Android Security & Exploitation

Aditya Gupta (@adi1391)

Founder, Attify (http://attify.com)

adi@attify.com

Certifications : http://securitytube-training.com
Pentester Academy : http://PentesterAcademy.com

JDB

- Could debug apps using the JDB (Java Debugger)
- Find out which apps are debuggable
  - **`dz > run app.package.debuggable`**
- If an app is not debuggable, reverse and add **`android:debuggable=true`** to AndroidManifest.xml
- Relies on JDWP
- Official docs : `https://docs.oracle.com/javase/7/docs/technotes/tools/windows/jdb.html`

JDB

jdb - The Java Debugger

jdb helps you find and fix bugs in Java language programs.

SYNOPSIS

```
jdb [ options ] [ class ] [ arguments ]
```

options
Command-line options, as specified below.

class
Name of the class to begin debugging.

arguments
Arguments passed to the `main()` method of `class`.

DESCRIPTION

The Java Debugger, **jdb**, is a simple command-line debugger for Java classes. It is a demonstration of the Java Platform Debugger Architecture that provides inspection and debugging of

Starting a jdb Session

There are many ways to start a jdb session. The most frequently used way is to have **jdb** launch a new Java Virtual Machine (VM) with the main class of the application to be debugged. Th
java in the command line. For example, if your application's main class is MyClass, you use the following command to debug it under JDB:

```
C:\> jdb MyClass
```

When started this way, **jdb** invokes a second Java VM with any specified parameters, loads the specified class, and stops the VM before executing that class's first instruction.

Another way to use **jdb** is by attaching it to a Java VM that is already running. A VM that is to be debugged with **jdb** must be started with the following options. These options load in-proces
connection to be made.

```
-agentlib:jdwp=transport=dt_shmem,server=y,suspend=n
```

For example, the following command will run the MyClass application, and allow **jdb** to connect to it at a later time.

```
C:\> java -agentlib:jdwp=transport=dt_shmem,address=jdbconn,server=y,suspend=n MyClass
```

JDWP

- Java Debugging Wire Protocol
- Introduced in JDK 1.4.2
- Used to debug java apps running in JVM
- Also supported by DVM
- Could also use JSwat (https://github.com/nlfiedler/jswat) which runs on JDWP

Debugging app using JDB

- **dz> run app.activity.start –component [package name] [activity name]**

- **adb shell ps | grep ‘packagename’ or**
  adb jdwp

- **adb forward tcp:[localport] jdwp:[port on device]**

- **jdb –attach localhost:[localport]**

Debugging app using JDB

- **classes**
- **methods com.android.insecurebank.RestClient**
- **stop in com.android.insecurebank.RestClient.dotransfer**
- **go to the app and initiate the transfer**
- **where**
- **locals**
- **set amount=“2000”**
- **dump this**
- **eval dotransfer(“192.168.161.180”,”8080","12345","56788","1337")**