

안드로이드 앱 취약점 분석 툴 개발 (Tool 명 : 다자바)

2019. 10. 29

중부대학교 정보보호학과
지도교수 : 양환석 교수님

4 조 유한열
 신혜주
 배초아
 정현경

목 차

- 조원 편성
- 주제 선정
- 구 상 도
- 추진 경과
- 개발 환경 및 개발 내용
- 개발 시스템 운영
- 결론 및 기대효과

조원 편성

이름	역할
유한열	툴 개발, DB 구축(프로젝트 총괄)
신혜주	툴 개발, HTML query, PPT 작성
배초아	툴 개발, 엑셀 DB 연동, 보고서 작성
정현경	툴 개발, GUI 설계

주제 선정(1/2)

◆ 모바일 앱 사용이 지속적으로 증가 ⇨ 보안 위협이 증대

※ 안드로이드/iOS 앱의 절반 정도가 해킹위험에 노출

◆ 앱 사용이 늘어나면서 관리 부실 또한 심각

디지털타임스

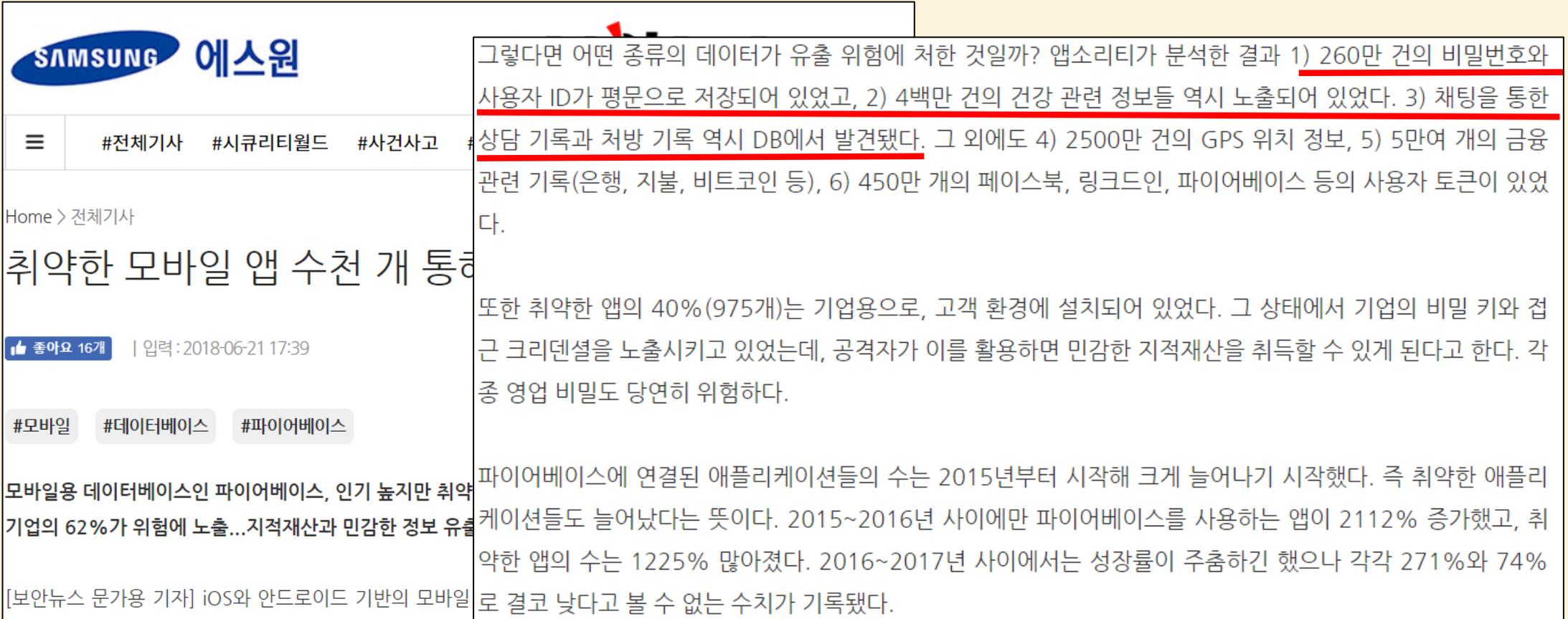
"무심코 깬 앱 '스파이 장치' 사용자 절반 해킹위험 노출"

시만텍 조사에 따르면, 인기가 높은 안드로이드 앱의 45%와 iOS 앱의 25%가 위치 확인 권한을 요청하고, 인기 안드로이드 앱의 46%와 iOS 앱의 24%가 사용자 기기의 카메라에 대한 접근 허가를 요청하는 것으로 나타났다. 또한 최고 인기 안드로이드 앱의 44%와 인기가 높은 iOS 앱의 48%에 이메일 주소가 공유되고 있다.

200개 이상의 앱과 서비스가 '스토커'에게 기본 위치 추적, 문자 수집 및 심지어 동영상 비밀 녹화 등 다양한 기능을 제공한다. 아울러 자녀, 친구 또는 분실된 휴대폰을 추적하기 위해 휴대폰 데이터를 수집하는 디지털 툴 역시 증가하면서 동의 없이 다른 사람을 추적할 수 있는 가능성이 높아지고 있다

주제 선정(2/2)

◆ 보안에 취약한 모바일 앱을 통해 개인정보 노출 심각



The image shows a screenshot of a Samsung News article. The article title is "취약한 모바일 앱 수천 개 통해..." (Through thousands of vulnerable mobile apps...). The article text discusses the security risks of mobile apps, mentioning that 40% of vulnerable apps are for business use. A text overlay box on the right side of the screenshot contains a summary of findings from a security analysis, listing several types of data that were leaked from various apps.

Text Overlay Box Content:

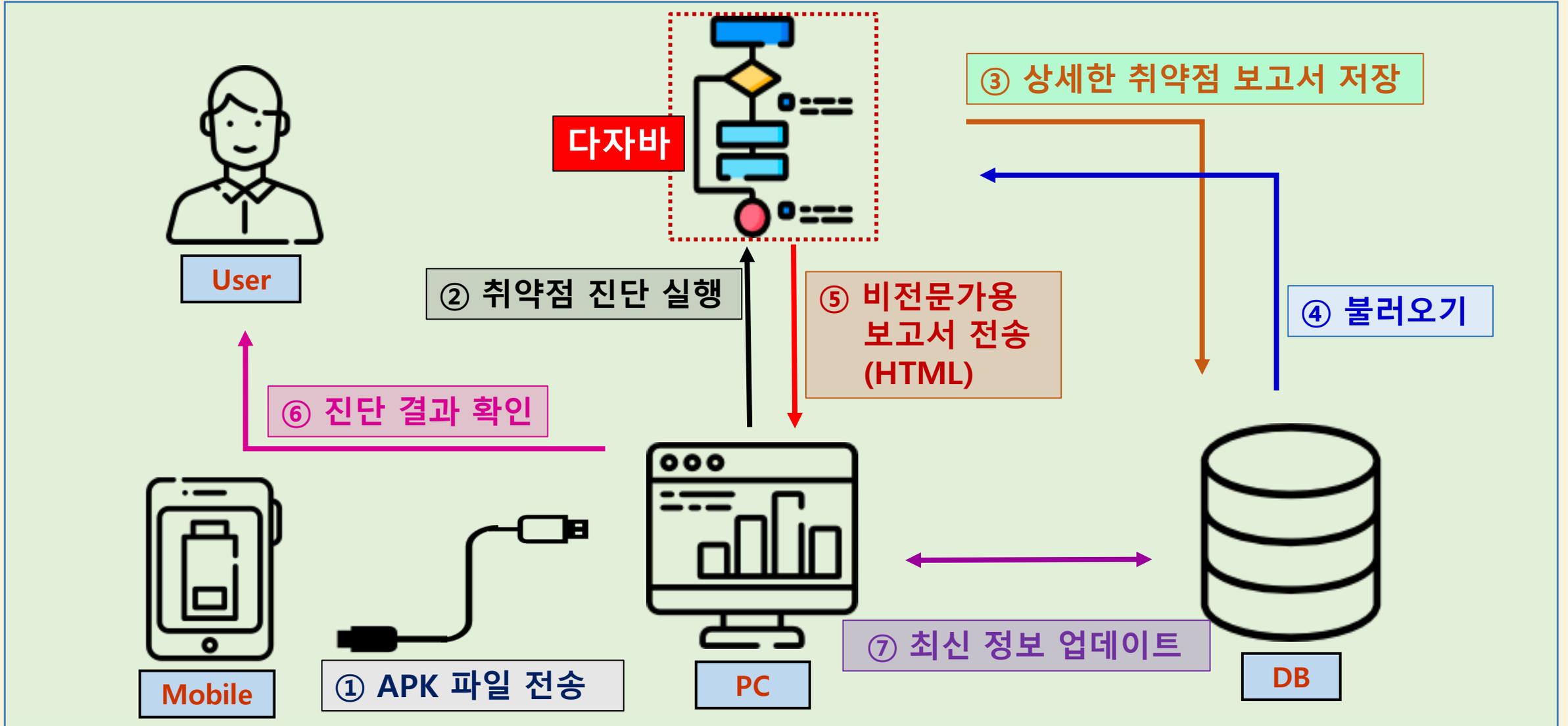
그렇다면 어떤 종류의 데이터가 유출 위험에 처한 것일까? 앱소리티가 분석한 결과 1) 260만 건의 비밀번호와 사용자 ID가 평문으로 저장되어 있었고, 2) 4백만 건의 건강 관련 정보들 역시 노출되어 있었다. 3) 채팅을 통한 상담 기록과 처방 기록 역시 DB에서 발견됐다. 그 외에도 4) 2500만 건의 GPS 위치 정보, 5) 5만여 개의 금융 관련 기록(은행, 지불, 비트코인 등), 6) 450만 개의 페이스북, 링크드인, 파이어베이스 등의 사용자 토큰이 있었다.

또한 취약한 앱의 40%(975개)는 기업용으로, 고객 환경에 설치되어 있었다. 그 상태에서 기업의 비밀 키와 접근 크리덴셜을 노출시키고 있었는데, 공격자가 이를 활용하면 민감한 지적재산을 취득할 수 있게 된다고 한다. 각종 영업 비밀도 당연히 위험하다.

파이어베이스에 연결된 애플리케이션들의 수는 2015년부터 시작해 크게 늘어나기 시작했다. 즉 취약한 애플리케이션들도 늘어났다는 뜻이다. 2015~2016년 사이에만 파이어베이스를 사용하는 앱이 2112% 증가했고, 취약한 앱의 수는 1225% 많아졌다. 2016~2017년 사이에서는 성장률이 주춤하긴 했으나 각각 271%와 74%로 결코 낮다고 볼 수 없는 수치가 기록됐다.

일반 사용자들이 위험을 쉽게 인지하고 취약점을 진단하는 보안시스템을 구현

구상도

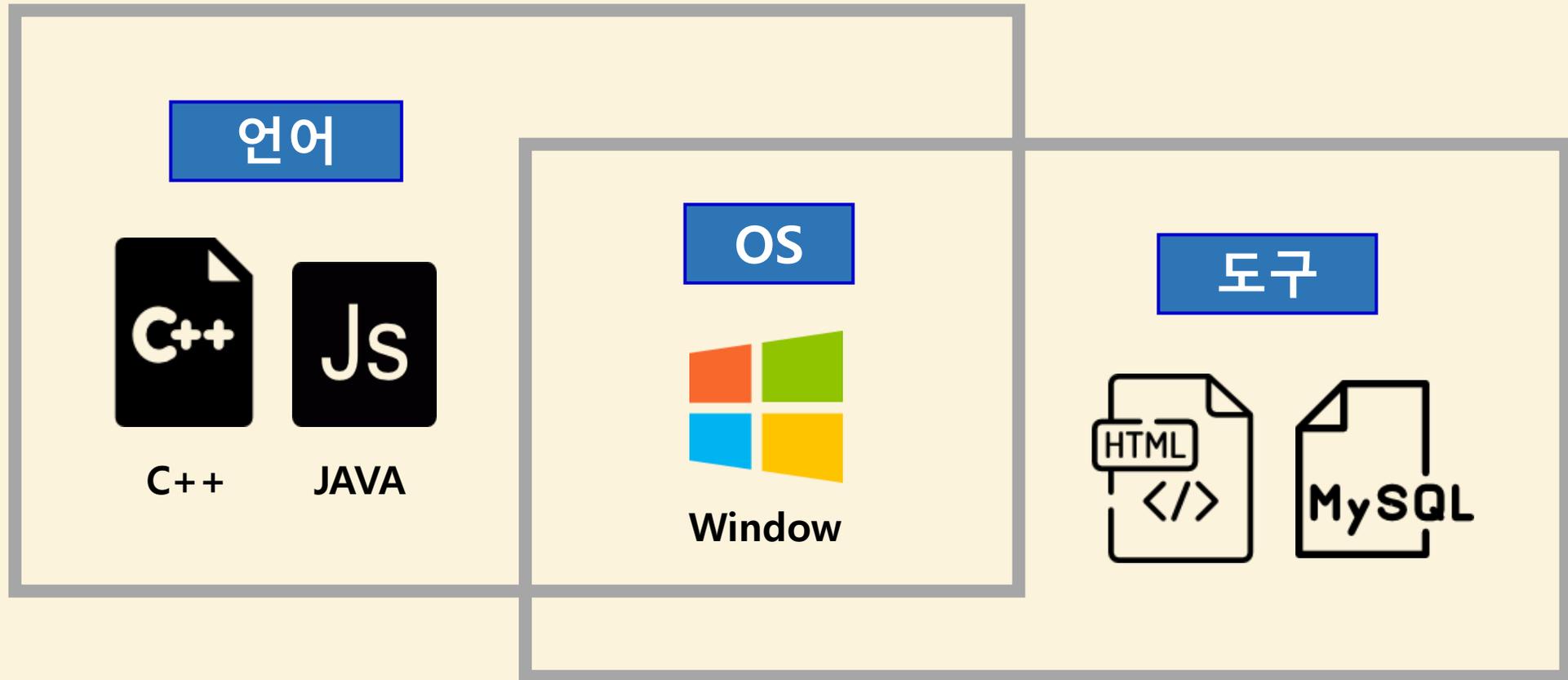


추진 경과

추진 업무	추진 기간 (2019년)	3월	4월	5월	6월	7월	8월	9월	10월
	주제 선정 및 계획 수립		■						
자료수집 - 취약점 공격기술 및 취약점 점검 방법 자료 수집			■	■	■				
Tool 개발 - 모바일 OWASP top10 기반 취약점 분석 툴 개발			■	■	■	■	■		
DB 구축 및 연동					■	■	■		
테스트 및 오류수정						■	■	■	
PPT 및 보고서 완성								■	■

개발 환경 및 개발 내용(1/8)

개발 환경



개발 환경 및 개발 내용(2/8)

앱 취약점 진단 항목 설정

◆ OWASP를 기반으로 한 안드로이드 앱 취약점 진단 시스템 개발

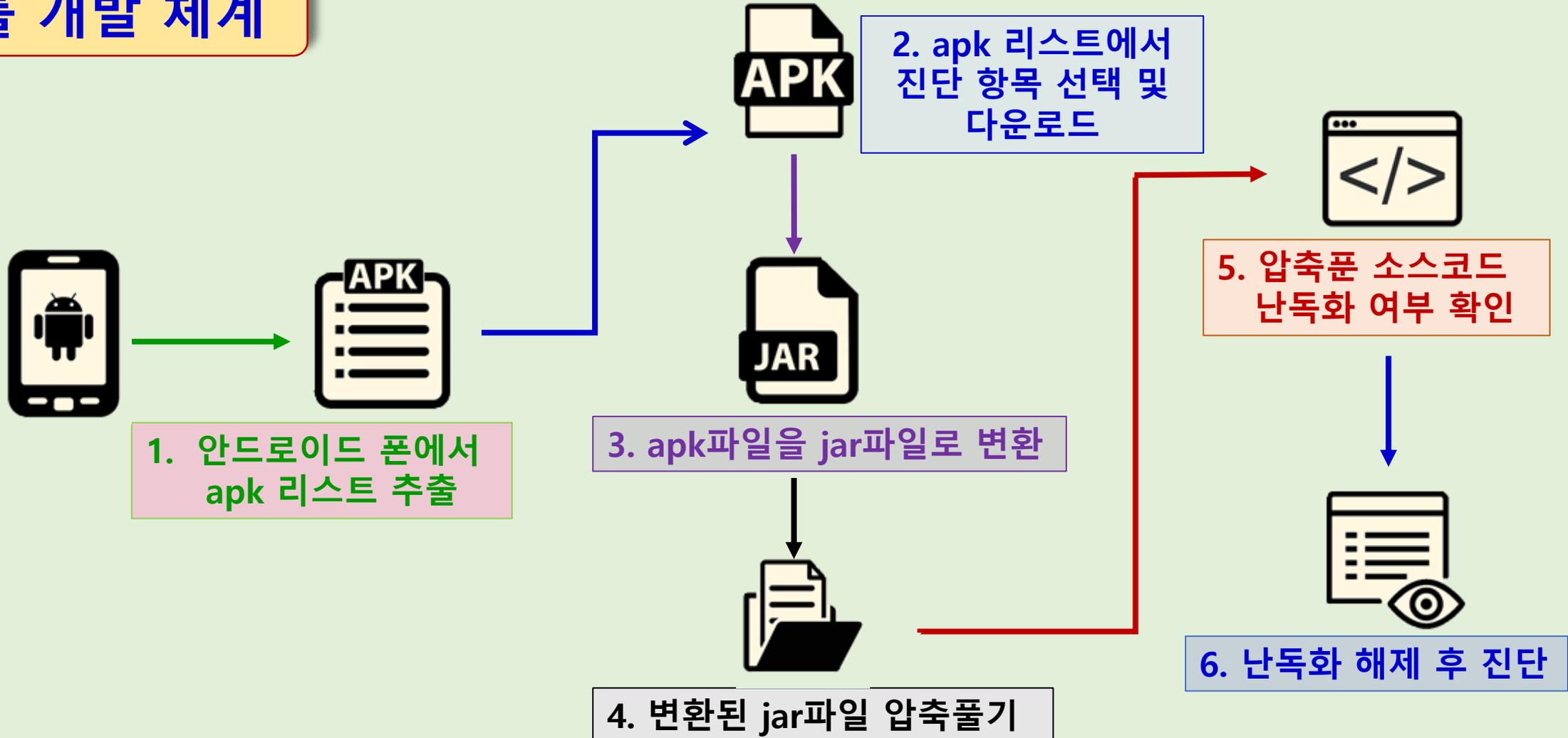
※ OWASP : The Open Web Application Security Project

앱 취약점 진단 항목

- 1) 적절하지 않은 플랫폼 사용
- 2) 취약한 데이터 저장소
- 3) 취약한 통신
- 4) 취약한 인증
- 5) 취약한 암호화
- 6) 취약한 권한부여
- 7) 취약한 코드품질
- 8) 코드 변조
- 9) 리버스 엔지니어링
- 10) 불필요한 기능

개발 환경 및 개발 내용 (3/8)

툴 개발 체계



개발 환경 및 개발 내용(4/8)

개발 프로그램 및 DB

◆ APK에서 소스 추출을 위한 툴 개발(프로그램 분석)

- ① 안드로이드 폰에서 APK 파일 추출 ⇨ ② APK 파일을 디컴파일하여 JAR 파일 추출
⇨ ③ JAR 파일 압축 해제 & 코드 난독화 해제 ⇨ ④ 취약점 분석

◆ 결과 보고서 출력 프로그램 개발

- 일반 사용자를 위한 보고서 : HTML & DB 연동
- 전문가를 위한 상세한 보고서 : EXCEL & DB 연동

◆ 사용자 편의성을 보장하기 위해 GUI 개발 및 보고서 관리를 위한 DB 구축

시스템 개발은 앱 취약점 분석 툴을 먼저 개발한 후 운영화면 등 GUI 개발

개발 환경 및 개발 내용(5/8)

APK 확인(안드로이드)

```
1 import java.io.IOException;
2 import java.io.BufferedReader;
3 import java.io.InputStreamReader;
4 import java.util.ArrayList;
5 import java.util.List;
6
7 class MainActivity {
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22     public String resultCommand(String cmd){
23         try{
24             p = Runtime.getRuntime().exec(cmd);
25             BufferedReader buffreader = new BufferedReader(new InputStreamReader(p.getInputStream()));
26             String line = null;
27             StringBuffer readbuff = new StringBuffer();
28
29             while((line = buffreader.readLine()) != null){
30                 if(line.indexOf("/data/app/") != -1){
31                     readbuff.append(line);
32                     readbuff.append("\n");
33                 }
34             }
35             return readbuff.toString();
36         }
37         catch (Exception e){
```

APK 리스트 출력

실행 결과

```
package:/data/app/com.kakao.story-2.apk=com.kakao.story
package:/data/app/com.kakao.talk-1.apk=com.kakao.talk
package:/data/app/com.nhn.android.band-2.apk=com.nhn.android.band
package:/data/app/com.nhn.android.kin-1.apk=com.nhn.android.kin
package:/data/app/com.nhn.android.mail-1.apk=com.nhn.android.mail
package:/data/app/com.nhn.android.ndrive-1.apk=com.nhn.android.ndrive
package:/data/app/com.nhn.android.search-1.apk=com.nhn.android.search
package:/data/app/com.picsart.studio-1.apk=com.picsart.studio
package:/data/app/com.sec.android.app.samsungapps-1.apk=com.sec.android.app.samsungapps
package:/data/app/com.sec.android.iap-1.apk=com.sec.android.iap
package:/data/app/com.skp.clink.invoke-1.apk=com.skp.clink.invoke
package:/data/app/com.skt.skaf.A000Z00040-2.apk=com.skt.skaf.A000Z00040
```

안드로이드 폰 내의 APK 리스트

개발 환경 및 개발 내용(6/8)

APK 추출/이동(안드로이드 ⇨ PC)

```
43 public String downloadFile(String apk, String fileName){
44     buff = new StringBuffer();
45     buff.append("cmd.exe ");
46     buff.append("/c ");
47     buff.append("adb pull ");
48     buff.append(apk);
49     buff.append(" ");
50     buff.append(fileName);
51
52     try {
53         P = Ru
54         buffre
55         String
56         readbu
57
58         while
59         re
60         re
61     }
62     return
63 }
64 catch (Exc
65     e.prin
66     System
```

```
75 Cmd cmd = new Cmd();
76 String apk = new String();
77 String fileName = new String();
78
79 String order = cmd.inputCommand("pm list packages -f");
```

APK파일 이동

`/data/app/com.kakao.talk-1.apk: 1 file pulled. 2.3 MB/s (33692654 byt`

실행 결과

이름	수정한 날짜	유형
.idea	2019-05-05 오후 7:17	파일 폴더
out	2019-05-02 오후 11:...	파일 폴더
src	2019-05-05 오전 3:45	파일 폴더
kakao.apk	2019-05-05 오후 7:19	APK 파일
Start_part1.iml		

선택한 APK 파일을 PC로 이동

개발 환경 및 개발 내용(7/8)

파일 변환/압축해제

```
1 package conversion2;
2 import java.io.*;
36 private static void createFile(File file, ZipInputStream
37 //디렉토리 확인
```

APK to ZIP

ZIP 압축해제

```
1 package conversion2;
2 import java.io.*;
3 class Zip2{
4     public static String FileName = "classes-dex2jar.jar";
5     public static String builder(String FileName) {
6     try {
7         ProcessBuilder builder = new ProcessBuilder(
8             "cmd.exe", "/c", "cd \"C:\\dex2jar-2.0\" && jar xvf ",FileName
9         builder.redirectErrorStream(true);
10        Process p = builder
11        BufferedReader r = r
12        while (true) {
13            String line;
14            line = r.readLine();
15            if (line == null)
16                System.out.println
17        }
18    }
19 }
20 }
21 }
22 }
23 }
```

DEX to JAR

JAR to CLASS

실행 결과

실행 결과(ZIP→DEX)

JAR 생성

CLASS 생성

APK 파일을 단계적으로 변환/압축해제하여 CLASS 파일 생성

개발 환경 및 개발 내용(8/8)

난독화 체크

※ 난독화 : 프로그래밍 언어로 작성된 코드를 읽기 어렵게 만드는 작업

ProGuard 설정 여부 확인

```
public static void Check_ProGuard() {
    // Proguard 설정 여부를 통해 난독화 판정
    try {
        String input1 = "minifyEnabled";
        String input2 = "true";

        File f = new File( pathname: "C:\\\\Users\\\\초아\\\\Desktop");
        FileReader fr = new FileReader(f);
        BufferedReader bufr = new BufferedReader(fr);
        String line = "";
        while ((line = bufr.readLine()) != null) {
            if (line.contains(input1)) {
                if (line.contains(input2))
                    unpack();
            }
            else
                continue;
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```
System.out.print("파일명 입력 (디렉토리): ");
Scanner sc2 = new Scanner(System.in);
String fileN = sc2.next();
File folder2 = new File( pathname: FolderPath + "\\\" + fileN);
FileReader fr = new FileReader(folder2);
BufferedReader bufr = new BufferedReader(fr);
String line = "";
while((line = bufr.readLine()) != null) {
    if(line.contains(input)) {
        int a = line.indexOf(input);
        int b = line.indexOf("(");
        String word = line.substring(a,b);

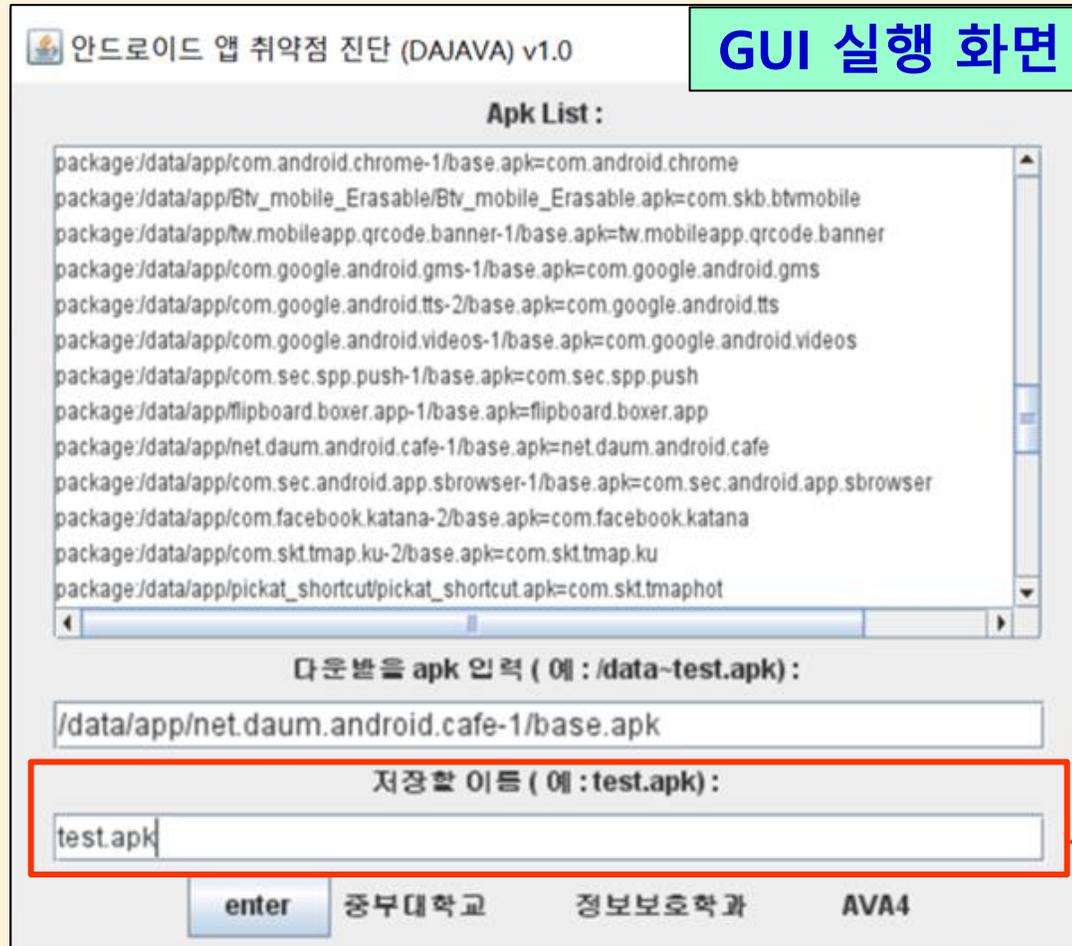
        String[] arr = word.split( regex: "(");
        if(arr[1].length() <= 2)
            continue;
    }
}
```

함수명 길이 확인

ProGuard 설정 여부/함수명 길이를 확인하여 난독화되어 있는지 체크

개발 시스템 운영(1/7)

취약 분석 시스템 구동



① 안드로이드 내부
APK 전체 리스트

② 취약점 진단을 원하는
APK 파일명 입력

③ PC에 저장할 이름 입력

개발 시스템 운영(2/7)

시스템 내부 분석작업 (파일 변환)

d2j-std-apk.sh	2014-10-27 오후 5...	APK 파일 PC로 이동	
InsecureBankv2.apk	2019-03-05 오후 1...	APK 파일	3,382KB
zipfstmp1566809201729348832.tmp	2019-05-05 오전 4...	TMP 파일	6,745KB

d2j-std-apk.sh	2014-10-27 오후 5...	APK 파일 ZIP 변환	
InsecureBankv2.zip	2019-03-05 오후 1...	압축(ZIP) 파일	3,382KB
zipfstmp1566809201729348832.tmp	2019-05-05 오전 4...	TMP 파일	6,745KB

classes.dex	2019-06-08			CLASS 파일 목록
classes-dex2jar.jar	2019-06-08			
BuildConfig.class	2019-05-04 오후 1...			
ChangePassword\$1.class	2019-05-04 오후 1...		CLASS 파일	
ChangePassword\$RequestChangePassw...	2019-05-04 오후 1...		CLASS 파일	
ChangePassword\$RequestChangePassw...	2019-05-04 오후 1...		CLASS 파일	
ChangePassword\$RequestChangePassw...	2019-05-04 오후 1...		CLASS 파일	
ChangePassword.class	2019-05-04 오후 1...		CLASS 파일	
CryptoClass.class	2019-05-04 오후 1...		CLASS 파일	
DoLogin\$RequestTask\$1.class	2019-05-04 오후 1...		CLASS 파일	
DoLogin\$RequestTask.class	2019-05-04 오후 1...		CLASS 파일	

DEX로부터 얻은 JAR 파일 압축 해제 후 CLASS 파일 획득

개발 시스템 운영(3/7)

시스템 내부 분석작업 (난독화 검증)

```
filelist = ClassList.Start(); // CLASS 파일 경로를 넘겨받아 난독화 검증
for(int i = 0; i < filelist.length; i++) {
    if(filelist[i] != null) {
        String classpath = filelist[i];
        cnt = CheckObfuscation.CheckFunction(classpath);
        if(cnt == 1) System.out.println("난독화 검증 실패");
    }
}
```

CLASS 파일 경로를 넘겨받아 난독화 검증

함수명 길이 점검

< 난독화 체크 >

- 1 : ProGuard 설정여부 확인
- 2 : 함수명 길이 확인

ProGuard 설정여부 점검

< 난독화 체크 >

- 1 : ProGuard 설정여부 확인
- 2 : 함수명 길이 확인

원하는 번호를 입력 : 1

난독화 설정 X

원하는 번호를 입력 : 2

파일 이름 = LogActivity.java

파일 이름 = MainActivity.java

파일 이름 = NotesProvider.java

파일 이름 = SQLInjectionActivity.java

파일명 입력 (디렉토리경로 포함) : LogActivity.java

난독화 설정 X

ProGuard 설정여부와 함수명 길이 점검으로 난독화 검증

개발 시스템 운영(4/7)

시스템 내부 분석작업 (작업 종합)

```
package:/data/app/com.sec.spp.push-1.apk=com.sec.spp.push
package:/data/app/net.daum.android.daum-1.apk=net.daum.android.d
```

앱 취약점 분석 종합

```
Write apk [예)/ 부터 .apk 까지 복붙] :
/data/app/com.hanjoon.scheduler-1.apk
Write file name [예) test.apk] :
scheduler-1.apk
```

APK파일 선택 후 PC로 이동

```
/data/app/com.hanjoon.scheduler-1.apk: 1 file pulled. 2.5 MB/s (10165325 bytes in 3.930s)
```

```
zip 변환 성공
zip파일 압축해제 성공
dex2jar classes.dex -> .\classes-dex2jar.jar
Detail Error Information in File .\classes-error.zip
Please report this file to http://code.google.com/p/dex2jar/issues/entry if possible.
dex -> jar 파일변환 성공
```

APK -> ZIP -> DEX -> JAR 파일 변환

```
생성됨: com/
생성됨: com/hanjoon/
생성됨: com/hanjoon/scheduler/
증가됨: com/hanjoon/scheduler/Application.class
증가됨: com/hanjoon/scheduler/MessageGuardException.class
Jar파일 압축해제 성공
```

JAR 압축 해제 후 CLASS파일 생성

```
Directory Name:C:\DaJaVa\dex2jar-2.0
0C:\DaJaVa\dex2jar-2.0\com\hanjoon\scheduler\Application.class
1C:\DaJaVa\dex2jar-2.0\com\hanjoon\scheduler\MessageGuardException.class
난독화 설정 X
```

CLASS파일 난독화 확인

개발 시스템 운영(5/7)

분석 결과 저장 (DB)

```
mysql> select * from result;
```

No	FileName	Date	Report
1	InsecureBankv2.apk	2019-09-10 17:31:26	C:DaJaVa

```
1 row in set (0.01 sec)
```

프로그램 실행 시 저장되는 APK 정보

```
mysql> select * from result2;
```

No	Grade	Name	Content
1	상	디버그 모드 확인	디버그 모드 작동 중
2	상	백업 허용 확인	adb를 통해 백업 허용
3	상	사용자의 위치 정보 접근 확인	위치 정보 접근 허용
4	상	Phone 권한 확인	장치의 전화 번호, 네트워크 정보, 진행중인 통화의 상태 읽어오기 허용
5	상	액티비티 컴포넌트 확인	노출된 Activity 갯수 :4

```
5 rows in set (0.01 sec)
```

검사한 APK의 취약점 정보

개발 시스템 운영(6/7)

분석 결과 저장 (엑셀)

 test 6KB Microsoft Excel 97-2003 Worksheet

insecurebank.apk / 2019-09-24 15:43:13.0			
점검 항목명	등급	상태	권유 조치 방안
디버그 모드 확인	상	디버그 모드 작동 중	임의의 코드를 주입 가능, 설정값을 false로 변경 권유
백업 허용 확인	중	adb를 통해 백업 허용	개인 데이터를 인가받지 않은 사용자가 추출 가능
사용자의 위치 정보 접근	중	위치 정보 접근 가능	Wi-Fi와 같은 네트워크 위치 소스를 통해 위치 정보 유출가능, 기본 권한에서 배제 권유
Phone 권한 확인	상	사용자의 통화 기록 읽기 허용	장치의 전화번호, 네트워크 정보, 진행중인 통화의 상태 및 기록 읽기 가능
액티비티 컴포넌트 확인	하	액티비티 개수 3개	다른 애플리케이션의 Activity를 실행 할 수 있음

엑셀 보고서

프로그램 실행 시 자동으로 상세한 취약점 진단 보고서를 엑셀로 저장

개발 시스템 운영(7/7)

분석 결과 저장 (HTML)

HTML 보고서

DaJaVa

made by JBU information security Team AVA4. 2019

< 어플리케이션 정보 >
점검 파일 명 : InsecureBankv2.apk
점검 일자 : 2019-10-14 20:57:20

◆ 점검 결과 : 취약



◆ 취약점 발견 갯수

상 : 5개
중 : 0개
하 : 0개

◆ 취약점 내용 설명

<등급 : 상> <점검 항목 명 : 디버그 모드 확인>

비전문가용 보고서를 프로그램 실행 시 html창으로 띄워 줌

결론 및 기대효과

◆ 결 론

- 안드로이드 폰에서 APK 파일을 변환/압축해제하여 취약점을 분석하고 HTML과 엑셀을 이용하여 취약점 분석 결과를 출력하는 시스템을 구현
- APK 디컴파일, .xml/.dex 파일 복호화 후 불러들이기와 소스코드 난독화 진단부문도 추가하여 취약점 분석시스템의 완성도를 향상

◆ 기대효과

- 일반 사용자들의 어플리케이션을 자체 개발한 툴로 취약점을 진단함으로써 보안에 대한 경각심을 일깨우는 계기 마련
- 시스템 개발기획 및 프로그램 개발을 합동으로 추진함으로써 보안시스템의 추진 노하우를 습득하고 코딩 역량을 배양. 끝.

Q & A

감사합니다