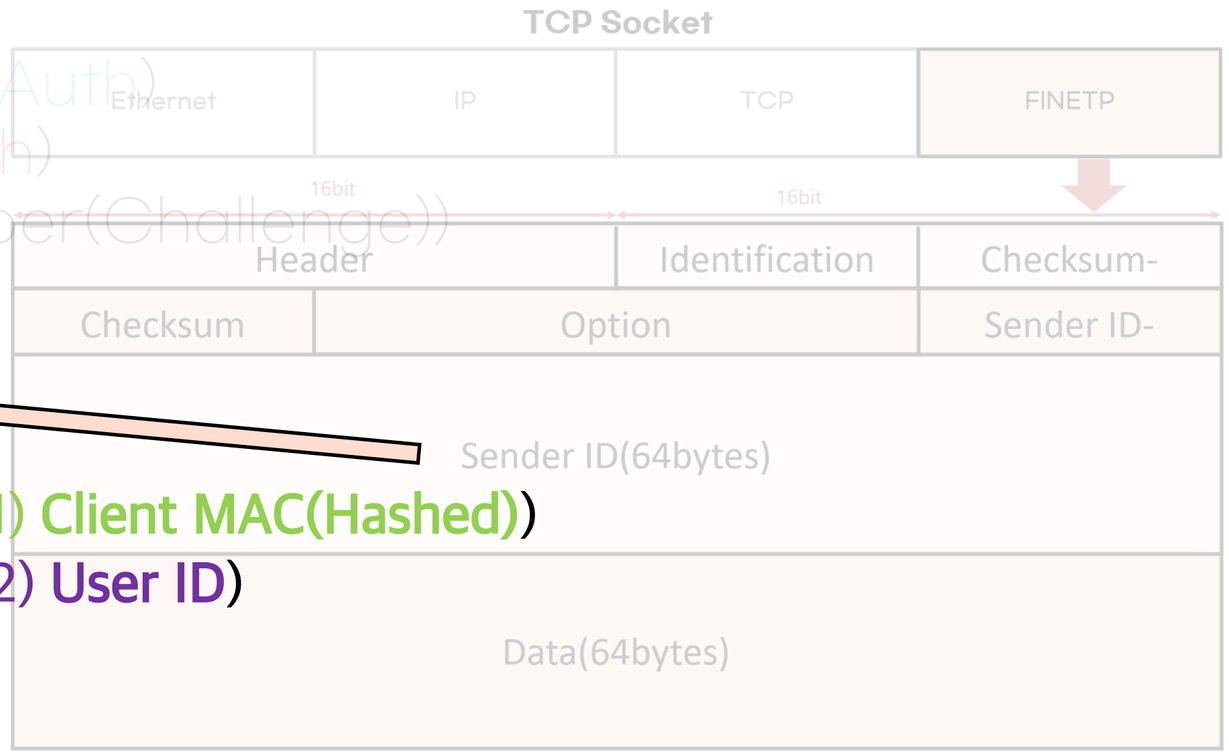
C/R 인증 프로토콜 설계

Option (2 bytes)

- 0x0001 (Success, When ID(0x03) FIDO Auth)
- 0x0002 (Fail, When ID(0x03) ID/PW Auth)
- 0xXXXX (When ID(0x02) Random Number(Challenge))

Sender ID (64 bytes)

- 0x0000…0000 (Padding)
- 0xXXXX…XXXX (When ID(0x03)/Option(0x0001) **Client MAC(Hashed)**)
- 0xXXXX…XXXX (When ID(0x03)/Option(0x0002) **User ID**)



02 프로젝트 내용

1. 설계

C/R 인증 프로토콜 설계

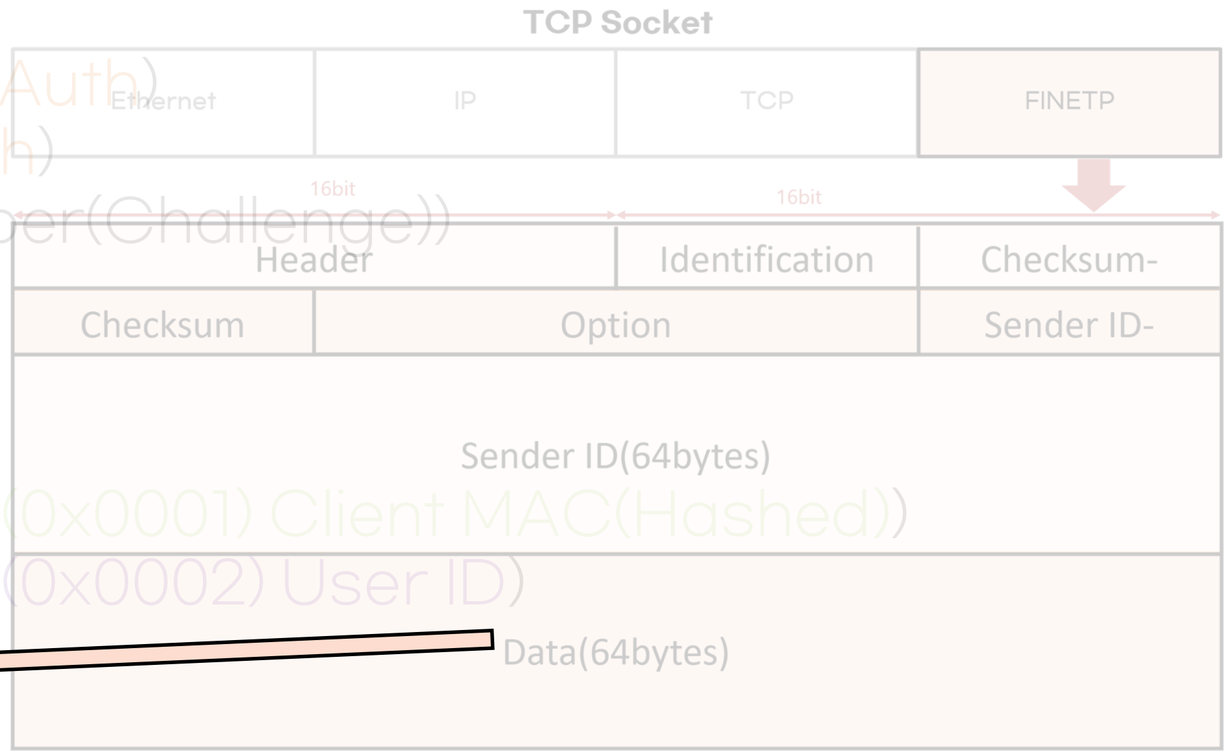
Option (2 bytes)

- 0x0001 (Success, When ID(0x03) FIDO Auth)
- 0x0002 (Fail, When ID(0x03) ID/PW Auth)
- 0xXXXX (When ID(0x02) Random Number(Challenge))

Sender ID (64 bytes)

- 0x0000…0000 (Padding)
- 0xXXXX…XXXX (When ID(0x03)/Option(0x0001) Client MAC(Hashed))
- 0xXXXX…XXXX (When ID(0x03)/Option(0x0002) User ID)

Data (64 bytes)



- 0x0000…0000 (Padding)
- 0xXXXX…XXXX (When ID(0x03)/Option(0x0001) GUID + Challenge(Hashed))
- 0xXXXX…XXXX (When ID(0x03)/Option(0x0002) User PW + Challenge(Hashed))

02 프로젝트 내용

1. 설계

C/R 인증 프로토콜 설계

Authentication Request(1)

Challenge-Response Challenge(2)

Challenge-Response Response(3)

Authentication Response(4)

Option (2 bytes)

- 0x0001 (Success, When ID(0x03) FIDO Auth)
- 0x0002 (Fail, When ID(0x03) ID/PW Auth)
- 0xXXXX (When ID(0x02) Random Number(Challenge))

초기 인증 성공 후 요청

Header	Identification	Checksum	Option	SenderID	Data
0xF1E7	0x01	0x53ab	0x0001	0x000…	0x000…

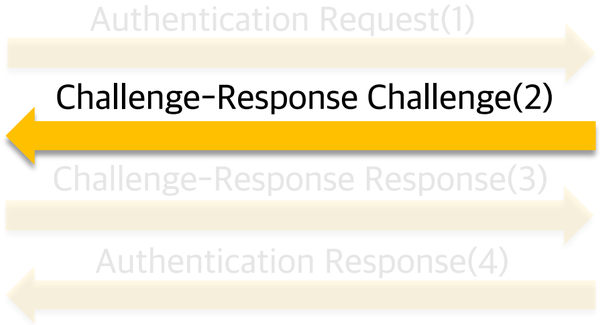
초기 인증 실패 후 요청

Header	Identification	Checksum	Option	SenderID	Data
0xF1E7	0x01	0x6f7c	0x0002	0x000…	0x000…

02 프로젝트 내용

1. 설계

C/R 인증 프로토콜 설계



Option (2 bytes)

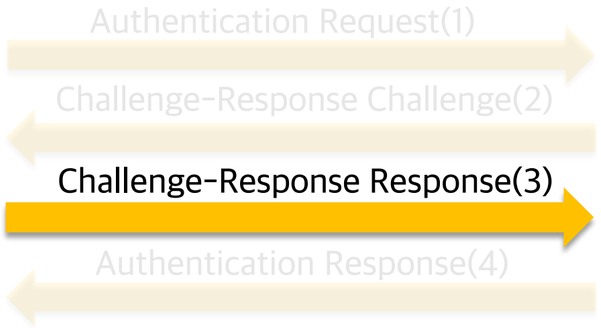
- 0x0001 (Success, When ID(0x03) FIDO Auth)
- 0x0002 (Fail, When ID(0x03) ID/PW Auth)
- 0xXXXX (When ID(0x02) Random Number(Challenge))

Header	Identification	Checksum	Option	SenderID	Data
0xF1E7	0x02	0x1bac	0x7b31	0x000…	0x000…

02 프로젝트 내용

1. 설계

C/R 인증 프로토콜 설계



Option (2 bytes)

- 0x0001 (When ID(0x03) FIDO Auth)
- 0x0002 (When ID(0x03) ID/PW Auth)

Sender ID (64 bytes)

- When ID(0x03)/Option(0x0001) Client MAC(Hashed)
- When ID(0x03)/Option(0x0002) User ID

Data (64 bytes)

- When ID(0x03)/Option(0x0001) GUID + Challenge(Hashed)
- When ID(0x03)/Option(0x0002) User PW + Challenge(Hashed)

FIDO 방식 선택

Header	Identification	Checksum	Option	SenderID	Data
0xF1E7	0x03	0x7dfe	0x0001	0xb22a…	0xfe19…

ID/PW 방식 선택

Header	Identification	Checksum	Option	SenderID	Data
0xF1E7	0x03	0x32a7	0x0002	test12	0x3b51…

02 프로젝트 내용

1. 설계

C/R 인증 프로토콜 설계

Authentication Request(1)

Challenge-Response Challenge(2)

Challenge-Response Response(3)

Authentication Response(4)

Option (2 bytes)

- 0x0001 (Success, When ID(0x03) FIDO Auth)
- 0x0002 (Fail, When ID(0x03) ID/PW Auth)
- 0xXXXX (When ID(0x02) Random Number(Challenge))

C/Response 인증 성공

Header	Identification	Checksum	Option	SenderID	Data
0xF1E7	0x04	0x193b	0x0001	0x000…	0x000…

C/Response 인증 실패

Header	Identification	Checksum	Option	SenderID	Data
0xF1E7	0x04	0xc471	0x0002	0x000…	0x000…

보안 요소

① 패킷 검증

Header
0xF1E7

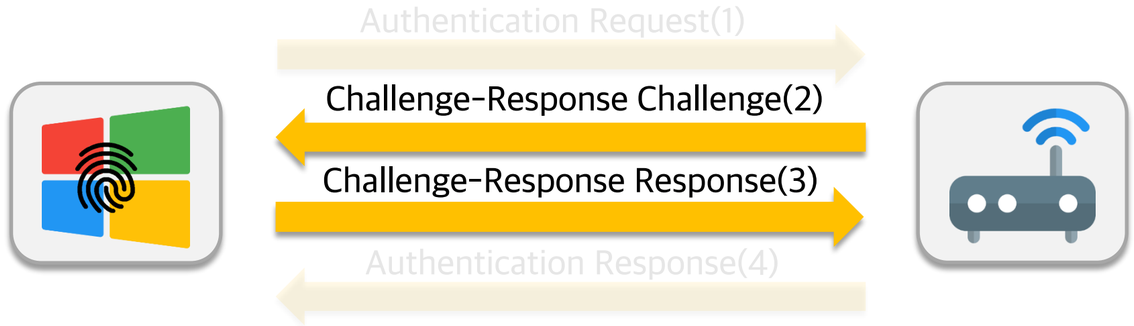
Checksum
0x193b

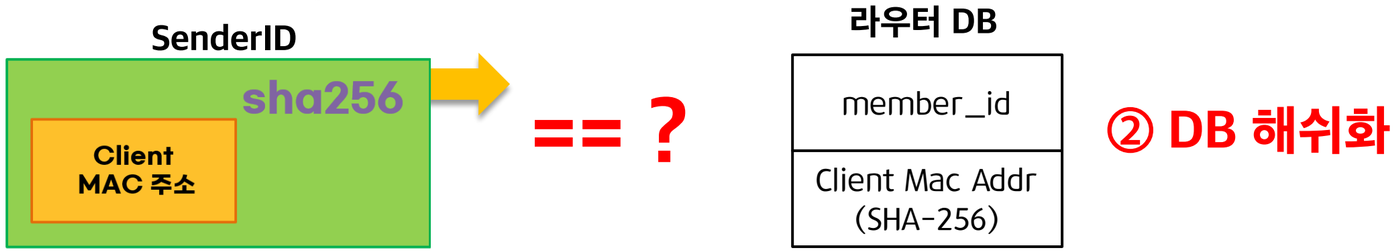
Identification
0x04

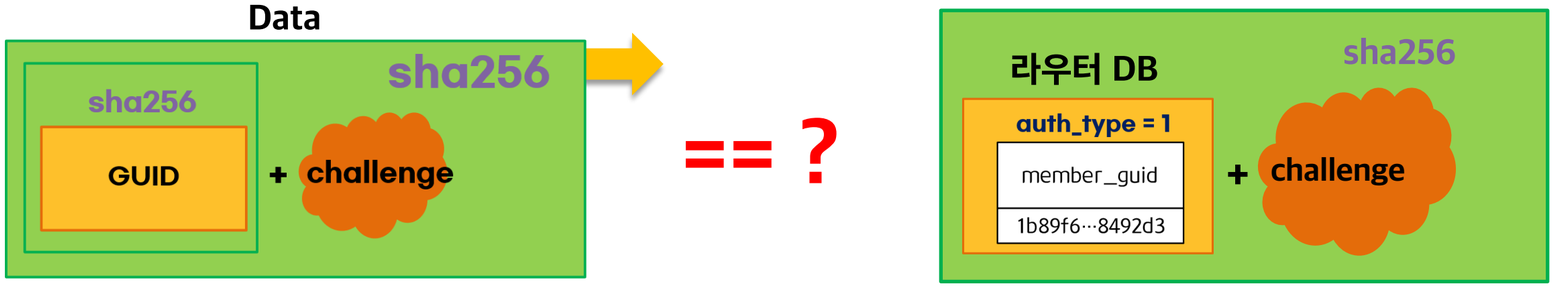
Option
0x0001

보안 요소

③ Challenge-Response 검증







보안 요소

④ 로깅

0 (ERR_NOERR) : 성공 (에러 없음)

1 (ERR_FLOW) : 플로우 에러

2 (ERR_CHKSUM) : 체크섬 에러

3 (ERR_OPTION) : 옵션 에러

4 (ERR_SENID) : SenID 에러

5 (ERR_DATA) : Data 에러

6 (ERR_FSCHK) : 초기인증에러

7 (ERR_CRCHK) : C/R 인증 에러

```
⚡ root@kali /var/log cd /var/log/
⚡ root@kali /var/log ls -l finet.log
-rw-r--r-- 1 root root 1296 May 18 11:48 finet.log
⚡ root@kali /var/log cat finet.log
172.31.13.132-2021/04/14-15:23:27-0
172.31.13.132-2021/04/14-15:23:42-0
172.31.13.132-2021/04/14-15:30:39-0
172.31.13.132-2021/04/14-15:30:50-7
172.31.13.132-2021/04/14-15:33:46-7
172.31.13.132-2021/04/14-15:33:58-0
172.31.13.132-2021/04/14-15:34:07-7
172.31.13.132-2021/04/14-15:34:58-3
172.31.13.132-2021/04/14-15:35:40-1
172.31.13.132-2021/04/14-15:36:17-2
172.31.13.132-2021/04/14-15:37:08-6
172.31.13.132-2021/04/26-16:29:47-2
172.31.13.132-2021/05/10-16:29:47-1
172.31.13.132-2021/05/11-16:29:47-3
172.31.13.132-2021/05/12-16:29:47-0
```

보안 요소

⑤ 관리자 페이지에서의 패킷 관제 및 사용자 등록

패킷 관제

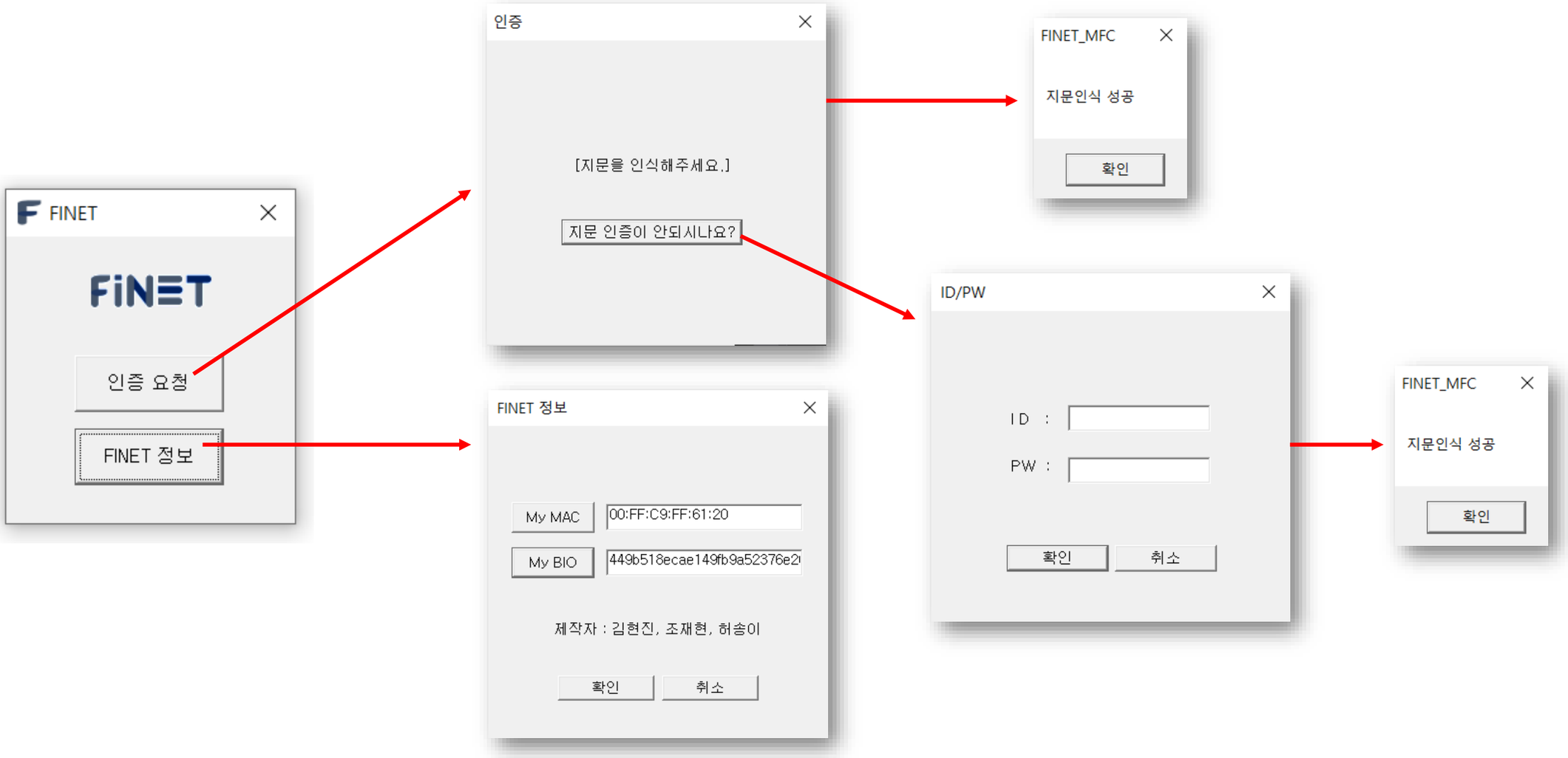
로그 분석하여 이상한 움직임 감지

사용자 등록

관리자 페이지에서 사용자의 MAC, guid 직접 등록

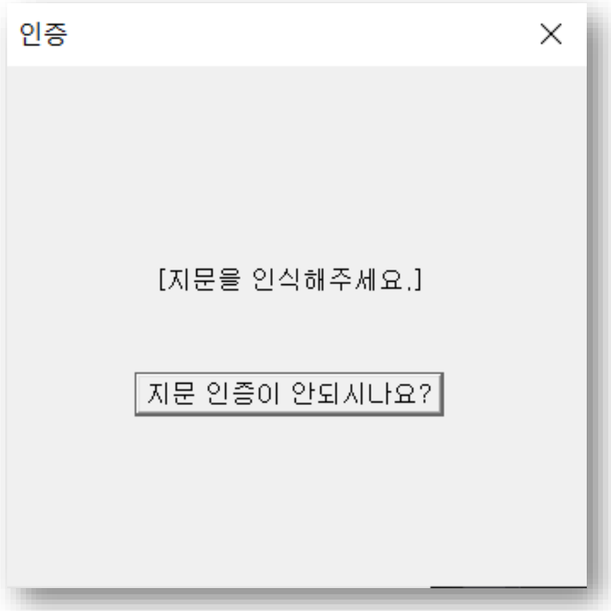
02 프로젝트 내용

2. 개발



02 프로젝트 내용

2. 개발



지문인증, ID/PW

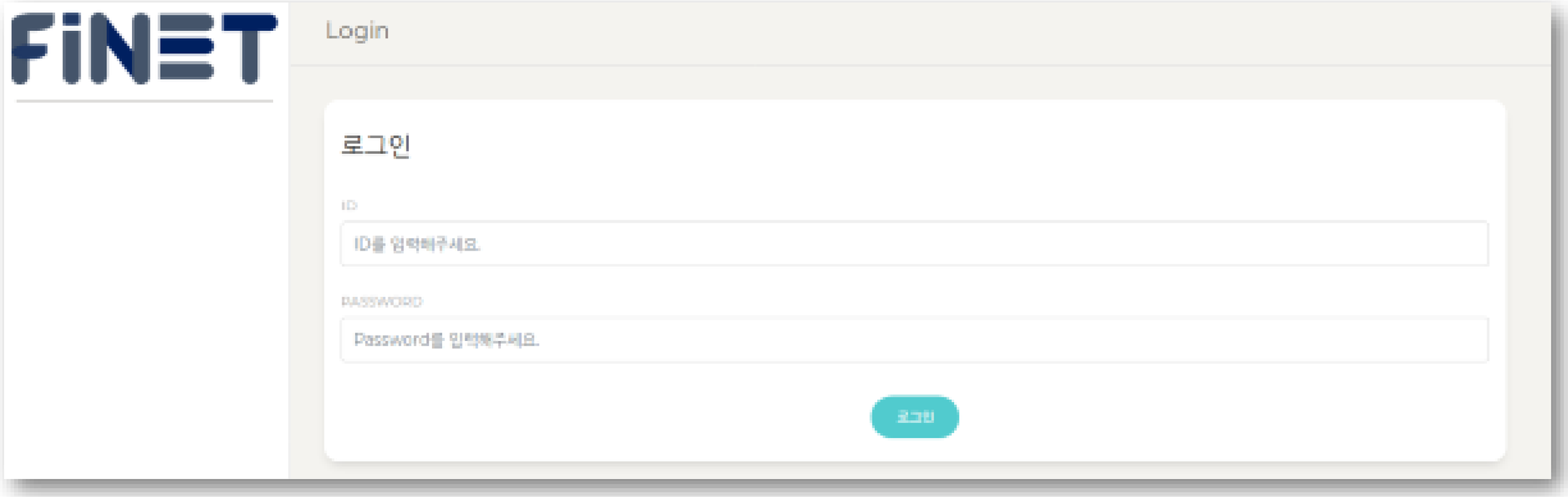
3번 이상 실패시

계정 잠금

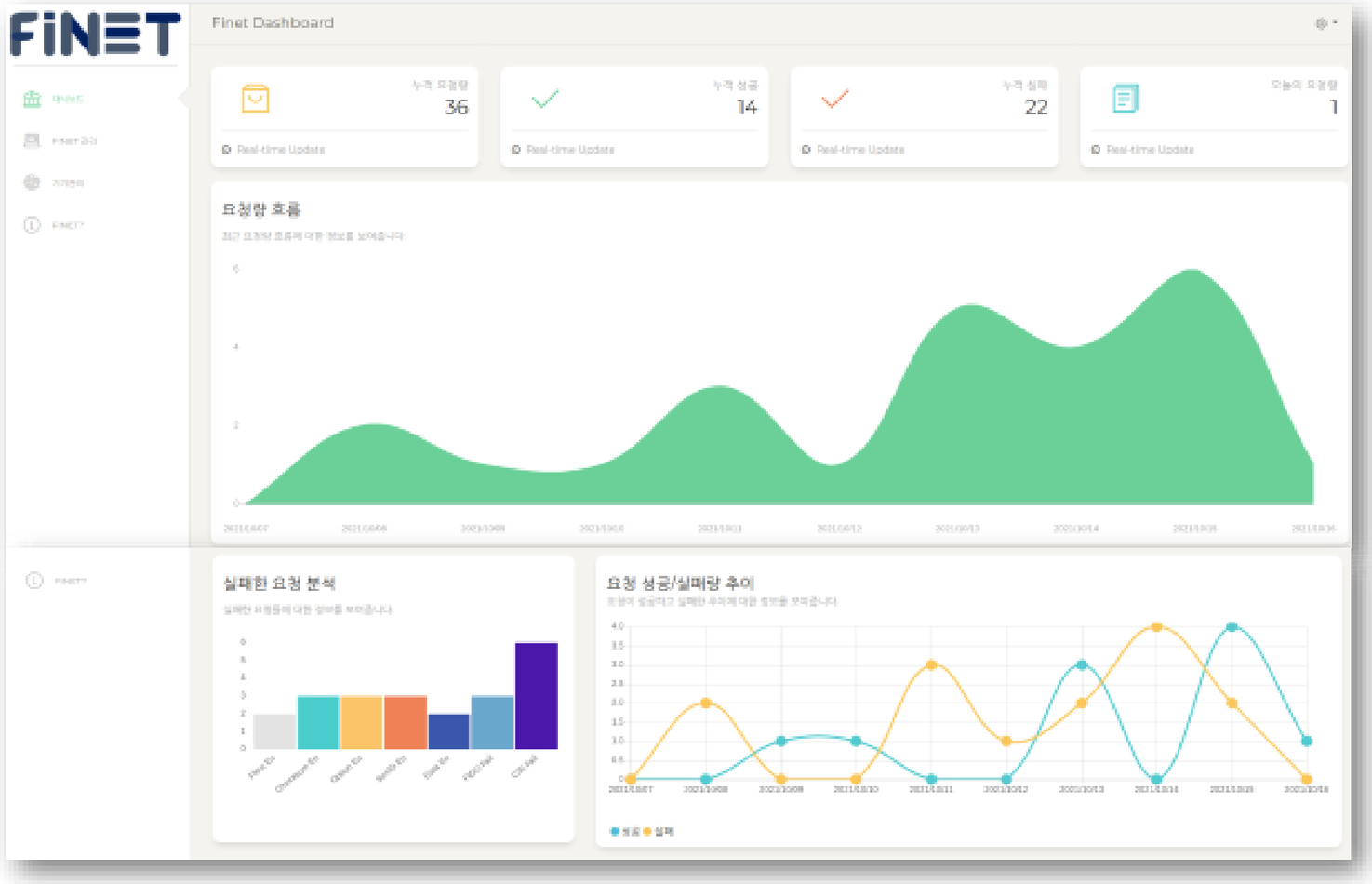
02 프로젝트 내용

2. 개발

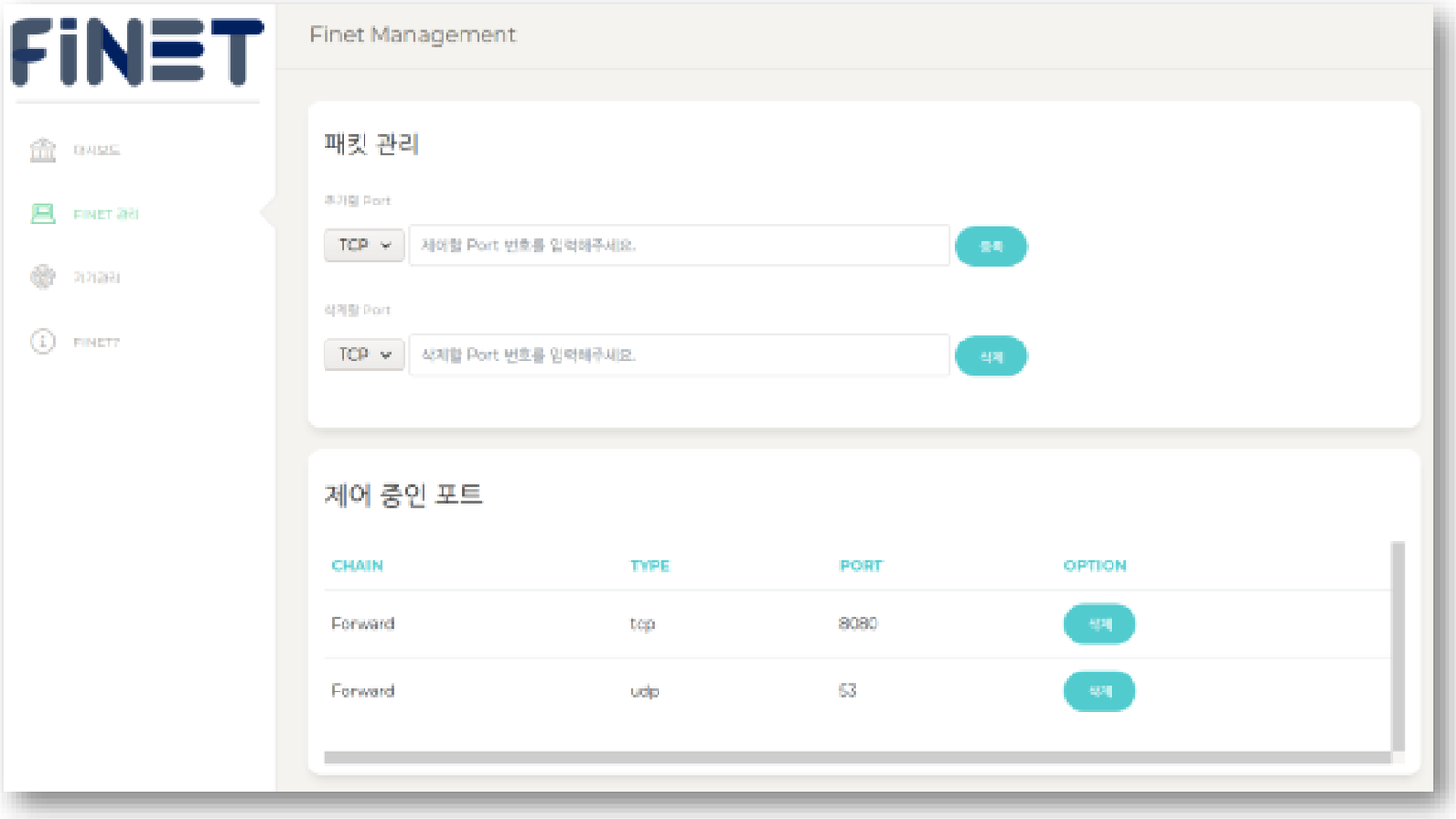
① 관리자 로그인 페이지



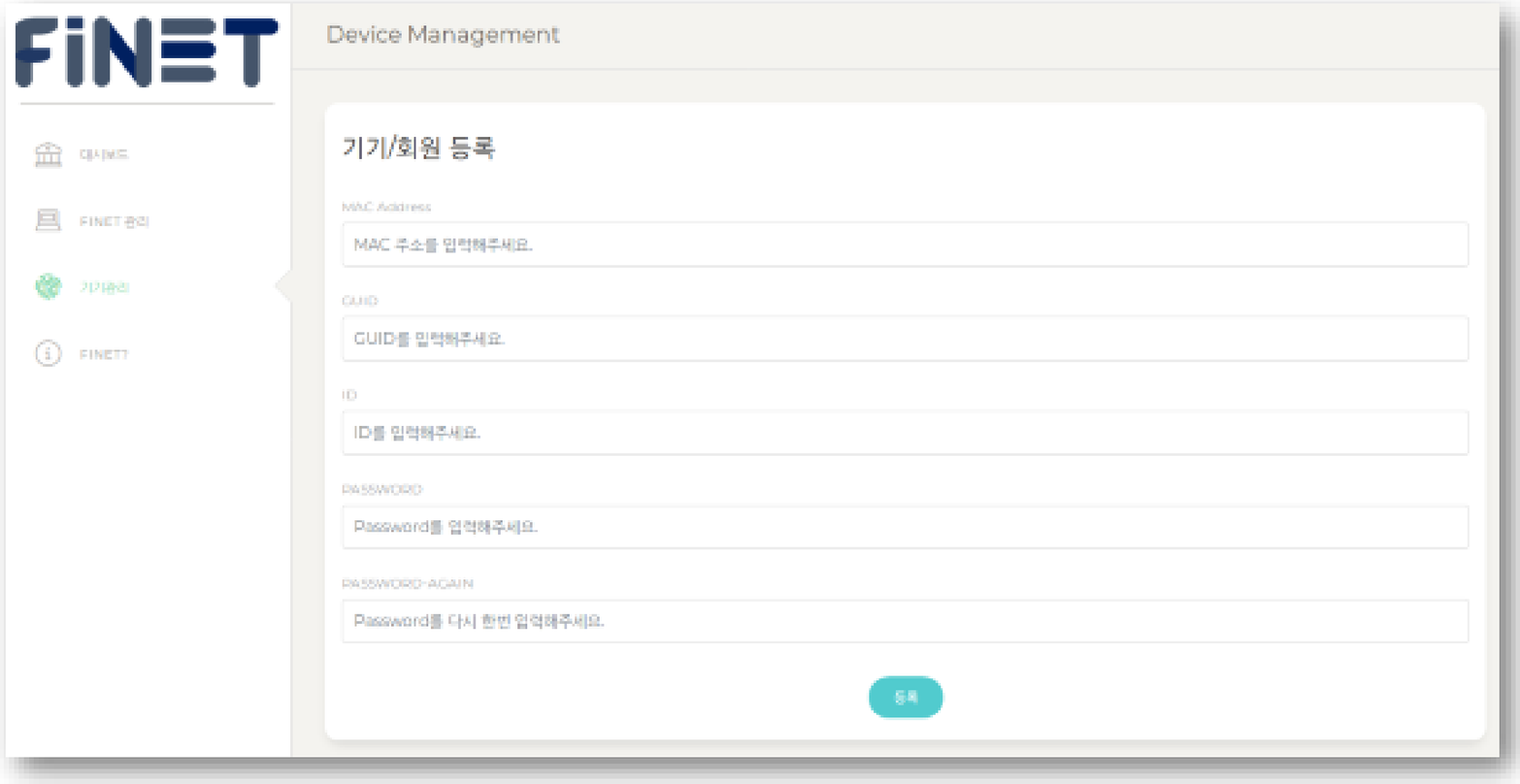
② 요청량 및 실패로그 등 모니터링 페이지



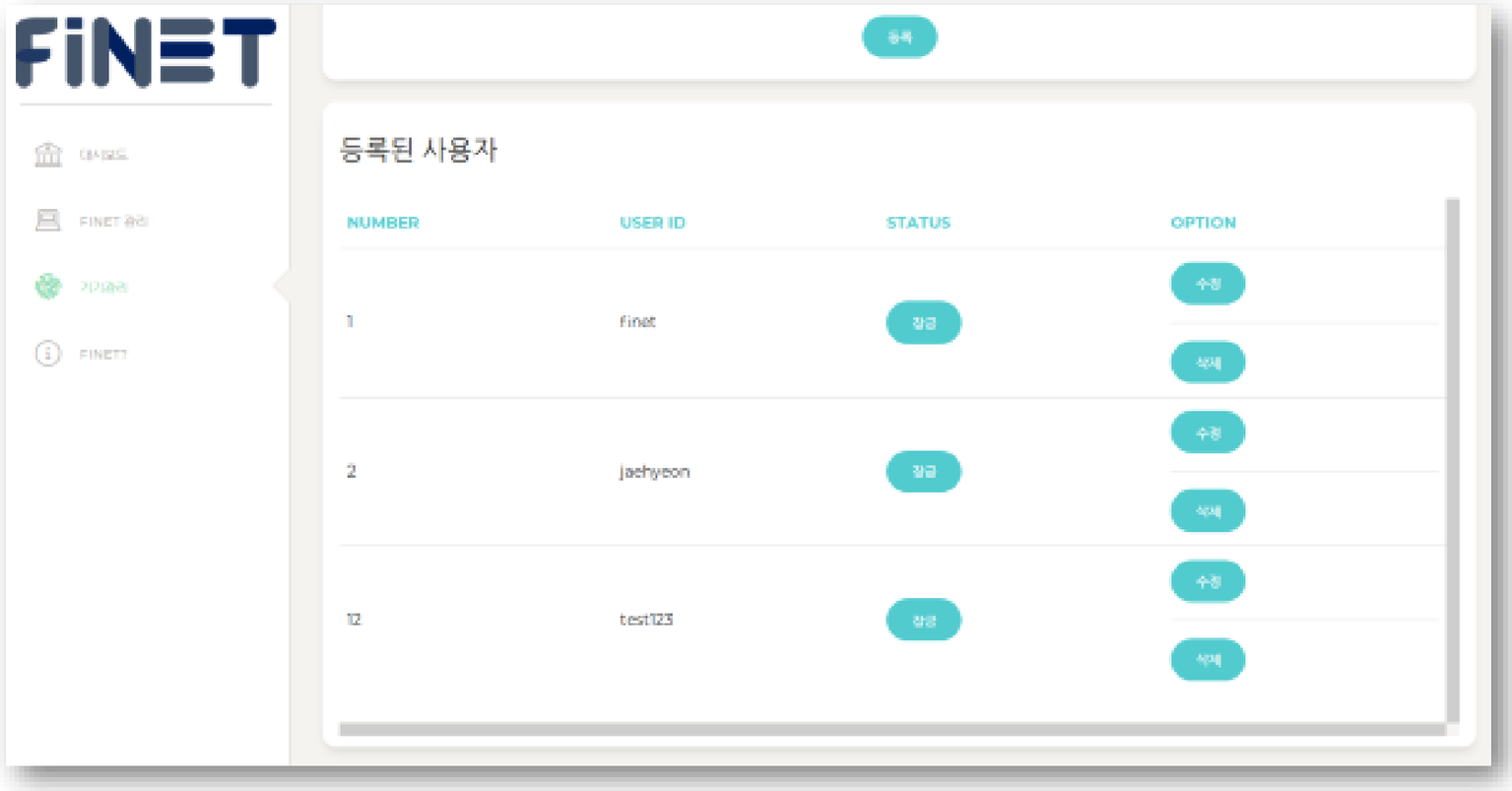
③ 포트 관리 페이지


FINET
Finet Management
대시보드
FINET 관리
기기관리
FINET?
패킷 관리
추가할 Port
TCP
제어할 Port 번호를 입력해주세요.
등록
삭제할 Port
TCP
삭제할 Port 번호를 입력해주세요.
삭제
제어 중인 포트
CHAIN
TYPE
PORT
OPTION
Forward
tcp
8080
삭제
Forward
udp
53
삭제

④ 기기 등록 페이지


FiNET
Device Management
기기/회원 등록
MAC Address
MAC 주소를 입력해주세요.
GUID
GUID를 입력해주세요.
ID
ID를 입력해주세요.
PASSWORD
Password를 입력해주세요.
PASSWORD-AGAIN
Password를 다시 한번 입력해주세요.
등록

⑤ 사용자 관리 페이지


FiNET
등록
등록된 사용자
NUMBER
USER ID
STATUS
OPTION
1
finet
수정
삭제
2
jaehyeon
수정
삭제
12
test123
수정
삭제

⑥ FINET 프로그램 소개 페이지

02 프로젝트 내용

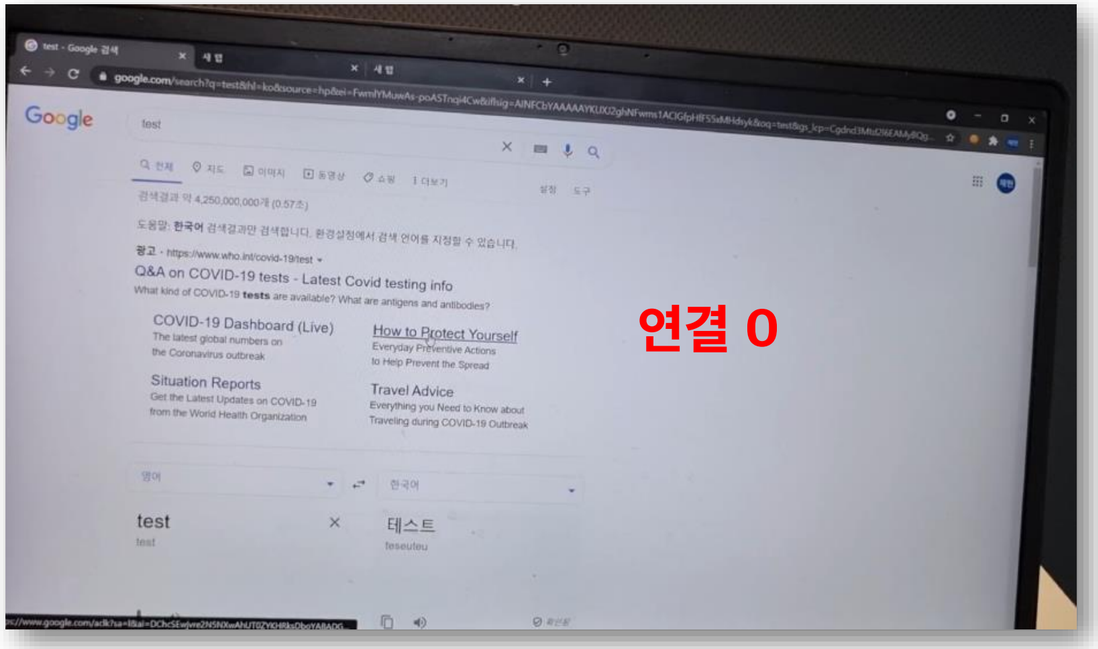
3. 시연

왼쪽: 라즈베리파이, 오른쪽: 사용자 pc 환경

02 프로젝트 내용

3. 시연

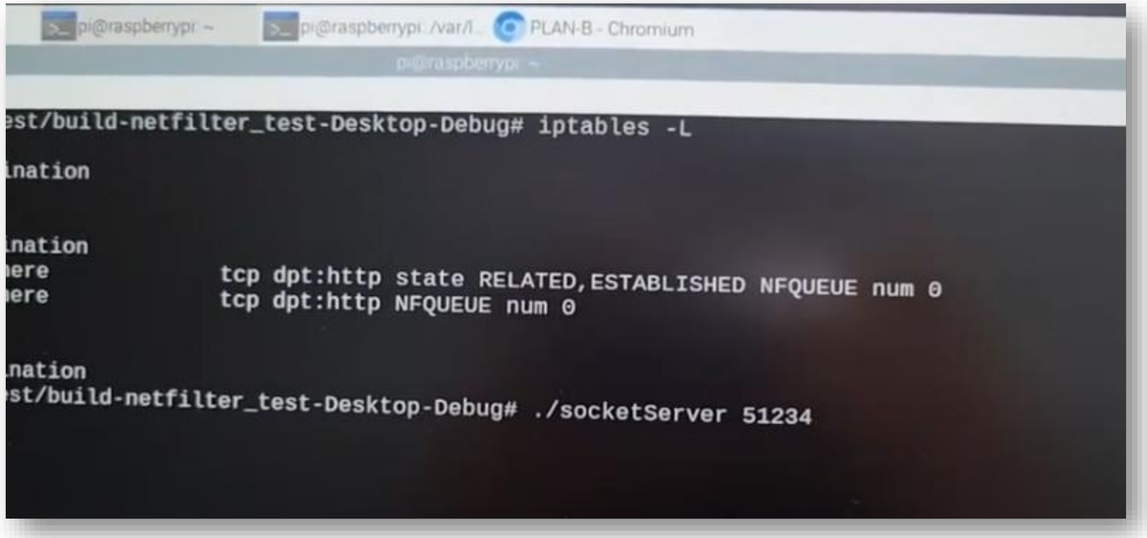
사용자 pc FINET-AP에 연결



02 프로젝트 내용

3. 시연

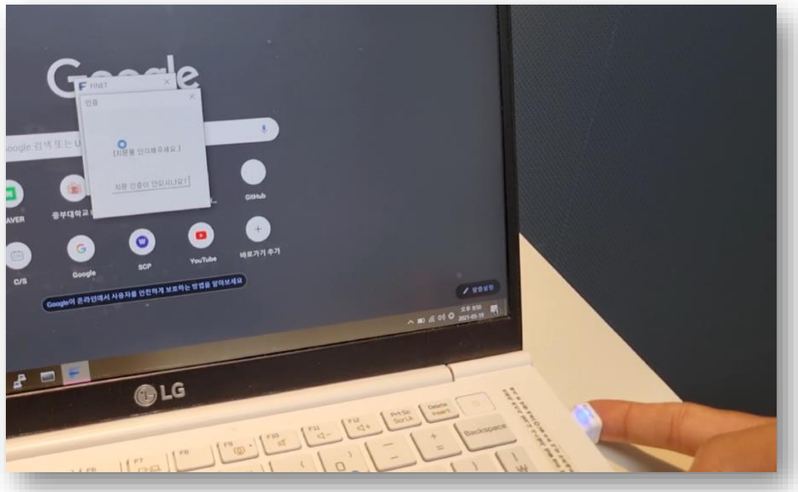
방화벽 설정을 통해 80번 포트 차단

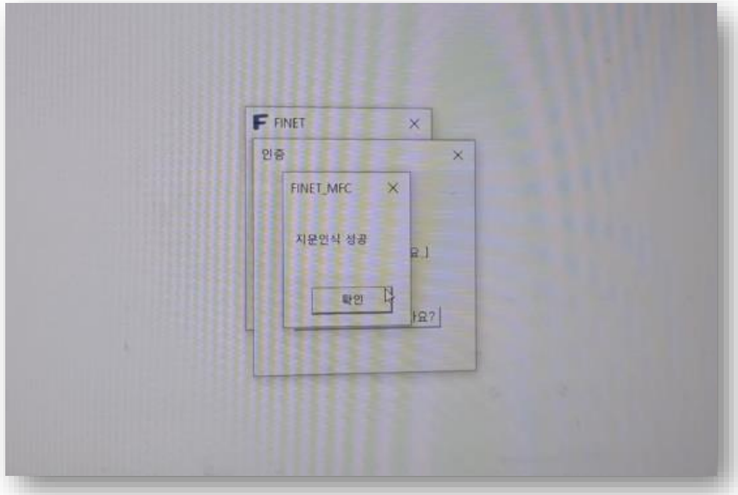


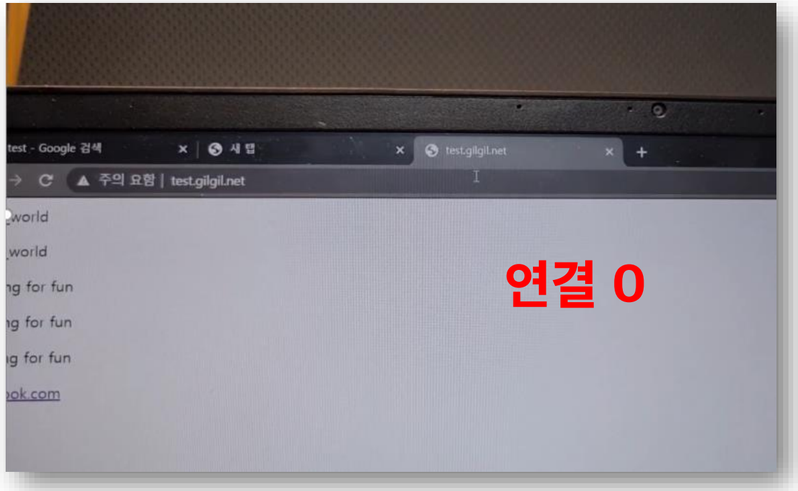
02 프로젝트 내용

3. 시연

FINET 프로그램을 이용하여 사용자 인증 수행



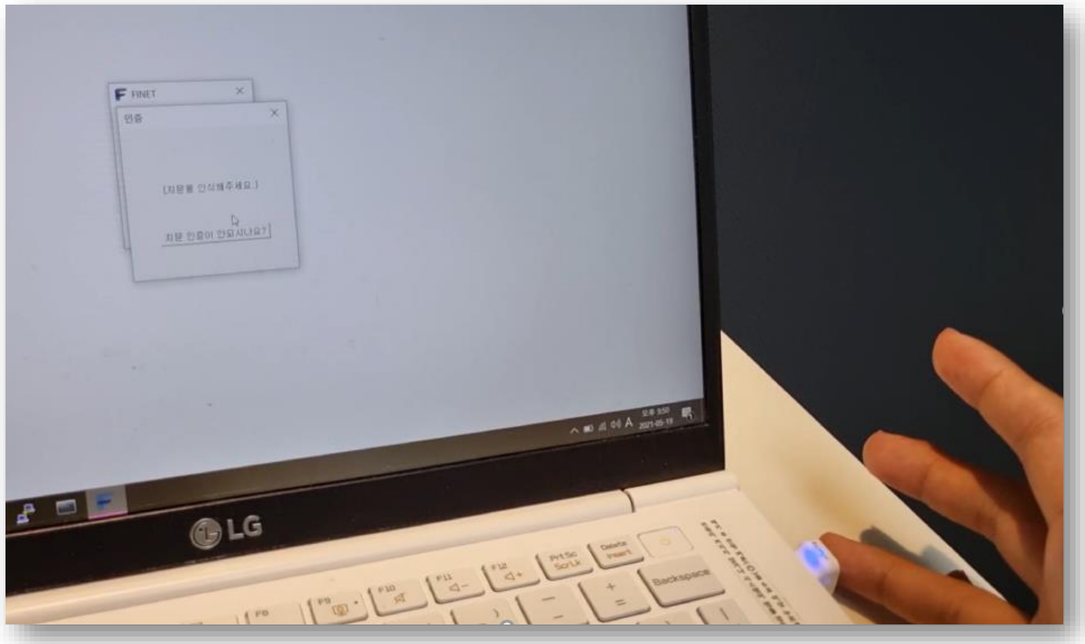


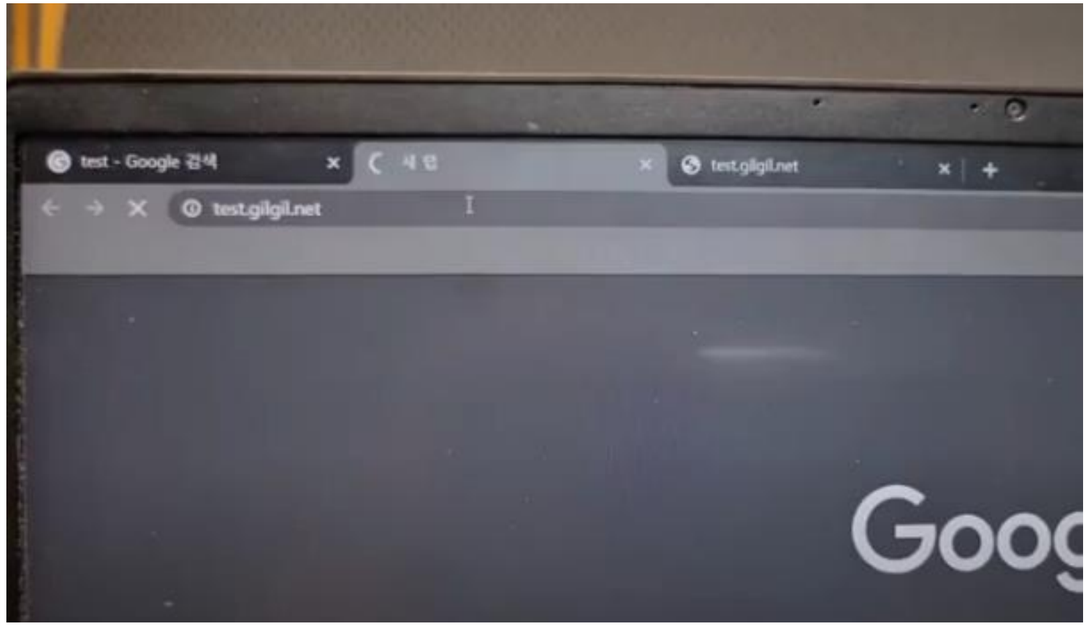


02 프로젝트 내용

3. 시연

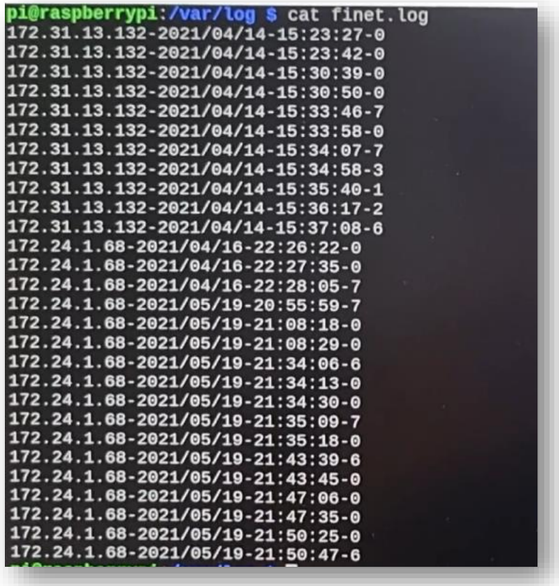
등록된 지문이 아니라면 사용자 인증 실패





성공 실패 로그 기록

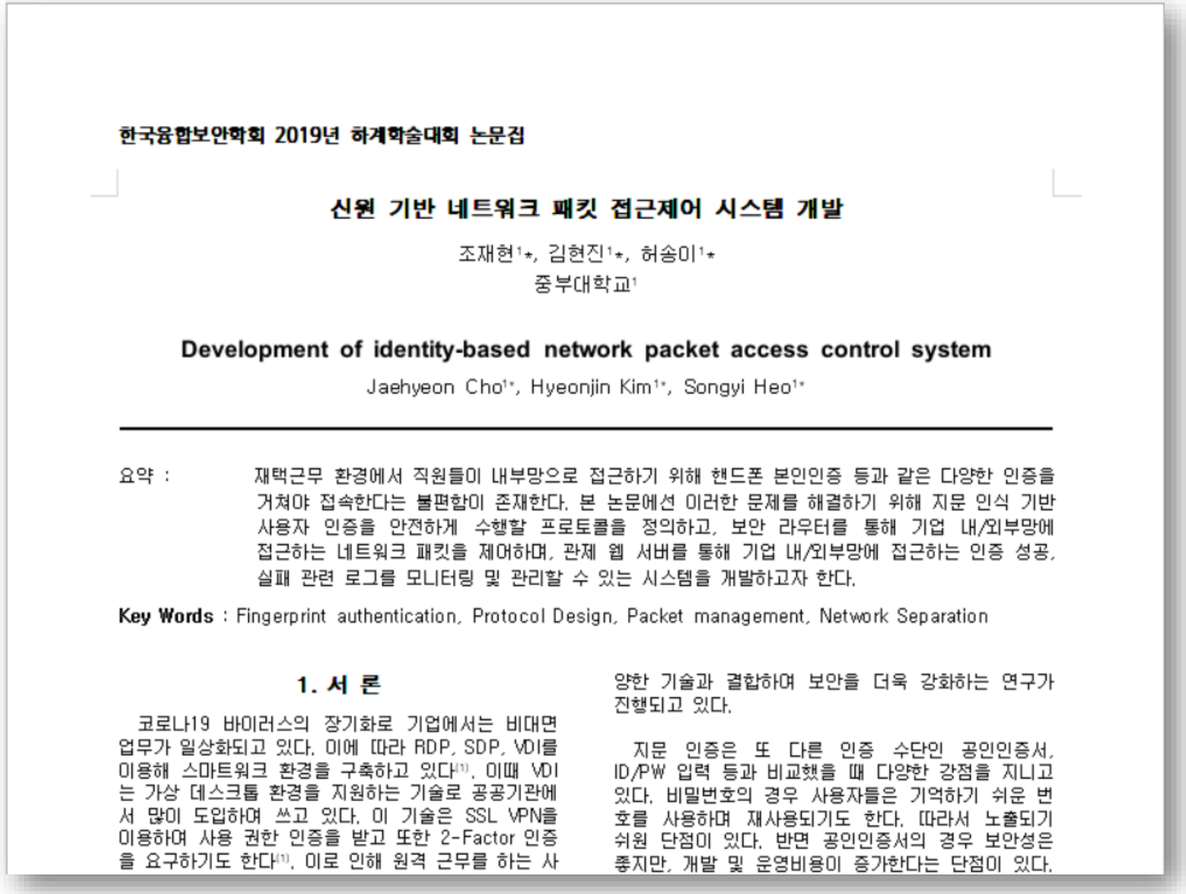

0. FINET - Received Packet
59377 61927
1. FINET - Valid HEADER
2. FINET - Valid CHECKSUM
3. FINET - Valid ID (1) Authentication Request
4. FINET - Valid Option - SUCCESS
5. FINET - Send C/R Challenge Packet
0. FINET - Received Packet
1. FINET - Valid HEADER
2. FINET - Valid CHECKSUM
3. FINET - Valid ID (3) C/R Response
4. FINET - Valid Option - SUCCESS
1f729e0504127759f41122e9751c6b587f754a216a1c5ae49987c2cfd08da092 64
auth_type : 1
q: select member_guid from member_tb where member_id = '89c056d22cdce155c196c796e2c6719d68f85aa26748ced6b299679b3a23ff4f'
challenge : 57955, res : 1f729e0504127759f41122e9751c6b587f754a216a1c5ae49987c2cfd08da092
DB - member_guid : 7f3ed960fa268242e453570f1de90a75d436b1f22b0f5fa8979f7f771ea3e0e9
res_2 : 7f3ed960fa268242e453570f1de90a75d436b1f22b0f5fa8979f7f771ea3e0e9
out_2 : 1f729e0504127759f41122e9751c6b587f754a216a1c5ae49987c2cfd08da092
cmp(res(<packet guid/pw+challenge>) == out_2(<db guid/pw+challenge))
CMP SUCCESS
bye
5. FINET - Valid Response Data
6. FINET - Send Response(success) Packet
wtrustip!!!!
7. LOGGING SUCCESS
Client Socket Thread Finish!
172.24.1.68 - trust_ip APPEND(10sec)!
172.24.1.68 - trust_ip EXPIRED
Client Socket Thread Start!
0. FINET - Received Packet
59377 61927
1. FINET - Valid HEADER
2. FINET - Valid CHECKSUM


pi@raspberrypi:/var/log $ cat finet.log
172.31.13.132-2021/04/14-15:23:27-0
172.31.13.132-2021/04/14-15:23:42-0
172.31.13.132-2021/04/14-15:30:39-0
172.31.13.132-2021/04/14-15:30:50-0
172.31.13.132-2021/04/14-15:33:46-7
172.31.13.132-2021/04/14-15:33:58-0
172.31.13.132-2021/04/14-15:34:07-7
172.31.13.132-2021/04/14-15:34:58-3
172.31.13.132-2021/04/14-15:35:40-1
172.31.13.132-2021/04/14-15:36:17-2
172.31.13.132-2021/04/14-15:37:08-6
172.24.1.68-2021/04/16-22:26:22-0
172.24.1.68-2021/04/16-22:27:35-0
172.24.1.68-2021/04/16-22:28:05-7
172.24.1.68-2021/05/19-20:55:59-7
172.24.1.68-2021/05/19-21:08:18-0
172.24.1.68-2021/05/19-21:08:29-0
172.24.1.68-2021/05/19-21:34:06-6
172.24.1.68-2021/05/19-21:34:13-0
172.24.1.68-2021/05/19-21:34:30-0
172.24.1.68-2021/05/19-21:35:09-7
172.24.1.68-2021/05/19-21:35:18-0
172.24.1.68-2021/05/19-21:43:39-6
172.24.1.68-2021/05/19-21:43:45-0
172.24.1.68-2021/05/19-21:47:06-0
172.24.1.68-2021/05/19-21:47:35-0
172.24.1.68-2021/05/19-21:50:25-0
172.24.1.68-2021/05/19-21:50:47-6

03 결과

1. 논문

논문 작성 및 발표



한국융합보안학회 2019년 하계학술대회 논문집

신원 기반 네트워크 패킷 접근제어 시스템 개발

조재현[1]*, 김현진[1]*, 허송이[1]*
중부대학교[1]

Development of identity-based network packet access control system

Jaehyeon Cho[1]*, Hyeonjin Kim[1]*, Songyi Heo[1]*

요약 : 재택근무 환경에서 직원들이 내부망으로 접근하기 위해 핸드폰 본인인증 등과 같은 다양한 인증을 거쳐야 접속한다는 불편함이 존재한다. 본 논문에선 이러한 문제를 해결하기 위해 지문 인식 기반 사용자 인증을 안전하게 수행할 프로토콜을 정의하고, 보안 라우터를 통해 기업 내/외부망에 접근하는 네트워크 패킷을 제어하며, 관제 웹 서버를 통해 기업 내/외부망에 접근하는 인증 성공, 실패 관련 로그를 모니터링 및 관리할 수 있는 시스템을 개발하고자 한다.

Key Words : Fingerprint authentication, Protocol Design, Packet management, Network Separation

1. 서 론

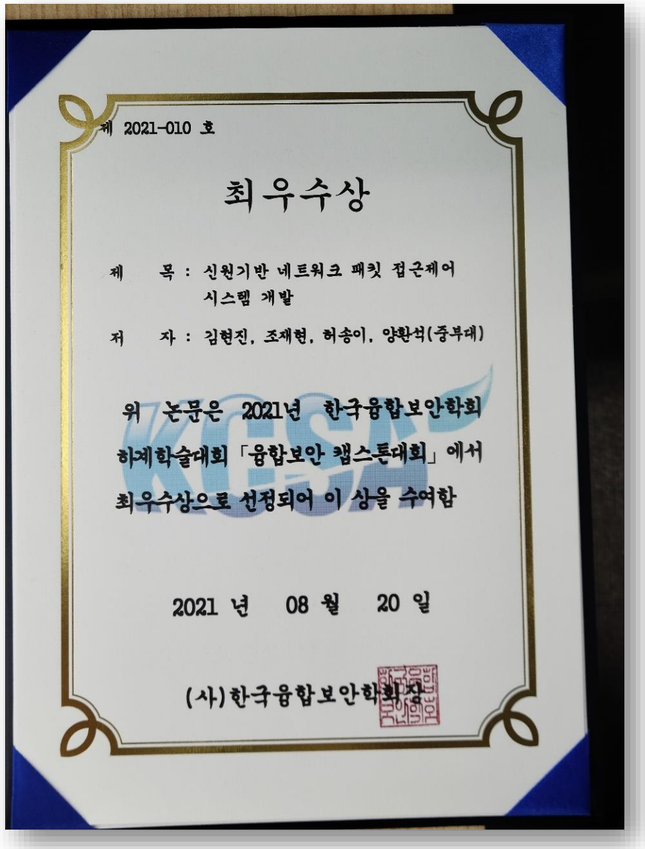
코로나19 바이러스의 장기화로 기업에서는 비대면 업무가 일상화되고 있다. 이에 따라 RDP, SDP, VDI를 이용해 스마트워크 환경을 구축하고 있다[1]. 이때 VDI는 가상 데스크톱 환경을 지원하는 기술로 공공기관에서 많이 도입하여 쓰고 있다. 이 기술은 SSL VPN을 이용하여 사용 권한 인증을 받고 또한 2-Factor 인증을 요구하기도 한다[1]. 이로 인해 원격 근무를 하는 사양한 기술과 결합하여 보안을 더욱 강화하는 연구가 진행되고 있다.

지문 인증은 또 다른 인증 수단인 공인인증서, ID/PW 입력 등과 비교했을 때 다양한 강점을 지니고 있다. 비밀번호의 경우 사용자들은 기억하기 쉬운 번호를 사용하며 재사용되기도 한다. 따라서 노출되기 쉬원 단점이 있다. 반면 공인인증서의 경우 보안성은 좋지만, 개발 및 운영비용이 증가한다는 단점이 있다.

03 결과

2. 수상

융합보안 캡스톤대회 최우수상



제 2021-010 호

최우수상

제 목 : 신원기반 네트워크 패킷 접근제어 시스템 개발

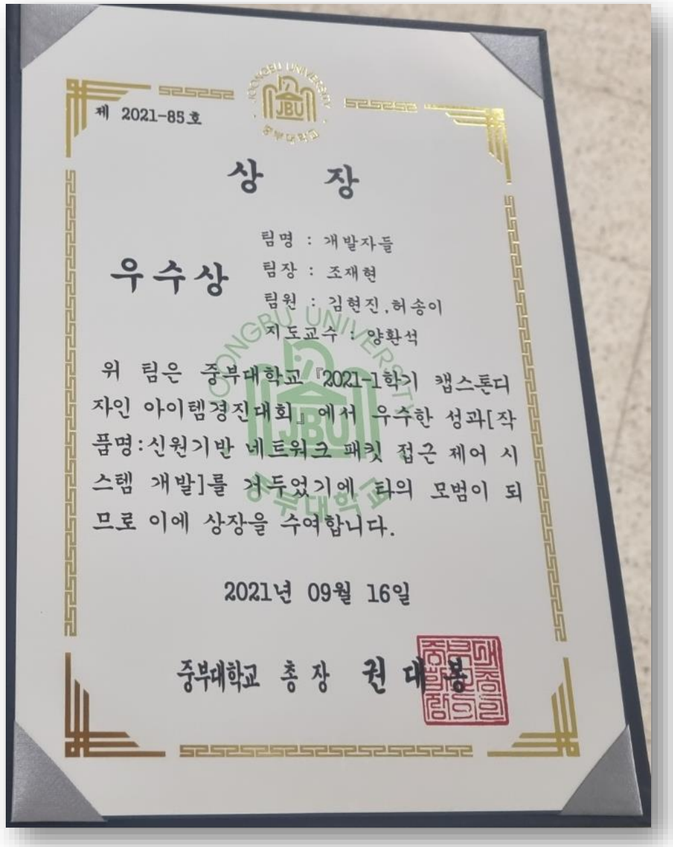
저 자 : 김현진, 조재현, 허송이, 양환석(중부대)

위 논문은 2021년 한국융합보안학회 하계학술대회「융합보안 캡스톤대회」에서 최우수상으로 선정되어 이 상을 수여함

2021 년 08 월 20 일

(사)한국융합보안학회장

캡스톤디자인 아이템경진대회 우수상



제 2021-85 호

상 장

우수상

팀명 : 개발자들
팀장 : 조재현
팀원 : 김현진,허송이
지도교수 : 양환석

위 팀은 중부대학교 「2021-1학기 캡스톤디자인 아이템경진대회」에서 우수한 성과[작품명:신원기반 네트워크 패킷 접근 제어 시스템 개발]를 거두었기에 타의 모범이 되므로 이에 상장을 수여합니다.

2021년 09월 16일

중부대학교 총장 권 대 용

감사합니다.