안드로이드 앱 취약점 분석 툴 개발
(Tool 명 : 다자바)

팀 명 : AVA4

지도 교수 : 양환석 교수님

팀 장 : 유한열

팀 원 : 배초아

신혜주

정현경

2019. 10.

중부대학교 정보보호학과

목 차

1. 서 론

1.1 연구 배경

연구의 배경으로 정보보호 컨설팅을 위해 모여서 주제를 선정하던 중, 모바일 특히, 안드로이드 앱 취약점에 흥미를 느끼게 되어 취약점 탐색 및 도구 개발을 시작하게 되었다.

1.2 연구 필요성

최근 모바일 앱 사용이 지속적으로 증가함에 따라 관리 부실 및 보안 위협이 증대되는 사례가 많았다. 특히, 보안에 취약한 모바일 앱을 통해 개인정보 노출이 심각하여 일반 사용자들이 위험을 쉽게 인지하고 취약점을 진단하는 보안시스템을 구현하는 것을 주제로 선정하였다.

1.3 연구 목적 및 주제선정

모바일 안드로이드의 수많은 취약점 OWASP TOP 10에 집중하기로 하였다. OWASP란 The Open Web Application Security Project로 오픈소스 웹 애플리케이션 보안프로젝이다. OWASP TOP 10은 웹 애플리케이션 취약점 중에서 빈도가 많이 발생하고, 보안상 영향을 크게 줄 수 있는 10가지를 선정하여 2004년, 2007년, 2010년, 2017년을 기준으로 발표되었고 문서가 공개 되었다.

OWASP 모바일 보안 프로젝트(OWASP Mobile Security Project)의 모바일 보안 위협 Top 10 OWASP는 4년마다 취약점을 발표하며 모바일은 2년마다 발표하고 있다. OWASP Mobile Top 10은 다음과 같다.

OWASP Mobile Top 10 - 2016
M1 - 적절하지 않은 플랫폼 사용
M2 - 취약한 데이터 저장소
M3 - 취약한 통신
M4 - 취약한 인증
M5 - 취약한 암호화
M6 - 취약한 권한부여
M7 - 취약한 코드품질
M8 - 코드 변조
M9 - 리버스 엔지니어링
M10 - 불필요한 기능

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.
현재 웹 애플리케이션 개발에 가장 많이 사용하는 언어 가운데 하나이고, 모바일 기기용 소프트웨어 개발에도 널리 사용하고 있다. 현재 버전 10까지 출시했다.
자바로 개발된 프로그램은 CPU나 운영 체제의 종류에 관계없이 JVM을 설치할 수 있는 시스템에서는 어디서나 실행할 수 있으며, 이 점이 웹 애플리케이션의 특성과 맞아떨어져 폭발적인 인기를 끌게 되었다.

2.2 Windows 10

마이크로소프트에서 개발한 Windows 8, Windows Phone 8, Windows 8.1, Windows Phone 8.1의 후속작이자 지금까지 넘버링 방식으로 발매된 마지막 Windows 이다. 한국 표준시(UTC+9) 기준 2015년 7월 29일 오후 1시에 공식 출시되었다. Windows NT 4.0 다음으로 지원 플랫폼이 많은 운영체제이기도 하다.

2.3 HTML

HTML은 하이퍼텍스트 마크업 언어(HyperText Markup Language)라는 의미의 웹 페이지를 위한 지배적인 마크업 언어다. 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법을 제공한다. 웹 페이지 콘텐츠 안의 꺾쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다. HTML은 웹 브라우저와 같은 HTML 처리 장치의 행동에 영향을 주는 자바스크립트와 본문과 그 밖의 항목의 외관과 배치를 정의하는 CSS 같은 스크립트를 포함하거나 불러올 수 있다.

2.4 VBA

마이크로소프트 사가 1990년대에 개발한 범용 프로그래밍 언어인 마이크로소프트 비주얼 베이직(Visual Basic)을 마이크로소프트 오피스에 탑재한 것으로, 이 VBA를 사용하여 엑셀(Excel), 액세스(Access), 워드(Word) 등의 오피스 응용프로그램 소프트웨어의 기능을 사용자가 정의하거나 확장할 수 있도록 한다. 기계어로 변환하는 컴파일 단계 없이 직접 실행이 가능하다.

2.5 GUI

GUI는 사용자가 그래픽을 통해 컴퓨터와 정보를 교환하는 작업 환경을 말한다. 이제까지의 사용자 인터페이스는 키보드를 통한 명령어로 작업을 수행시켰고, 화면에 문자로 표시하였다. 그래픽 유저 인터페이스에서는 마우스 등을 이용하여 화면의 메뉴 중에서 하나

를 선택하여 작업을 지시한다.
GUI는 도스(DOS)의 명령어 인터페이스와는 대조적이다. GUI의 요소를 살펴보면 윈도(Windows), 스크롤바, 아이콘 이미지, 단추들을 포함한다. 1980년대 후반부터 IBM PC 및 워크스테이션에서도 GUI가 보급되어 현재의 컴퓨터는 GUI를 사용하고 있다. 마이크로소프트사의 윈도, 애플 매킨토시의 GUI가 그 예이다.

2.6 MySQL

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. 하나의 옵션은 GPL이며, GPL 이외의 라이선스로 적용시키려는 경우 전통적인 지적재산권 라이선스의 적용을 받는다.
데이터베이스를 관리하기 위한 GUI 기반 툴을 따로 내장하지 않기 때문에 일반적으로 명령줄 인터페이스를 사용하거나, MySQL 워크벤치와 같은 MySQL 프론트앤드 소프트웨어 및 웹 애플리케이션을 이용한다.

2.7 PHP

PHP(Hypertext Preprocessor)는 프로그래밍 언어의 일종이다. 원래는 동적 웹 페이지를 만들기 위해 설계되었으며 이를 구현하기 위해 PHP로 작성된 코드를 HTML 소스 문서 안에 넣으면 PHP 처리 기능이 있는 웹 서버에서 해당 코드를 인식하여 작성자가 원하는 웹 페이지를 생성한다. 근래에는 PHP 코드와 HTML을 별도 파일로 분리하여 작성하는 경우가 일반적이며, PHP 또한 웹서버가 아닌 php-fpm(PHP FastCGI Process Manager)을 통해 실행하는 경우가 늘어나고 있다. ASP(Active Server Pages)와 같이 스크립트에 따라 내용이 다양해서 동적 HTML 처리 속도가 빠르며, PHP 스크립트가 포함된 HTML 페이지에는 .php, .php3, .phtml이 붙는 파일 이름이 부여된다. 처음에는 'PersonalHome Page Tools'이라 불렸으며, 공개된 무료 소스이다.

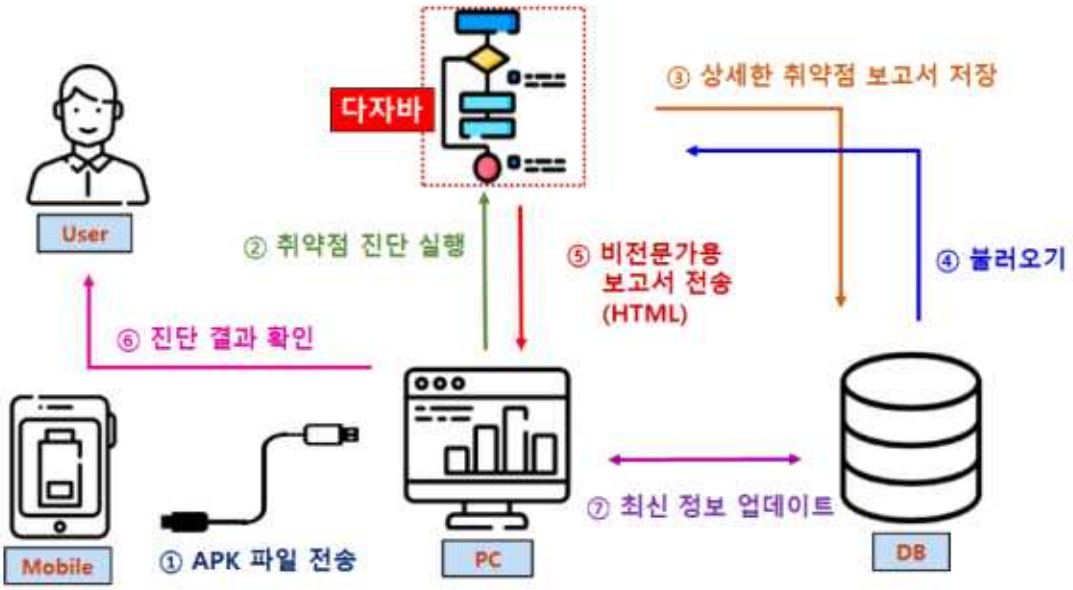
2.8 Excel

미국의 컴퓨터 소프트웨어 회사인 마이크로소프트사에서 개발한 윈도 환경의 스프레드시트 프로그램이다. 퍼스널컴퓨터(personal computer)와 매킨토시(Macintosh) 컴퓨터용의 스프레드시트(spread sheet) 프로그램이다.
스프레드시트란 여러 가지 도표 형태의 양식에 계산, 표기되는 사무업무를 자동으로 하는 표계산 프로그램으로 계산기와 계산용지 등이 통합되어 연산 및 표를 작성하고 그래프를 그리는 소프트웨어를 말한다. 윈도 환경에서 많은 스프레드시트를 연결하고 통합하여 여러 가지 도형과 그래프 등을 작성할 수 있다.
엑셀은 윈도 환경에서 사용자의 그래픽 환경을 제공하는데 스프레드시트 기능을 비롯해 매크로, 그래픽, 데이터베이스 기능과 지도·차트 작성 등 통합 문서작성에 필요한 기능도 제공한다.

3. 본론

3.1 시스템 구성

사용자가 PC에 단말기(스마트폰)를 연결 후 검사를 원하는 앱의 APK파일을 선택하면 다자바 자동화 툴이 선택한 APK파일을 PC로 이동 후 취약점 진단을 실행한다. 사용자 친화적인 비전문가용 HTML 결과 보고서와 전문적 내용을 포함한 전문가용 엑셀 파일 형식의 결과 보고서를 통해 취약점을 알 수 있고 이를 통해 단말기에 유해한 앱인지를 관리할 수 있다.

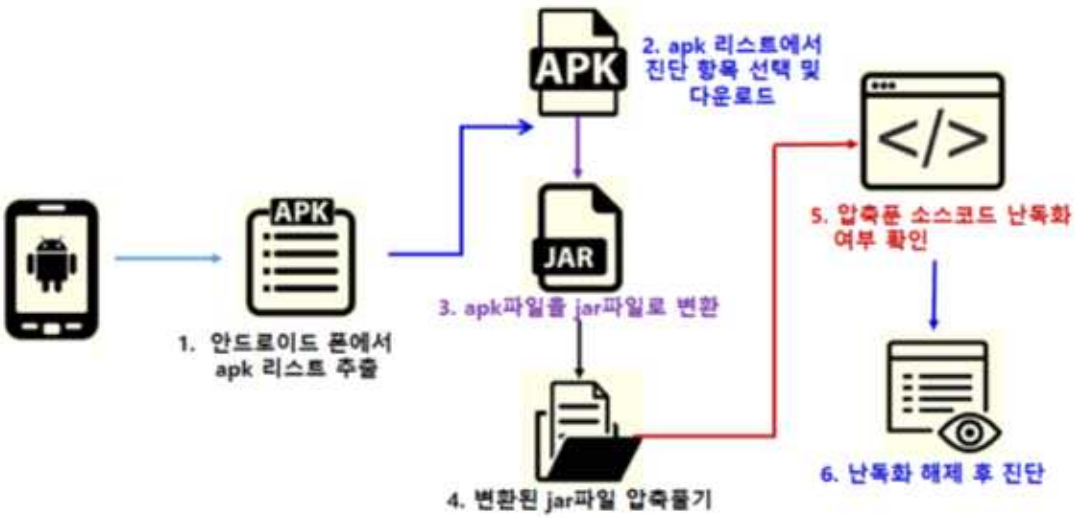


3.2 프로그램 구성

본 프로그램의 구성은 Windows10 환경에서 객체 지향 프로그래밍 언어로서 보안성이 뛰어난 JAVA, 하이퍼텍스트 마크업 언어(HyperText Markup Language)라는 의미의 웹 페이지를 위한 지배적인 마크업 언어인 HTML, 마이크로소프트 사가 1990년대에 개발한 범용 프로그래밍 언어인 마이크로소프트 비주얼 베이직(Visual Basic), JAVA에서 제공하는 GUI, 세계에서 가장 많이 쓰이는 오픈 소스의 관계형 데이터베이스 관리 시스템 MYSQL, 동적 웹 페이지를 만들기 위해 설계된 프로그래밍 언어의 일종 PHP, 마이크로소프트사에서 개발한 윈도 환경의 스프레드시트 프로그램 Excel로 되어 있다. 모바일 앱을 통해 개인정보 노출이 심각함으로 필요성을 느꼈고, 일반 사용자들이 위험을 쉽게 인지하고 취약점을 진단하는 보안시스템을 구현을 목표로 하고 있다.

3.2.1 APK 소스 추출 툴

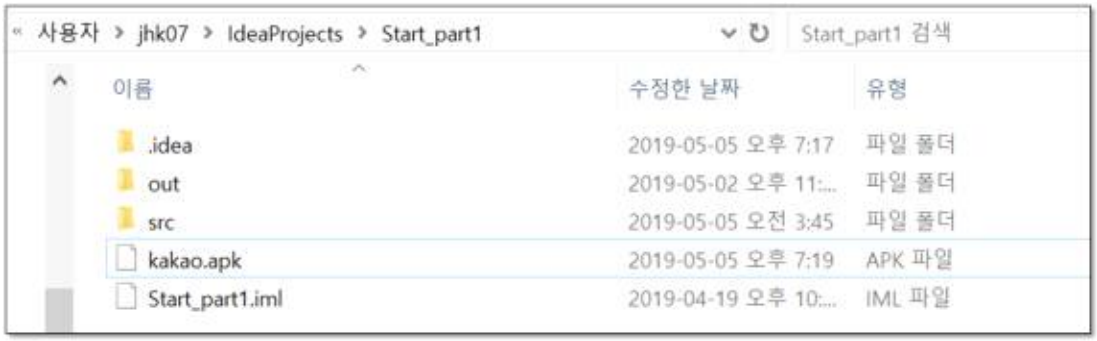
다자바 툴은 단말기(스마트폰)를 PC에 연결 후 프로그램을 구동 시키면 APK파일의 리스트를 보여주고 선택한 APK파일 PC에 다운 받을 수 있다. 다운받은 APK를 ZIP파일로 파일 변환 한다. ZIP파일로부터 순차적으로 DEX파일과 JAR파일로 변환 및 압축 해제하여 최종적으로 CLASS파일을 생성한다. 생성된 파일들의 난독화 여부를 확인 후 앱 취약점 분석을 시작한다.



< APK 추출 과정 >

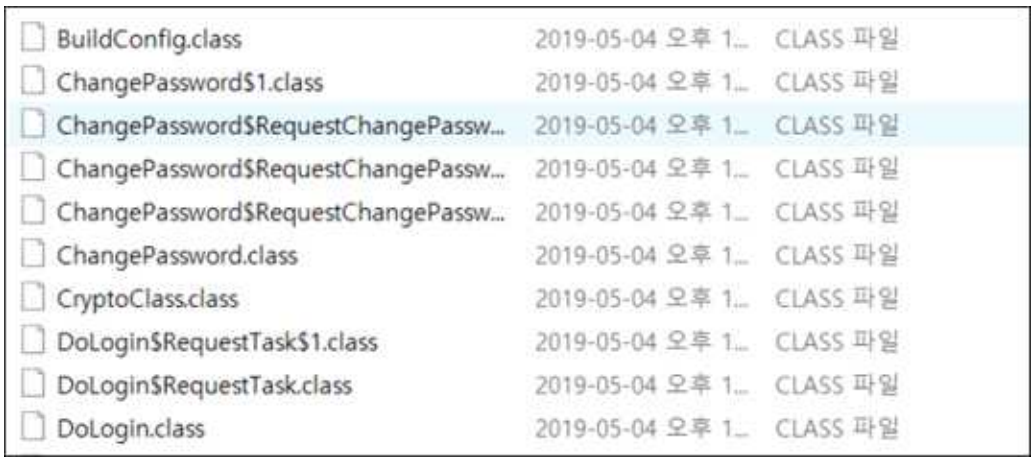
```
package:/data/app/com.kakao.story-2.apk=com.kakao.story
package:/data/app/com.kakao.talk-1.apk=com.kakao.talk
package:/data/app/com.nhn.android.band-2.apk=com.nhn.android.band
package:/data/app/com.nhn.android.kin-1.apk=com.nhn.android.kin
package:/data/app/com.nhn.android.mail-1.apk=com.nhn.android.mail
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.lap-1.apk=com.sec.android.lap
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
package:/data/app/com.skt.skaf.OA00018282-2.apk=com.skt.skaf.OA00018282
```

```
/data/app/com.kakao.talk-1.apk: 1 file pulled. 2.3 MB/s (33692654 bytes in 14.066s)
```



< 안드로이드 단말 내 APK 리스트출력 및 APK 추출 >

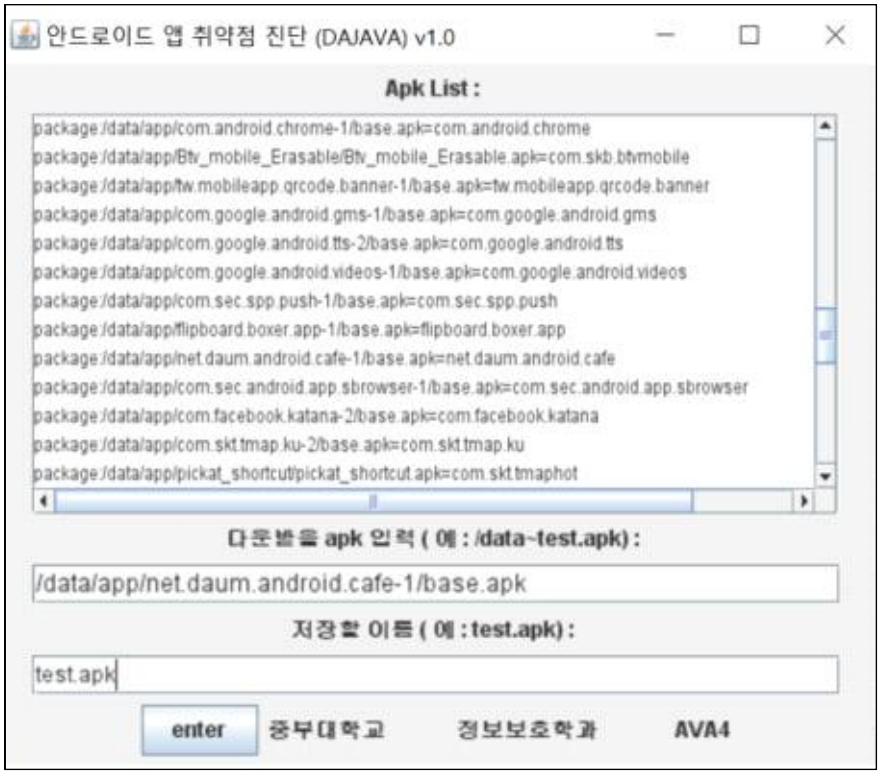
```
zip 변환 성공
zip파일 압축해제 성공
dex2jar classes.dex -> .\classes-dex2jar.jar
Detail Error Information in File .\classes-error.zip
Please report this file to http://code.google.com/p/dex2jar/issues/entry if possible.
dex -> jar 파일변환 성공
 생성됨: com/
 생성됨: com/hanjoon/
 생성됨: com/hanjoon/scheduler/
증가됨: com/hanjoon/scheduler/Application.class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



< 파일 변환 및 압축해제하여 Class 파일 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
```

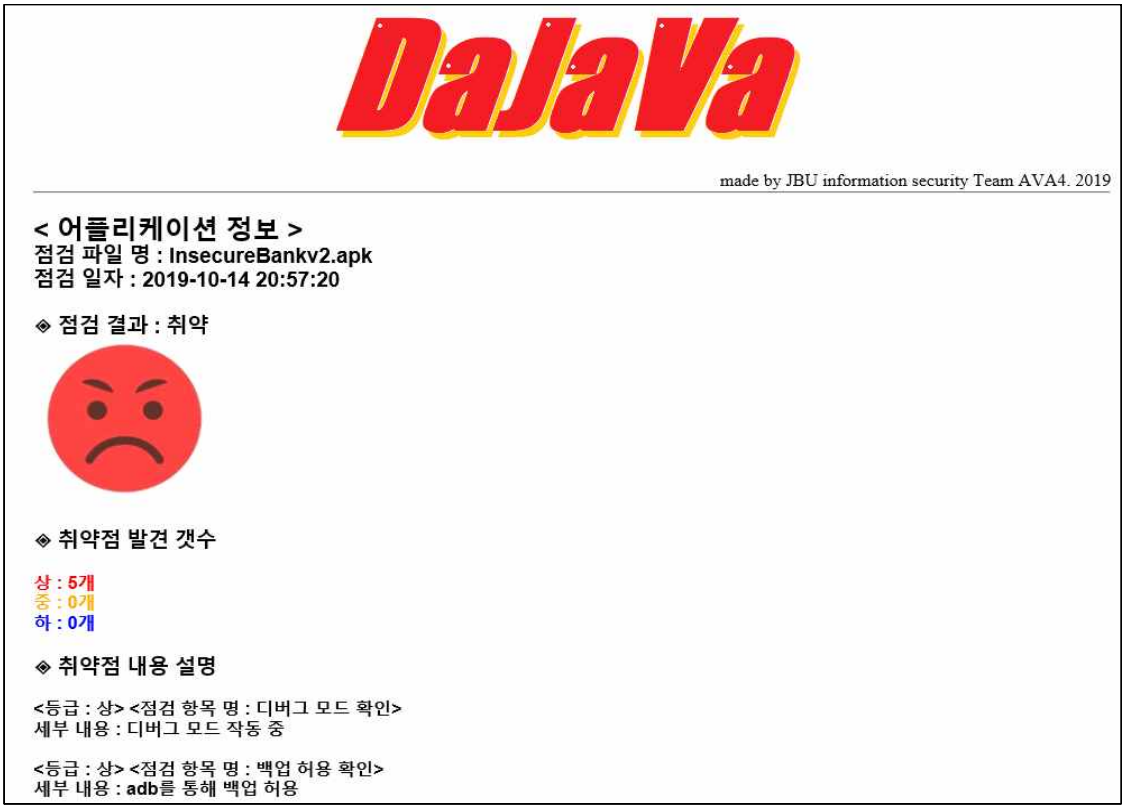
< 난독화 체크 >



< GUI 실행창 >

3.2.2 결과 보고서 출력 프로그램

취약점 분석이 끝나면 한 눈에 보기 쉬운 비전문가용 HTML 형식의 결과 보고서와 상세 정보를 확인할 수 있는 전문가용 Excel 형식의 결과 보고서를 자동으로 출력 및 저장한다. 결과 보고서 관리와 최신 정보 업데이트를 위해 DB와도 연동되어 있다.



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>
세부 내용 : 디버그 모드 작동 중

<등급 : 상> <점검 항목 명 : 백업 허용 확인>
세부 내용 : adb를 통해 백업 허용

< HTML결과 보고서 >

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

< Excel 결과 보고서 >

3.2.3 DB 구축

앱의 이름과 취약점 분석을 시작한 시간이 DB에 저장되며 그에 따른 취약점 분석 결과 역시 따로 DB에 저장된다. 새로운 취약점의 업데이트와 최신 정보 업데이트에도 DB를 사용한다.

```
mysql> select * from result;
+----+-------------------+---------------------+-----------+
| No | FileName          | Date                | Report    |
+----+-------------------+---------------------+-----------+
|  1 | InsecureBankv2.apk | 2019-09-10 17:31:26 | C:DaJaVa  |
+----+-------------------+---------------------+-----------+
1 row in set (0.02 sec)
```

< DB 결과 - ① 검사한 파일 및 검사 시간 >

```
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                                               |
|  6 | 상    | 루팅탐지                      | 노출된 루팅 갯수 : 2                                                  |
|  7 | 상    | 루팅탐지2                     | Root권한 획득 가능                                                    |
|  8 | 상    | Su 권한                       | 시스템 명령 실행중                                                    |
|  9 | 상    | Su 권한2                      | 응용 프로그램이 시스템 명령 실행 가능                                 |
| 10 | 상    | 수출업체                      | 수출업체 취약                                                         |
| 11 | 상    | 수출업체2                     | 다른 응용 프로그램이 사용 가능한 공급자 발견                          |
| 12 | 상    | 수신기                        | 수신기 취약                                                           |
| 13 | 상    | 수신기2                       | 다른 응용 프로그램이 사용 가능한 수신기 발견                          |
| 14 | 상    | 계정 가져오기 권한            | 계정 취약                                                             |
| 15 | 상    | 계정 가져오기 권한2           | 전화로 알려진 계정 목록 취득 가능                                     |
| 16 | 상    | 인터넷 허가                   | 인터넷 허가 취약                                                      |
| 17 | 상    | 인터넷 허가2                  | 앱이 네트워크 소켓 및 사용자 지정 네트워크 프로토콜 사용 가능         |
| 18 | 상    | 연락처 읽기 권한              | 연락처 취약                                                           |
| 19 | 상    | 연락처 읽기 권한2             | 앱이 전화에 저장된 연락처 읽기 허용                                   |
| 20 | 상    | 외부 저장소 읽기              | 외부 저장소 읽기 취약                                                 |
| 21 | 상    | 외부 저장소 읽기2             | 앱이 SD카드 내용 읽기 허용                                            |
| 22 | 상    | 외부 저장소 쓰기              | 외부 저장소 쓰기 취약                                                 |
| 23 | 상    | 외부 저장소 쓰기2             | 앱이 SD카드 내용 쓰기 허용                                            |
| 24 | 상    | Base64 인코딩                 | Base64 인코딩 사용                                                    |
| 25 | 상    | Base64 인코딩2                | 안전하지 않은 인코딩 사용                                             |
| 26 | 상    | Base64 디코드                 | Base64 디코드 사용                                                    |
| 27 | 상    | Base64 디코드2                | 안전하지 않은 디코드 사용                                             |
| 28 | 상    | SMS 체크                      | SmsManager 사용                                                       |
| 29 | 상    | SMS 체크2                     | 사용자 몰래 sms,mms 전송 허가                                         |
| 30 | 상    | catch의 일반 예외             | catch 예약 취약                                                       |
| 31 | 상    | catch의 일반 예외2            | 구체적이지 않은 예외 포착                                             |
| 32 | 상    | 로그 확인                     | 민감정보 로그 취약                                                    |
| 33 | 상    | 로그 확인2                    | 로그에서 확인되지 않은 출력 확인                                      |
+----+-------+-------------------------------+-----------------------------------------------------------------------+
33 rows in set (0.00 sec)
```

< DB 결과 - ② 발견된 취약점 및 내용 >

4. 결론

모바일 앱 사용이 지속적으로 증가하고 있고 그에 따른 앱 보안 위협이 증대되고 있다. 안드로이드 앱의 절반 정도가 해킹 위험에 노출되어 있다는 사실을 통해 관리 부실의 심각성을 깨닫고 안드로이드 앱 취약점을 분석할 수 있는 툴을 구현하였다.

안드로이드 단말기에서 APK 파일을 변환 및 압축해제, OWASP Mobile Top 10을 기반으로 한 취약점 자동화 분석 프로그램을 GUI로 개발함으로써 시간의 절약과 접근성을 높였

다. HTML과 엑셀을 이용한 결과 출력 시스템 구현을 통해 편의성의 증대 효과를 기대한다. 또한 DB를 이용하여 보고서들을 관리하고 최신 취약점을 업데이트 할 수 있도록 구현하였다. 일반 사용자들이 흔히 사용하는 앱을 자체 개발한 툴로 진단함으로써 보안에 대한 경각심을 일깨우는 계기 또한 마련될 것으로 기대되어진다.

5. 별첨

5.1 프로그램 소스 코드

5.2 발표 PPT

5.1 프로그램 소스 코드

5.1.1 Main System

<Main Source>

```
package Ha;

import java.io.*;
import java.util.Scanner;
import java.util.zip.ZipEntry;
import java.util.zip.ZipInputStream;
import java.net.URI;
import java.net.URISyntaxException;
import java.awt.Desktop;

public class Dajava {
    public static void createfile(){
        String path = "C:₩₩DaJaVa";
        File folder = new File(path);

        if(!folder.exists()) {
            try {
                folder.mkdir();
                System.out.println("DaJaVa 폴더 생성 되었습니다.");
            }
            catch(Exception e) {
                e.getStackTrace();
            }
        }
        else {
            System.out.println("폴더가 있습니다.");
        }
    }
    public static void Move(){

        try {
            ProcessBuilder builder = new ProcessBuilder( "cmd.exe", "/c", "move
C:₩₩dex2jar-2.0.zip C:₩₩DaJaVa₩₩dex2jar-2.0.zip");
            builder.redirectErrorStream(true);
            Process p = builder.start();
            BufferedReader r = new BufferedReader(new
InputStreamReader(p.getInputStream()));
            Scanner s = new Scanner(p.getInputStream());
            while(true) {
                String line = r.readLine();
                if (line == null) {break;}
                System.out.println(line);
            }
            System.out.println("파일 이동 성공");
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("파일 이동 실패");
        }
```

```
    }
    //public static class GetAPK{
        public static StringBuffer buff, readbuff;
        public static Process p;
        public static BufferedReader buffreader;

        public static String inputCommand(String cmd) {
            buff = new StringBuffer();

            buff.append("cmd.exe ");
            buff.append("/c ");
            buff.append("adb shell ");
            buff.append(cmd);

            return buff.toString();
        }
        public static String resultCommand(String cmd){
            try{
                p = Runtime.getRuntime().exec(cmd);
                buffreader = new BufferedReader(new
InputStreamReader(p.getInputStream()));
                String line = null;
                readbuff = new StringBuffer();

                while((line = buffreader.readLine()) != null){
                    if(line.indexOf("/data/app") > -1) {
                        readbuff.append(line);
                        readbuff.append("₩n");
                    }
                }
                return readbuff.toString();
            }
            catch (Exception e){
                e.printStackTrace();
                System.exit(1);
            }
            return  null;
        }
        public String downloadFile(String apk, String fileName){
            buff = new StringBuffer();
            buff.append("cmd.exe ");
            buff.append("/c ");
            buff.append("adb pull ");
            buff.append(apk);
            buff.append(" C:₩₩DaJaVa₩₩dex2jar-2.0₩₩");
            buff.append(fileName);

            try {
                p = Runtime.getRuntime().exec(buff.toString());
                buffreader = new BufferedReader(new
InputStreamReader(p.getInputStream()));
                String arr = null;
```

```
                readbuff = new StringBuffer();

                while ((arr = buffreader.readLine()) != null) {
                    readbuff.append(arr);
                    readbuff.append("₩n");
                }
                return readbuff.toString();
            }
            catch (Exception e){
                e.printStackTrace();
                System.exit(1);
            }
            return null;
        }

   // }
    public static String XmlFunc(String args) {
        String[] cmd = { "cmd", "/c",  "cd c:₩₩DaJaVa && apktool d -f", args};
        Process process = null;
        try {
            process = new ProcessBuilder(cmd).start();

            Scanner s = new Scanner(process.getInputStream(), "EUC-KR");
            while (s.hasNextLine() == true) {
                System.out.println(s.nextLine());
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return  "c:₩₩DaJaVa₩₩"+args+"₩₩AndroidManifest.xml";  //xml경로를  String으로
return
        //return "C:₩₩DaJaVa₩₩dex2jar-2.0₩₩AndroidManifest.xml"; //xml경로를 String으로
return
    }

    public static String ApkToZip(String FileName){
        File file = new File("C:₩₩DaJaVa₩₩dex2jar-2.0",FileName);
        int pos = FileName.lastIndexOf(".");
        String filename = FileName.substring(0, pos);
        StringBuilder _filename = new StringBuilder(filename);
        _filename.append(".zip");
        filename = _filename.toString();
        file.renameTo(new File("C:₩₩DaJaVa₩₩dex2jar-2.0₩₩",filename));
        System.out.println("zip 변환 성공");

        return filename;
    }

    public static class ZipDecode{
        public static void decompress(String zipFileName, String directory) throws
Throwable {
            File zipFile = new File("C:₩₩DaJaVa₩₩dex2jar-2.0₩₩",zipFileName);
```

```
            FileInputStream fis = null;
            ZipInputStream zis = null;
            ZipEntry zipentry = null;
            try {

                fis = new FileInputStream(zipFile);
                zis = new ZipInputStream(fis);
                while ((zipentry = zis.getNextEntry()) != null) {
                    String filename = zipentry.getName();
                    File file = new File(directory, filename);
                    if (zipentry.isDirectory()) {
                        file.mkdirs();
                    } else {
                        //파일이면 파일 만들기
                        createFile(file, zis);
                    }
                }
            } catch (Throwable e) {
                throw e;
            } finally {
                if (zis != null)
                    zis.close();
                if (fis != null)
                    fis.close();
            }
        }
        private static void createFile(File file, ZipInputStream zis) throws Throwable {
            //디렉토리 확인
            File parentDir = new File(file.getParent());
            //디렉토리가 없으면 생성
            if (!parentDir.exists()) {
                parentDir.mkdirs();
            }
            //파일 스트림 선언
            try (FileOutputStream fos = new FileOutputStream(file)) {
                byte[] buffer = new byte[256];
                int size = 0;

                while ((size = zis.read(buffer)) > 0) {
                    //byte로 파일 만들기
                    fos.write(buffer, 0, size);
                }
            } catch (Throwable e) {
                throw e;
            }
        }
    }

    public static String DexToJar(String dexName){
        try {
            ProcessBuilder builder = new ProcessBuilder(
                    "cmd.exe", "/c", "cd ₩"C:₩₩DaJaVa₩₩dex2jar-2.0₩" &&
```

```
d2j-dex2jar.bat ",dexName);
                builder.redirectErrorStream(true);
                Process p = builder.start();
                BufferedReader r = new BufferedReader(new
InputStreamReader(p.getInputStream()));
                while (true) {
                    String line;
                    line = r.readLine();
                    if (line == null) {break;}
                    System.out.println(line);
                } System.out.println("dex -> jar 파일변환 성공");
            }
            catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        }

        public static class JarDecode {
            public static void JarFunc(String args) {
                StringBuilder jarname = new StringBuilder(args);
                jarname.append("-dex2jar.jar");
                args = jarname.toString();
                String[] cmd = { "cmd", "/c","cd C:₩₩DaJaVa₩₩dex2jar-2.0 && jar xvf ",args};
                Process process = null;
                try {
                    process = new ProcessBuilder(cmd).start();
                    Scanner s = new Scanner(process.getInputStream(), "EUC-KR");
                    while (s.hasNextLine() == true) { System.out.println(s.nextLine()); }
                    System.out.println("Jar파일 압축해제 성공");
                } catch (IOException e) { e.printStackTrace(); }
            }
        }
        //class 리스트 목록
        public static class ClassList {
            public static String[] classFile() {
                String[] classfilelist = new String[100];
                File path = new File("C:₩₩DaJaVa₩₩InsecureBackv2-classes");
                File[] fileList = path.listFiles();

                if (fileList.length > 0) {
                    for (int i = 0; i < fileList.length; i++) {
                        classfilelist[i] = String.valueOf(fileList[i]);
                        System.out.println(classfilelist[i]);
                    }

                }
                return classfilelist;
            }
        }

        //난독화 체크
```

```
public static class CheckObfuscation {
    public static int CheckFunction(String classPath) {
        int cnt = 0;
        try {
            File f = new File(classPath);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String input = "void";
            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);

                    if (word.length() <= 2) {
                        //System.out.println("난독화 설정 O");
                        cnt  = 0;
                    }
                    else cnt = 1; //System.out.println("난독화 설정 X");
                } }
            bufr.close();
        }
        catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } return cnt;
    }
}

public static int Base64Encode(String path) {
//  Base64 인코딩 (CryptoClass.java 에서 확인함)
//  Base64 인코딩 방법은 안전한 인코딩 방법이 아님
int num = 0;
try {
    String check = "Base64.encode";

    File f = new File(path);
    FileReader fr = new FileReader(String.valueOf(f));
    BufferedReader bufr = new BufferedReader(fr);
    String line = "";
    while ((line = bufr.readLine()) != null) {
        if (line.contains(check))
            System.out.println("Base64 인코딩 사용 -> 취약");
        num++;
    }
    bufr.close();
} catch (FileNotFoundException e) {
    System.out.println("파일을 찾을 수 없음");
} catch (IOException e) {
    System.out.println(e);
```

```
    }
    if(num>0) return 1;
    else return 0;
}

    public static int Base64Decode(String path) {
        //  Base64 디코드 (CryptoClass.java 에서 확인함)
        int num = 0;
        try {
            String check = "Base64.decode";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("Base64 디코드 사용 -> 취약");
                num++;
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0) return 1;
        else return 0;
    }



    public static void main(String[] args){
    //     GetAPK cmd = new GetAPK();
        XmlChecks checkxml = new XmlChecks();
        String apk = new String();
        String ApkName = new String();
        String apk_to_zip = new String();
        String[] filelist;
        int cnt;
        //Create DaJaVa File
        createfile();
        //move dex2 tool zip file
        Move();
        //dex2.zip decode
        try{
            String zipfile = "dex2jar-2.0.zip";
            ZipDecode.decompress(zipfile, "C:₩₩DaJaVa₩₩");
            System.out.println("zip파일 압축해제 성공");
        }catch(Throwable e){
            e.printStackTrace();
        }
```

```
        //Get ApK
        /*String order = cmd.inputCommand("pm list packages -f");
        String result = cmd.resultCommand(order);

        System.out.println(result);
        System.out.println("Write apk [예) InsecureBankv2.apk /  .apk 까지 복붙] :");
        Scanner apkScan = new Scanner(System.in);
        apk = apkScan.next();
        System.out.println("Write file name [예) InsecureBankv2.apk] :");
        Scanner fileNameScan = new Scanner(System.in);
        ApkName = fileNameScan.next();

        String download = cmd.downloadFile(apk, ApkName);
        System.out.println(download);*/
        //XML decode

        //apk to zip
        apk_to_zip = ApkToZip(ApkName);
        //application zip decode
        try{
            ZipDecode.decompress(apk_to_zip, "C:₩₩DaJaVa₩₩dex2jar-2.0");
            System.out.println("zip파일 압축해제 성공");
        }catch(Throwable e){
            e.printStackTrace();
        }
        //dex to jar
        String DexName = "classes.dex";
        String dex_to_jar;
        dex_to_jar = DexToJar(DexName);

        JarDecode.JarFunc("classes"); // jar 파일 압축 풀기 함수선언

        filelist = ClassList.classFile(); // .class 경로 넣기 함수 선언
        for(int i = 0; i < filelist.length; i++) {
            if(filelist[i] != null) {
                String classpath = filelist[i];
                cnt = CheckObfuscation.CheckFunction(classpath); // 난독화 체크
                if(cnt == 1) System.out.println("난독화 설정 O");
                if(cnt == 0) System.out.println("난독화 설정 X"); break;

            }
        }
        String XmlPath;
        XmlPath = XmlFunc(ApkName);
        System.out.println(XmlPath);
        //XML Check
        String                                          path                                    =
"C:₩₩DaJaVa₩₩dex2jar-2.0₩₩InsecureBankv2₩₩AndroidManifest.xml";
        //=============================================
        int[] ResultArray = new int[50];

        ResultArray[0]=checkxml.Manifest_Debug(path);
```

```
            ResultArray[1]=checkxml.Manifest_Backup(path);
            ResultArray[2]=checkxml.Manifest_Location(path);
            ResultArray[3]=checkxml.Manifest_Phone(path);
            ResultArray[4]=checkxml.Manifest_Activity(path);
            ResultArray[5]=checkxml.find(path);
            System.out.println(ResultArray);
            //===== 추가한 부분 10.21
            checkxml.Manifest_Provider(path);
            checkxml.Manifest_Receiver(path);
            checkxml.Manifest_Accounts(path);
            checkxml.Manifest_Internet(path);
            checkxml.Manifest_Contacts(path);
            checkxml.Manifest_Rstorage(path);
            checkxml.Manifest_Wstorage(path);
            //class 취약점 점검
            DoCheck.ClassCheck();

            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result (FileName,Date,Report)" + "VALUES('" +apk+"',
DEFAULT,' C:₩₩DaJaVa') ");
            //ot.update("delete from result where Name=' 취약점A'" );
            ot.select("select * from result");

            try {
                  ot.rs.beforeFirst();
                  while(ot.rs.next()) {

System.out.println(ot.rs.getString("No")+ot.rs.getString("FileName")+ot.rs.getString("Date")+ot.r
s.getString("Report"));

                  }
            }
            catch(Exception e) {
                  System.out.println("getString error:" + e);
            }
            ot.close();
            try {

                  Desktop.getDesktop().browse(new URI("http://localhost/index1.php"));


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

            Excel.Excel();
      }
}
```

<OpenTable>

```
package Ha;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class OpenTable {
    java.sql.Connection con = null;
    // 데이터베이스와 연결을 위한 객체
    java.sql.Statement stmt = null;
    // SQL문을 데이터베이스에 보내기위한 객체
    java.sql.ResultSet rs = null;
    // SQL문 질의에 의해 생성된 테이블을 저장하는 객체

    String driver = "com.mysql.jdbc.Driver";
    //JDBC Driver Class = com.mysql.jdbc.Driver

    OpenTable(){
        connect();
    }

    void connect() {

        String url = "jdbc:mysql://localhost:3306/test";
        String user = "root";
        String pw = "1111";

        // 데이터베이스에 연결하기위한 정보

        try {
            Class.forName(driver);
            this.con = java.sql.DriverManager.getConnection(url, user, pw);
            this.stmt = con.createStatement();
        }
        catch(Exception e)
        {
            System.out.println("connection error:" +e);
        }
    }

    void update(String dbCommand) {
        try {
            this.stmt.executeUpdate(dbCommand);
        }
        catch(Exception e) {
            System.out.println("update error:"+e);
        }
    }
    void select(String dbSelect) {
        try {
```

```
            this.rs = this.stmt.executeQuery(dbSelect);
        }
        catch(Exception e) {
            System.out.println("select error:"+e);
        }
    }

    void close() {
        try {
            con.close();
        }
        catch(Exception e) {
            System.out.println("close error:"+e);
        }
    }
}
```

<Excel>

```
package Ha;

import org.apache.poi.hssf.usermodel.HSSFCellStyle;
import org.apache.poi.hssf.usermodel.HSSFFont;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.hssf.util.HSSFColor;
import org.apache.poi.ss.usermodel.*;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;
import java.util.Scanner;

public class Excel {
    public static void Excel() {
        Connection con = null;

        String server = "localhost"; // MySQL 서버 주소
        String database = "test"; // MySQL DATABASE 이름
        String user_name = "root"; //  MySQL 서버 아이디
        String password = "1111"; // MySQL 서버 비밀번호

        // 1.드라이버 로딩
        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            System.err.println(" !! <JDBC 오류> Driver load 오류: " + e.getMessage());
            e.printStackTrace();
        }

        // 2.연결
        try {
```

```
            con = DriverManager.getConnection("jdbc:mysql://" + server + "/" +
database + "?useSSL=false", user_name, password);
            System.out.println("정상적으로 연결되었습니다.");

            String qu = "select * from result2";
            String qu2 = "select * from result";
            Statement st = con.createStatement();
            Statement st2 = con.createStatement();
            ResultSet rs = st.executeQuery(qu);
            ResultSet rs2 = st2.executeQuery(qu2);

            // 3. 엑셀 작성 시작
            HSSFWorkbook workbook = new HSSFWorkbook();

            Row row = null;
            Cell cell = null;
            int rowCount = 0;
            int cellCount = 0;

            //Sheet 생성
            Sheet sheet = workbook.createSheet("Reporting");
            sheet.setColumnWidth(0, 20 * 256);
            sheet.setColumnWidth(1, 5 * 256);
            sheet.setColumnWidth(2, 30 * 256);
            sheet.setColumnWidth(3, 75 * 256);

            //해더부분셀에 스타일을 주기위한 인스턴스 생성
            HSSFCellStyle cellStyle = workbook.createCellStyle();
            cellStyle.setBorderBottom(BorderStyle.THIN);
            cellStyle.setBorderLeft(BorderStyle.THIN);
            cellStyle.setBorderRight(BorderStyle.THIN);
            cellStyle.setBorderTop(BorderStyle.THIN);
            cellStyle.setAlignment(HorizontalAlignment.CENTER);

cellStyle.setFillForegroundColor(HSSFColor.HSSFColorPredefined.LIGHT_ORANGE.getInd
ex());
            cellStyle.setFillPattern(FillPatternType.SOLID_FOREGROUND);

            HSSFCellStyle cellStyle2 = workbook.createCellStyle();
            cellStyle2.setAlignment(HorizontalAlignment.CENTER);

            HSSFCellStyle cellStyle3 = workbook.createCellStyle();
            cellStyle3.setAlignment(HorizontalAlignment.RIGHT);

            HSSFFont f = workbook.createFont();
            f.setColor(Font.COLOR_RED);
            HSSFCellStyle font = workbook.createCellStyle();
            font.setFont(f);
            font.setAlignment(HorizontalAlignment.CENTER);


            row = sheet.createRow(rowCount++);
```

```
cellCount=3;
String FileName = new String();
while(rs2.next()) {
    FileName = rs2.getString("FileName");
    String Date = rs2.getString("Date");

    cell = row.createCell(cellCount++);
    cell.setCellValue(FileName + "  /  " + Date);
    cell.setCellStyle(cellStyle3); }

//      rowCount++;                              //첫줄 개행
row = sheet.createRow(rowCount++);  //열 생성
cellCount=0;

//셀 생성
cell = row.createCell(cellCount++);
cell.setCellValue("점검 항목명");
cell.setCellStyle(cellStyle);

cell = row.createCell(cellCount++);
cell.setCellValue("등급");
cell.setCellStyle(cellStyle);

cell = row.createCell(cellCount++);
cell.setCellValue("상태");
cell.setCellStyle(cellStyle);

cell = row.createCell(cellCount++);
cell.setCellValue("권유 조치 방안");
cell.setCellStyle(cellStyle);

//2열 작성
while(rs.next()) {
    String Name = rs.getString("Name");
    String Grade = rs.getString("Grade");
    String Content = rs.getString("Content");
    // System.out.format("%s, %s, %s \r\n", Name, Grade, Content);

    row = sheet.createRow(rowCount++);
    cellCount = 0;

    cell = row.createCell(cellCount++);
    cell.setCellValue(Name);

    cell = row.createCell(cellCount++);
    cell.setCellValue(Grade);
    if(Grade.equals("상")) cell.setCellStyle(font);
    else cell.setCellStyle(cellStyle2);

    cell = row.createCell(cellCount++);
    cell.setCellValue(Content);
```

```
                cell = row.createCell(cellCount++);
                File file = new File("C:\\Users\\karin\\IdeaProjects\\Ha.Dajava\\src\\권유 조치 방안.txt");
            //경로 다자바로 변경
                BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(file),"euc-kr"));
                String line = "";
                while((line = br.readLine()) != null) {
                    if (line.contains(Name)) {
                        int a = line.indexOf(":");
                        String result = line.substring(a+1);
                        cell.setCellValue(result);
                    }
                }
            } st.close();

            String filename = FileName+" 점검 결과";
            FileOutputStream outFile = new FileOutputStream("C:\\Users\\karin\\Desktop\\" + filename + ".xls");
            System.out.println("Done");
            workbook.write(outFile);
            outFile.close();

        } catch (SQLException e) {
            System.err.println("con 오류:" + e.getMessage());
            e.printStackTrace();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        // 4. 해제
        try {
            if (con != null)
                con.close();
        } catch (SQLException e) {}
    }

    public static void main(String[] args) {
        Excel();
    }
}
```

5.1.2. Check System

<ClassCheck.>

```
package Ha;

import java.io.*;

public class ClassCheck {
    public static int Base64Encode(String path) {
        //  Base64 인코딩 (CryptoClass.java 에서 확인함)
        //  Base64 인코딩 방법은 안전한 인코딩 방법이 아님
        int num = 0;
        try {
            String check = "Base64.encode";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("Base64 인코딩 사용 -> 취약");
                num++;
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
Base64 인코딩',' Base64 인코딩 사용 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
Base64 인코딩2',' 안전하지 않은 인코딩 사용') ");
            return 1;
        }
        else return 0;
    }

    public static int Base64Decode(String path) {
        //  Base64 디코드 (CryptoClass.java 에서 확인함)
        int num = 0;
        try {
            String check = "Base64.decode";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
```

```
                    System.out.println("Base64 디코드 사용 -> 취약");
                num++;
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
Base64 디코드',' Base64 디코드 사용 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
Base64 디코드2',' 안전하지 않은 디코드 사용 ') ");
            return 1;
        }
        else return 0;
    }

    public static int SMS(String path) {
        //  사용자 몰래 sms,mms 보내는지 체크 (MyBroadCastReceiver.java에서 확인함)
        int num = 0;
        try {
            String check = "SmsManager";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("SmsManager 사용 -> 취약");
                num++;
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', ' SMS
체크',' SmsManager 사용 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', ' SMS
체크2',' 사용자 몰래 sms,mms 전송 허가 ') ");
            return 1;
        }
        else return 0;
    }
```

```
    public static int Catch(String path) {
        //  catch의 일반 예외
        //예외 포착은 구체적이어야합니다. 일반 예외 유형이 안전하지 않아 자동 오류 방지로 이어짐
        int num = 0;
        try {
            String check = "catch (Exception paramContext)";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("catch 예약 취약");
                num++;
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', ' catch의 일반 예외',' catch 예약 취약 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', ' catch의 일반 예외2',' 구체적이지 않은 예외 포착 ') ");
            return 1;
        }
        else return 0;
    }

    public static int Log(String path) {
        //  로그에서 확인되지 않은 출력
        //  민감한 정보는 정보가 공개 될 수 있으므로 기록해서는 안됩니다.
        int num = 0;
        try {
            String check = "username + ₩":₩" + DoLogin.this.password)";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("민감정보 로그 취약");
                num++;
            }
            bufr.close();
        } catch (FileNotFoundException e) {
```

```
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', ' 로그
확인',' 민감정보 로그 취약 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', ' 로그
확인2',' 로그에서 확인되지 않은 출력 확인 ') ");
            return 1;
        }
        else return 0;
    }

    public static int findExternalStorage(String path){
        //Class파일에서 외부 저장소에서 쓰기 읽기 취약점
        String word = "(?i).*getExternalStorage.*";
        int num=0; int num1=0;
        try {
            BufferedReader reader = new BufferedReader(new FileReader(path));
            String s;

            int line =1;

            while ((s = reader.readLine()) != null) {
                if (s.matches(word)){
                    //System.out.println(line + " : " + s);
                    num++;
                }
                line++; }
        } catch (FileNotFoundException e) { e.printStackTrace(); }
        catch (IOException e) { e.printStackTrace(); }
        if(num>0){
            System.out.println("외부저장소 쓰기 읽기 사용 -> 취약"); num1++;
                OpenTable ot = new OpenTable();
                ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
외부저장소 쓰기 읽기',' 외부저장소의 중요한 정보 노출 가능 ') ");

            return 1; }
        else
            return 0;

    }

    public static int PostLogin_Rooting(String path2) {
        // 루팅탐지 (PostLogin.java 파일에서 확인 가능)
        // 사용자가 Root권한을 획득하여 시스템을 사용가능
        int num = 0;
        try {
            int i = 0;
            int count = 0;
```

```
            String check = "/system/app/Superuser.apk";

            File f = new File(path2);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    // 라인 중 /system/app/Superuser.apk 포함한 라인 찾음
                    i++;
                count = i;
            }
            if (i > 0)
            {
                System.out.println("노출된 루팅 갯수 : "+count+" -> 취약");
                num++;
                if(num>0){
                    OpenTable ot = new OpenTable();
                    ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES('
상', ' 루팅탐지',' 노출된 루팅 갯수 : "+count+"') ");
                    ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES('
상', ' 루팅탐지2',' Root권한 획득 가능') ");
                }
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0)
            return 1;
        else
            return 0;
    }

    public static int PostLogin_Su(String path2) {
        // Su 권한 확인 (PostLogin.java 파일에서 확인 가능)
        // 응용 프로그램이 시스템 명령을 실행할 수 있습니다.
        int num = 0;
        try {
            String check = "/system/xbin/which₩", ₩"su";

            File f = new File(path2);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    // 라인 중 "/system/xbin/which", "su" 포함한 라인 찾음
                    System.out.println("시스템 명령 실행중 -> 취약");
                num++;
```

```
                }
                bufr.close();
            } catch (FileNotFoundException e) {
                System.out.println("파일을 찾을 수 없음");
            } catch (IOException e) {
                System.out.println(e);
            }
            if(num>0){
                OpenTable ot = new OpenTable();
                ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', ' Su
권한',' 시스템 명령 실행중 ') ");
                ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', ' Su
권한2',' 응용 프로그램이 시스템 명령 실행 가능 ') ");
                return 1;
            }
            else
                return 0;
        }


}
```

<DoCheck.>

```
package Ha;

public class DoCheck {
    public static void ClassCheck(){
        String[] filelist;
        int[] a = new int[20];
        filelist = Dajava.ClassList.classFile();
        for(int i = 0; i < filelist.length; i++) {

            if(a[0]==0) a[0] =ClassCheck.Base64Decode(filelist[i]);
            if(a[1]==0) a[1] =ClassCheck.Base64Encode(filelist[i]);
            if(a[2]==0) a[2] =ClassCheck.SMS(filelist[i]);
            if(a[3]==0) a[3] =ClassCheck.Catch(filelist[i]);
            if(a[4]==0) a[4] =ClassCheck.Log(filelist[i]);
            if(a[5]==0) a[5] =ClassCheck.findExternalStorage(filelist[i]);
            if(a[6]==0) a[6] =ClassCheck.PostLogin_Rooting(filelist[i]);
            if(a[7]==0) a[7] =ClassCheck.PostLogin_Su(filelist[i]);
        }
    }
}
```

<XmlCheck>

```
package Ha;

import java.io.*;
public class XmlChecks {
```

```
    public static int Manifest_Debug(String path) {
        // 디버그 모드 확인 (AndroidManifest.xml 파일에서 확인 가능)
        // 디버그 모드면 외부에서 임의의 코드를 주입 가능, 개발시에만 사용해야함
        int num = 0;
        int weak = 0;
        try {
            String check = "android:debuggable=\"true\"";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    // 라인 중 android:debuggable="true" 포함한 라인 찾음
                    System.out.println("디버그 모드 작동 중 -> 취약");
                num++;
                if (num > 0) {
                    OpenTable ot = new OpenTable();
                    ot.update("INSERT INTO result2 (Grade,Name,Content)" +
"VALUES(' 상', ' 디버그 모드 확인',' 디버그 모드 작동 중 ') ");
                    break;

                }
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0)

            return 1;
        else
            return 0;
    }

    public static int Manifest_Backup(String path) {
        // 백업 허용 확인 (AndroidManifest.xml 파일에서 확인 가능)
        // 악의적 사용자가 adb를 사용하여 앱의 개인 데이터를 자신의 PC에 가져올 수 있
음.
        int num = 0;
        try {
            String check = "android:allowBackup=\"true\"";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
```

```
                    // 라인 중 android:allowBackup="true" 포함한 라인 찾음
                    System.out.println("adb를 통해 백업 허용 중 -> 취약");
                num++;
                if (num > 0) {
                    OpenTable ot = new OpenTable();
                    ot.update("INSERT INTO result2 (Grade,Name,Content)" +
"VALUES(' 상', ' 백업 허용 확인',' adb를 통해 백업 허용 ') ");
                    break;

                }
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0)
            return 1;
        else
            return 0;
    }

    public static int Manifest_Location(String path) {
        // 사용자의 위치 정보 접근 확인 (AndroidManifest.xml 파일에서 확인 가능)
        // Wi-Fi와 같은 네트워크 위치 소스를 통해 위치 얻어짐. 기본 권한으로 설정되어
있으면 위험함.
        int num = 0;
        try {
            String check = "android.permission.ACCESS_COARSE_LOCATION";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    // 라인 중 android.permission.ACCESS_COARSE_LOCATION 포함한
라인 찾음
                    System.out.println("위치 정보 접근 허용 -> 취약");
                num++;
                if (num > 0) {
                    OpenTable ot = new OpenTable();
                    ot.update("INSERT INTO result2 (Grade,Name,Content)" +
"VALUES(' 상', ' 사용자의 위치 정보 접근 확인',' 위치 정보 접근 허용 ') ");
                    break;

                }
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
```

```
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0)
            return 1;
        else
            return 0;
    }

    public static int Manifest_Phone(String path) {
        // Phone 권한 확인 (AndroidManifest.xml 파일에서 확인 가능)
        // 장치의 전화번호, 네트워크정보, 진행중인 통화의 상태 읽어오기, 사용자의 통화 기
록 읽기 가능
        int num = 0;
        try {
            String check = "android.permission.READ_CALL_LOG";
            String check2 = "android.permission.READ_PHONE_STATE";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    // 라인 중 android.permission.READ_CALL_LOG 포함한 라인 찾음
                    System.out.println("사용자의 통화 기록 읽기 허용 -> 취약");
                num++;
                if (num > 0) {
                    OpenTable ot = new OpenTable();
                    ot.update("INSERT INTO result2 (Grade,Name,Content)" +
"VALUES(' 상', ' Phone 권한 확인',' 장치의 전화 번호, 네트워크 정보, 진행중인 통화의 상
태 읽어오기 허용 ') ");
                    break;

                }
                if (line.contains(check2))
                    // 라인 중 android.permission.READ_PHONE_STATE 포함된 라인 찾
음
                    System.out.println("장치의 전화 번호, 네트워크 정보, 진행중인 통화의
상태 읽어오기 허용 -> 취약");
                num++;
                if (num > 0) {
                    OpenTable ot = new OpenTable();
                    ot.update("INSERT INTO result2 (Grade,Name,Content)" +
"VALUES(' 상', ' Phone 권한 확인',' 사용자의 통화 기록 읽기 허용 ') ");
                    break;
                    //String weak1 = "VALUES(' 상', ' 디버그 모드 확인',' 디버그 모드
작동 중 ') ";
                }

            }
            bufr.close();
```

```
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0)
            return 1;
        else
            return 0;
    }

    public static int Manifest_Activity(String path) {
        // 액티비티 컴포넌트 확인 (AndroidManifest.xml 파일에서 확인 가능)
        // 다른 에플리케이션의 Activity를 실행 할 수 있음.
        int num = 0;
        try {
            int i = 0;
            int count = 0;
            String check = "activity android:exported=\"true\"";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {

                if (line.contains(check))
                    // 라인 중 android:exported="true" 포함한 라인 찾음
                    //System.out.println("다른 애플리케이션의 Activity 실행가능 -> 취약
");
                    i++;
                count = i;
            }
            if (i > 0)
            {
                System.out.println("노출된 Activity 갯수 : "+count+" -> 취약");
                num++;
                if (num > 0) {
                    OpenTable ot = new OpenTable();
                    ot.update("INSERT INTO result2 (Grade,Name,Content)" +
"VALUES(' 상', ' 액티비티 컴포넌트 확인',' 노출된 Activity 갯수 :"+ count +"') ");
                }
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0)
            return 1;
        else
```

```
            return 0;
    }

    public static int find(String path){
        //xml파일에서 provider 검색 함수
        String word = "(?i).*provider.*";
        int num=0;
        try {
            BufferedReader reader = new BufferedReader(new FileReader(path));
            String s;

            int line =1;

            while ((s = reader.readLine()) != null) {
                if (s.matches(word)){
                    System.out.println(line + " : " + s);
                    num++;
                }
                line++; }
        } catch (FileNotFoundException e) { e.printStackTrace(); }
        catch (IOException e) { e.printStackTrace(); }
        if(num>0)
            return 1; //content provide 취약점이 있으면 1 return
        else
            return 0;
    }

    public static int Manifest_Provider(String path) {
        // 수출 업체
        // 내보낸 공급자를 찾았습니다. 다른 응용 프로그램에서 사용할 수 있습니다.
        int num = 0;
        try {
            String check = "<provider";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("수출 업체 취약"); }
            num++;
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0) {
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 중', '
수출 업체',' 수출 업체 취약 ') ");
```

```
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 중', '
수출 업체2',' 다른 응용 프로그램이 사용 가능한 공급자 발견 ') ");
            return 1;
        }
        else
            return 0;
    }

    public static int Manifest_Receiver(String path) {
        // 수출 수신기
        // 내보낸 수신기를 찾았습니다. 다른 응용 프로그램에서 사용할 수 있습니다.
        int num = 0;
        try {
            String check = " <receiver";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("수신기 취약");
                num++;
            }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 중', '
수출 수신기',' 수신기 취약 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 중', '
수출 수신기2',' 다른 응용 프로그램이 사용 가능한 수신기 발견 ') ");
            return 1;
        }
        else
            return 0;
    }

    public static int Manifest_Accounts(String path) {
        // 계정 가져 오기 권한
        // 앱이 전화로 알려진 계정 목록을 가져올 수 있도록 허용합니다. 여기에는 설치 한
응용 프로그램에서 만든 계정이 포함될 수 있습니다. 권한이 실제로 필요한지 확인하십시오.
        int num = 0;
        try {
            String check = "android.permission.GET_ACCOUNTS";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
```

```
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("계정 취약");
                num++; }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
계정 가져오기 권한',' 계정 취약 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
계정 가져오기 권한2',' 전화로 알려진 계정 목록 취득 가능 ') ");
            return 1;
        }
        else
            return 0;
    }

    public static int Manifest_Internet(String path) {
        // 인터넷 허가
        // 앱이 네트워크 소켓을 만들고 사용자 지정 네트워크 프로토콜을 사용할 수 있도록
허용합니다. 브라우저 및 기타 응용 프로그램은 인터넷으로 데이터를 전송하는 수단을 제공하므
로 인터넷으로 데이터를 전송하는 데이 권한이 필요하지 않습니다. 권한이 실제로 필요한지 확
인하십시오.
        int num = 0;
        try {
            String check = "android.permission.INTERNET";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("인터넷 허가 취약");
                num++;      }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
인터넷 허가',' 인터넷 허가 취약 ') ");
```

```
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
인터넷 허가2',' 앱이 네트워크 소켓 및 사용자 지정 네트워크 프로토콜 사용 가능 ') ");
            return 1;
        }
        else
            return 0;
    }

    public static int Manifest_Contacts(String path) {
        // 연락처 읽기 권한
        // 앱이 전화, 이메일 또는 특정 방식으로 다른 사람과 통신 한 빈도를 비롯하여 휴대
전화에 저장된 연락처에 대한 데이터를 읽을 수 있도록 허용합니다. 이 권한을 통해 앱은 연락
처 데이터를 저장할 수 있으며 악성 앱은 사용자 모르게 연락처 데이터를 공유 할 수 있습니
다.
        int num = 0;
        try {
            String check = "android.permission.READ_CONTACTS";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("연락처 취약");
                num++; }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
연락처 읽기 권한',' 연락처 취약 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
연락처 읽기 권한2',' 앱이 전화에 저장된 연락처 읽기 허용 ') ");
            return 1;
        }
        else
            return 0;
    }

    public static int Manifest_Rstorage(String path) {
        // 외부 저장소 읽기
        // 앱이 SD 카드의 내용을 읽을 수 있도록 허용합니다
        int num = 0;
        try {
            String check = "android.permission.READ_EXTERNAL_STORAGE";

            File f = new File(path);
```

```
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("외부 저장소 읽기 취약");
                num++; }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
외부 저장소 읽기',' 외부 저장소 읽기 취약 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
외부 저장소 읽기2',' 앱이 SD카드 내용 읽기 허용') ");
            return 1;
        }
        else
            return 0;
    }

    public static int Manifest_Wstorage(String path) {
        // 외부 저장소 쓰기
        // 앱이 SD 카드에 쓸 수 있도록 허용합니다.
        int num = 0;
        try {
            String check = "android.permission.WRITE_EXTERNAL_STORAGE";

            File f = new File(path);
            FileReader fr = new FileReader(String.valueOf(f));
            BufferedReader bufr = new BufferedReader(fr);
            String line = "";
            while ((line = bufr.readLine()) != null) {
                if (line.contains(check))
                    System.out.println("외부 저장소 쓰기 취약");
                num++; }
            bufr.close();
        } catch (FileNotFoundException e) {
            System.out.println("파일을 찾을 수 없음");
        } catch (IOException e) {
            System.out.println(e);
        }
        if(num>0){
            OpenTable ot = new OpenTable();
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
외부 저장소 쓰기',' 외부 저장소 쓰기 취약 ') ");
            ot.update("INSERT INTO result2 (Grade,Name,Content)" + "VALUES(' 상', '
외부 저장소 쓰기2',' 앱이 SD카드 내용 쓰기 허용 ') ");
```

```
            return 1;
        }
        else
            return 0;
    }
}
```

5.1.3. GUI

<Java GUI>

```
package Hu;
import java.awt.Panel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.ScrollPaneConstants;

import Ha.Dajava;
import java.awt.BorderLayout;

public class Hu extends JFrame {


    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        new Hu();
    }

    /**
     * Create the frame.
     */
    public Hu() {
        super("안드로이드 앱 취약점 진단 (DAJAVA) v1.0");
        //setBounds(300,300,600,600);
        //setVisible(true);
        Dajava test1 = new Dajava();
        //GetAPK test = new GetAPK();
        Panel p = new Panel();
        p.setSize(350, 1500);
        JFrame f1 = new JFrame();
        JLabel lbl1 = new JLabel("다운받을 apk 입력 ( 예 : /data~test.apk) : ");
        JLabel lbl2 = new JLabel("저장할 이름 ( 예 : test.apk) : ");
```

```
        //JLabel lbl3 = new JLabel("안드로이드 앱 취약점 진단 (DAJAVA) v1.0");
        JLabel lbl4 = new JLabel("Apk List : ");
        JLabel lbl5 = new JLabel("중부대학교                                    정보보호학과
AVA4");

        JTextArea txt3 = new JTextArea(12,40);
        JTextField txt1 = new JTextField(40);
        JTextField txt2 = new JTextField(40);
        JScrollPane scroll = new JScrollPane(txt3);


scroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEED
ED);

scroll.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEV
ER);

        p.add(lbl4, BorderLayout.NORTH);
        //p.add(lbl4);
        String order = test1.inputCommand("pm list packages -f");
        String result = test1.resultCommand(order);

        txt3.setText(result);
        p.add(scroll);
        p.add(lbl1);
        p.add(txt1);

        JButton btnEn = new JButton("enter");
        btnEn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String apklink = txt1.getText();
                String filename = txt2.getText();
                String download = test1.downloadFile(apklink, filename);
            }
        });

        p.add(lbl2);
        p.add(txt2);
        p.add(btnEn);
        p.add(lbl5);
        getContentPane().add(p);


        //JButton btnNext = new JButton("next");

        //p.add(btnNext);
        setSize(500,400);
        setVisible(true);
    }

}
```

5.2. 발표 PPT


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
2

조원 편성

이 름	역 할
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

3

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 깐 앱 '스파이 장치' 사용자 절반 해킹위험 노출"

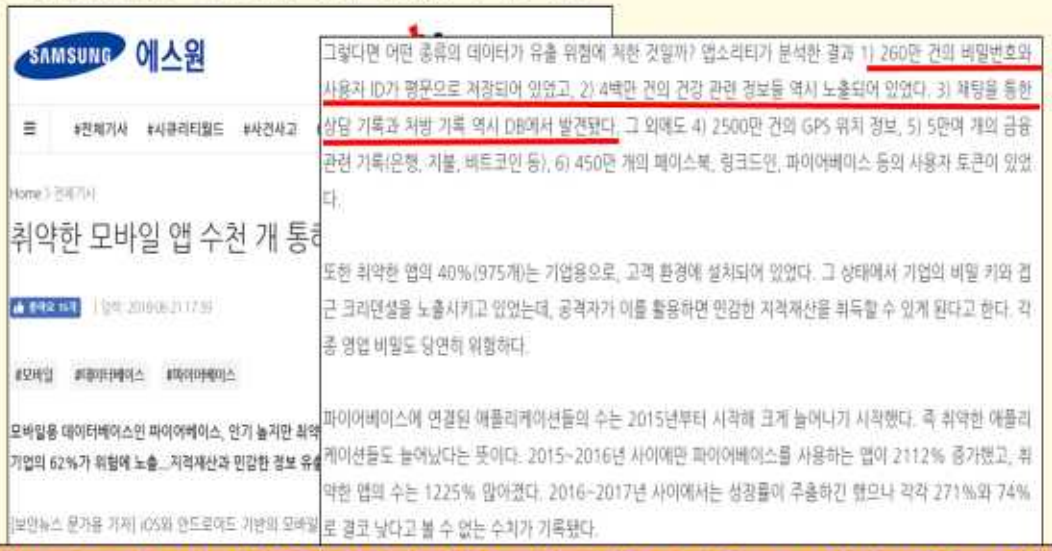
시만텍 조사에 따르면, 인기가 높은 안드로이드 앱의 45%와 iOS 앱의 25%가 위치 확인 권한을 요청하고, 인기 안드로이드 앱의 46%와 iOS 앱의 24%가 사용자 기기의 카메라에 대한 접근 허가를 요청하는 것으로 나타났다. 또한 최고 인기 안드로이드 앱의 44%와 인기가 높은 iOS 앱의 48%에 이메일 주소가 공유되고 있다.

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

4

주제 선정(2/2)

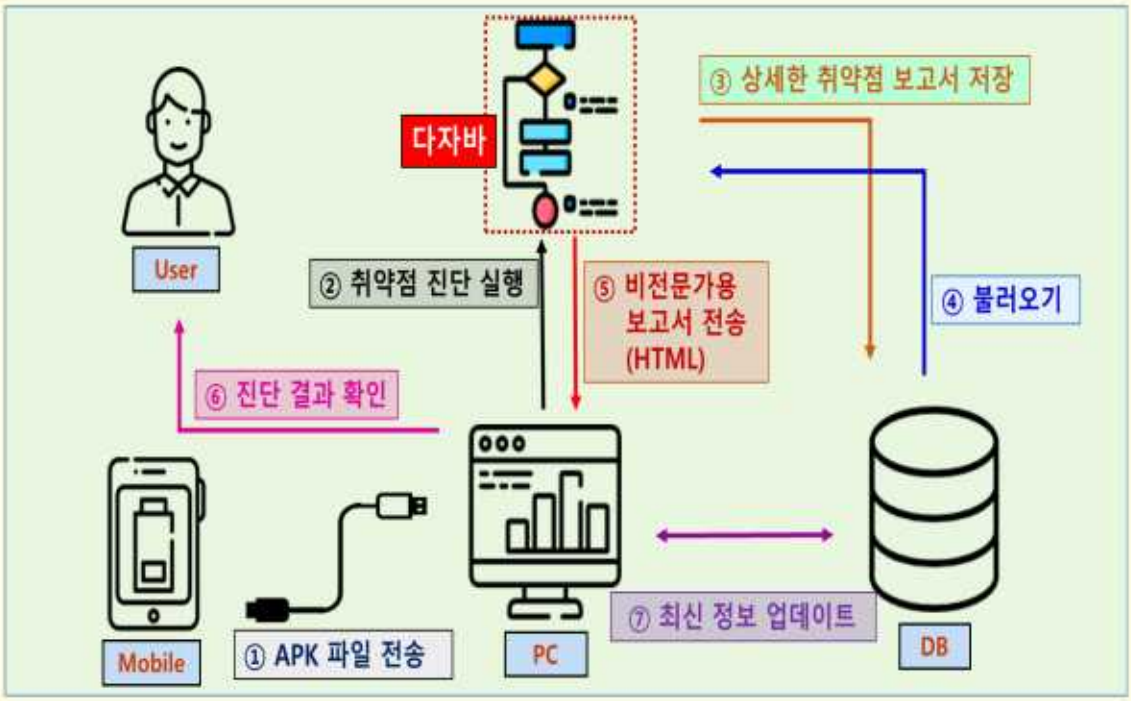
◆ 보안에 취약한 모바일 앱을 통해 개인정보 노출 심각



일반 사용자들이 위험을 쉽게 인지하고 취약점을 진단하는 보안시스템을 구현

5

구상도



6

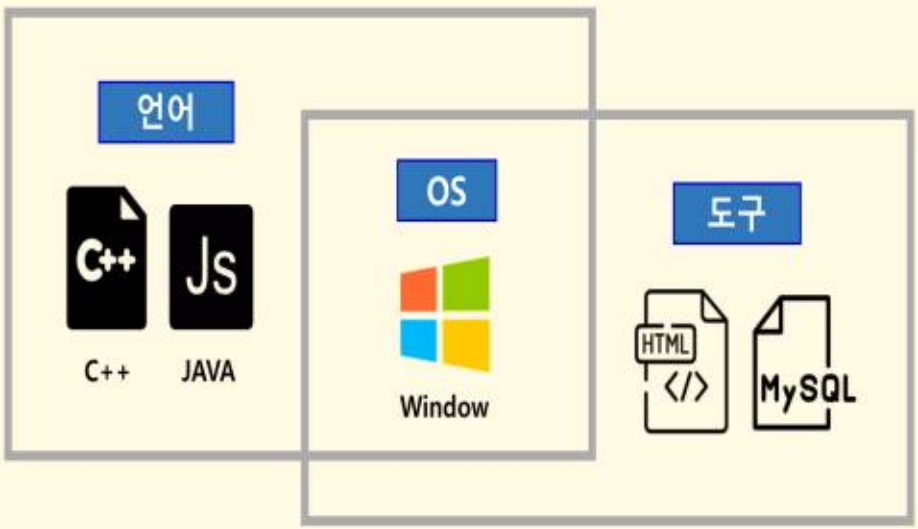
추진 경과

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								

7

개발 환경 및 개발 내용(1/8)

개발 환경



8

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

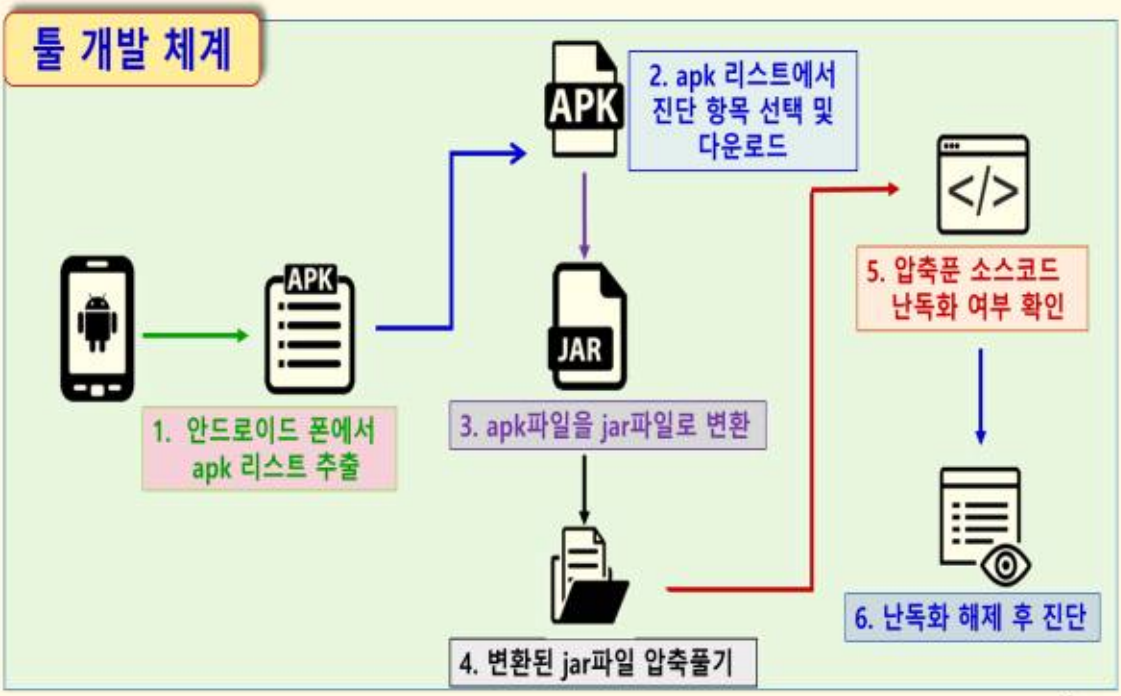
앱 취약점 진단 항목

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

9

개발 환경 및 개발 내용(3/8)

툴 개발 체계



10

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

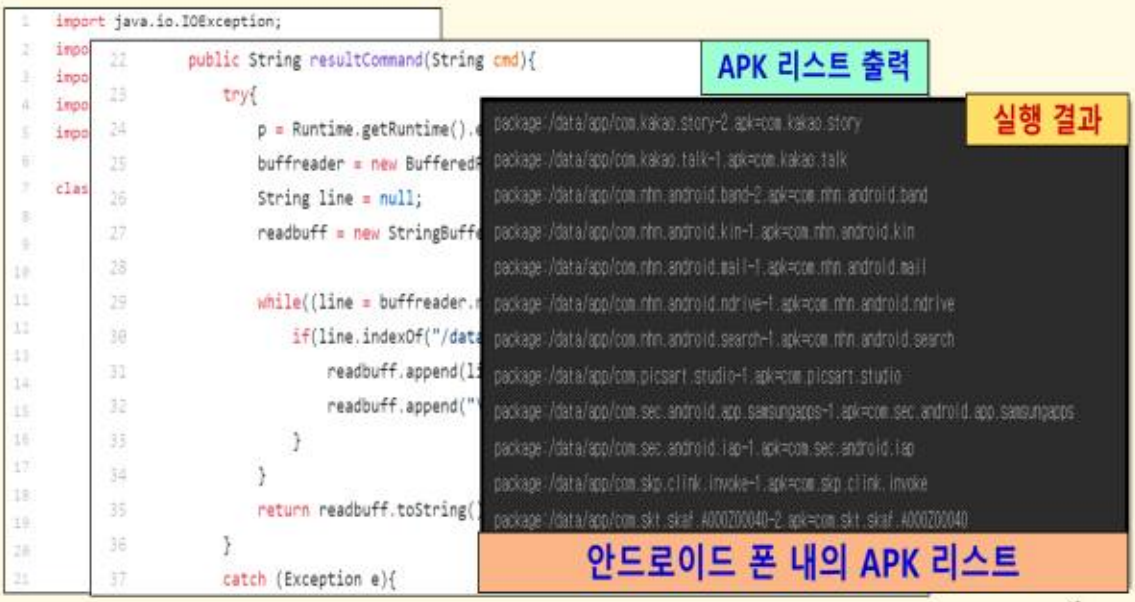
◆ 사용자 편의성을 보장하기 위해 GUI 개발 및 보고서 관리를 위한 DB 구축

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

11

개발 환경 및 개발 내용(5/8)

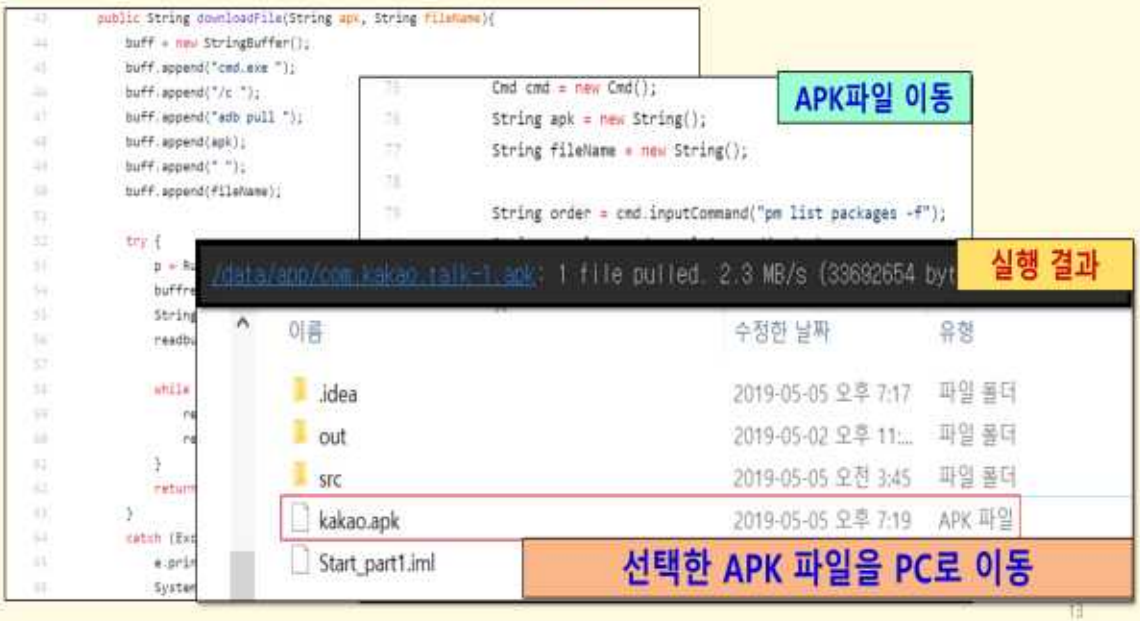
APK 확인(안드로이드)



12

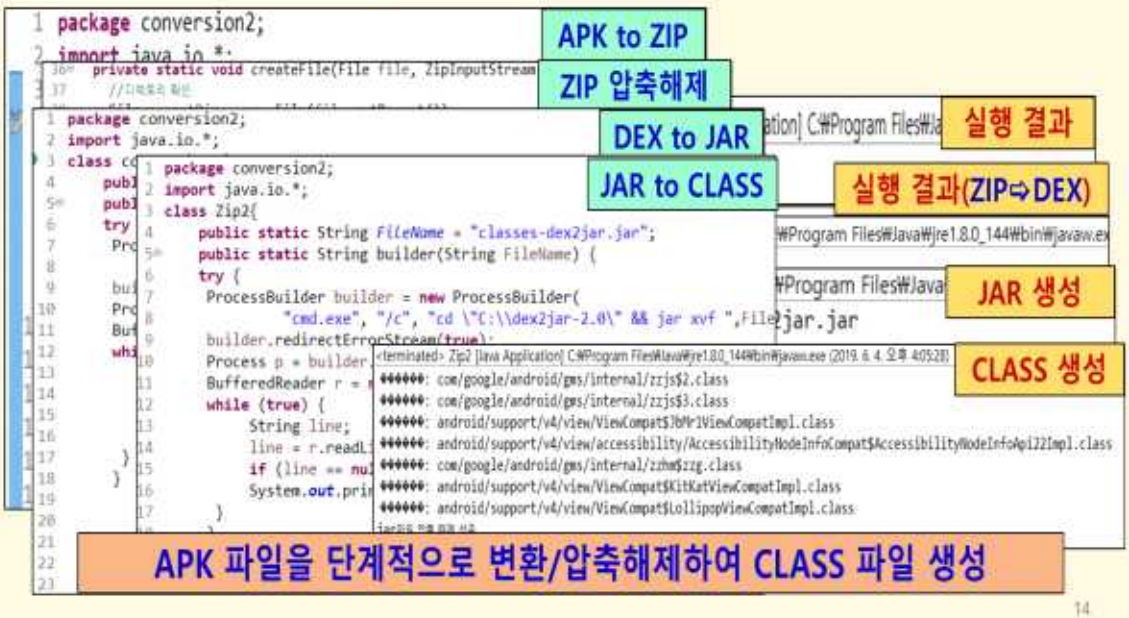
개발 환경 및 개발 내용(6/8)

APK 추출/이동(안드로이드 ⇨ PC)



개발 환경 및 개발 내용(7/8)

파일 변환/압축해제



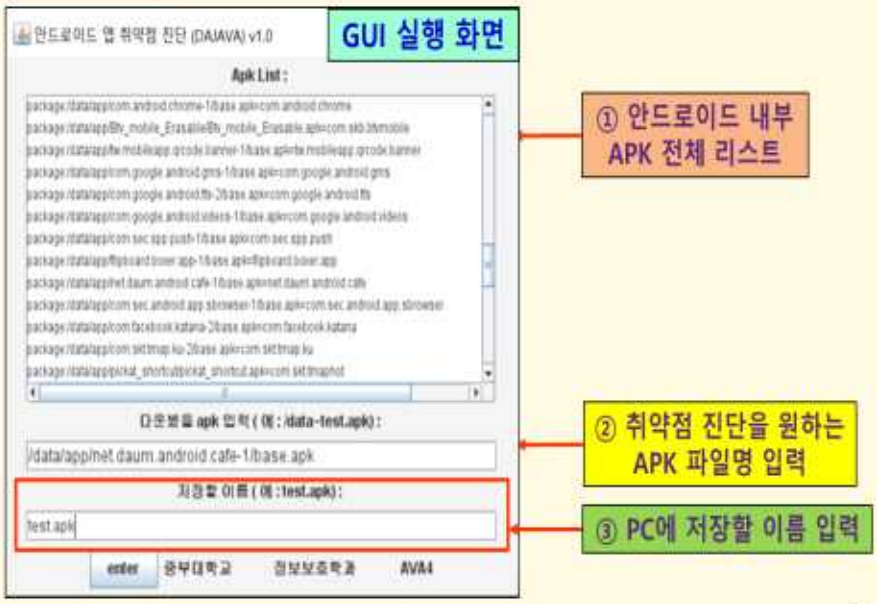
개발 환경 및 개발 내용(8/8)

난독화 체크 ※ 난독화 : 프로그래밍 언어로 작성된 코드를 읽기 어렵게 만드는 작업

15

개발 시스템 운영(1/7)

취약 분석 시스템 구동



16

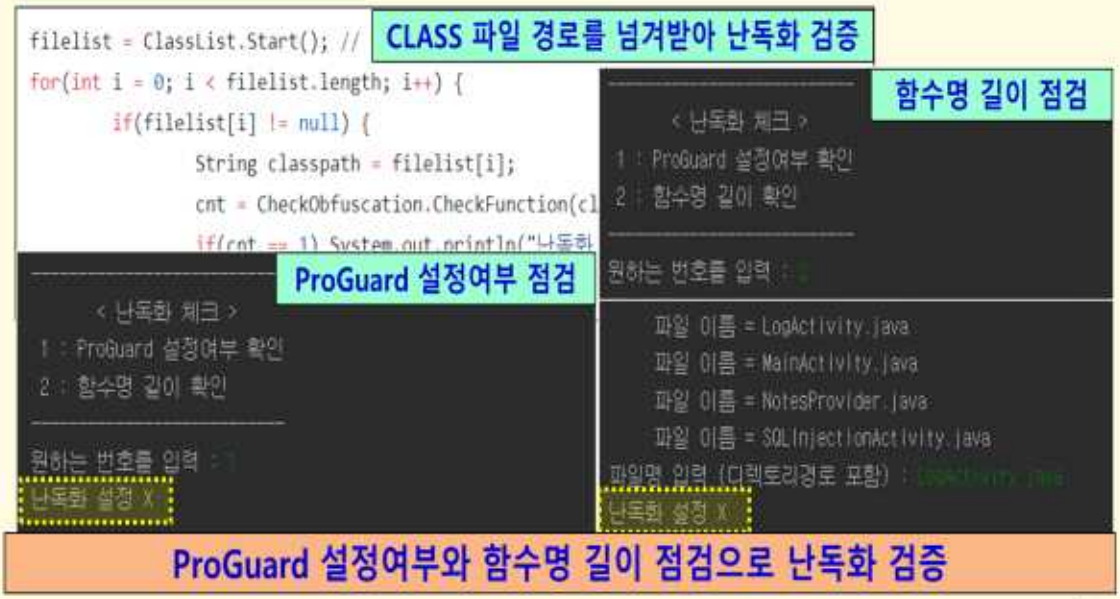
개발 시스템 운영(2/7)

시스템 내부 분석작업 (파일 변환)

17

개발 시스템 운영(3/7)

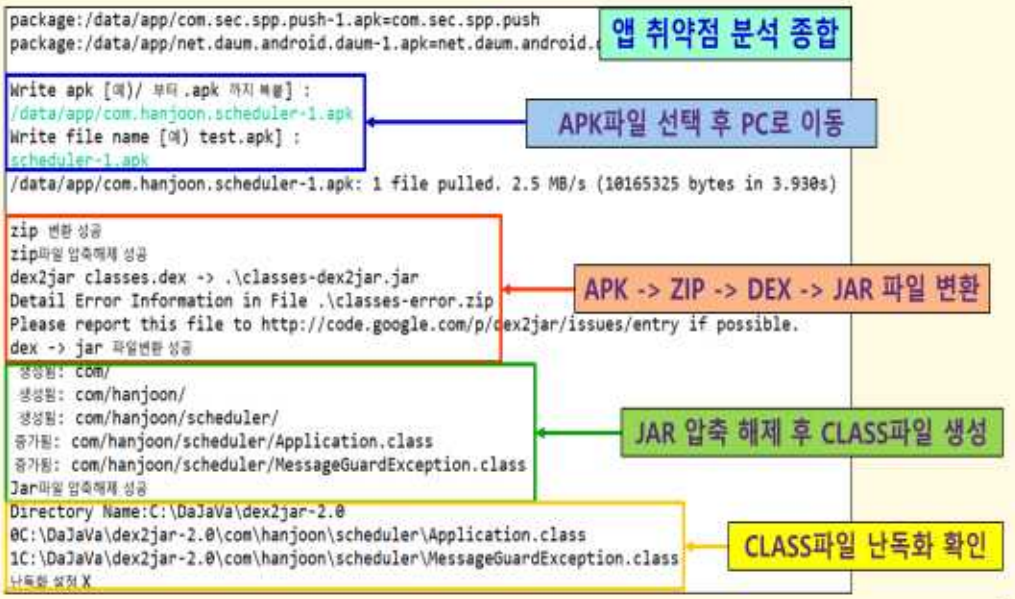
시스템 내부 분석작업 (난독화 검증)



18

개발 시스템 운영(4/7)

시스템 내부 분석작업 (작업 종합)



19

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

20

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
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

21

개발 시스템 운영(7/7)

분석 결과 저장 (HTML)

비전문가용 보고서를 프로그램 실행 시 html창으로 띄워 줌

22

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.

23

Q & A

감사합니다

24