Android Mobile Pentest 101

© tsug0d, September 2018

Lecture 10.2 – Creating Exploit: Making HTTP Request

Goal: Known how to make a http request in android app

Introduction

- This lecture will help you understand how to send a http request to external server via android app
- *Why? To send data of other user to our server*

Let’s dev

- First, creating new project with empty activity (previous lecture)
- We will work mainly in MainActivity.java & AndroidManifest.xml

Let’s dev

- Let talk about “Permission” (https://developer.android.com/guide/topics/permissions/overview)
- A central design point of the Android security architecture is that no app, by default, has permission to perform any operations that would adversely impact other apps, the operating system, or the user.
- To send HTTP Request, we have to defined in AndroidManifest.xml two permissions:

```
android.permission.INTERNET
android.permission.ACCESS_NETWORK_STATE
```

- How to? Using <uses-permission> tag:

```xml
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Let’s dev

- We are moving to coding part!
- Included their HTTP support lib in MainActivity.java:

```
import androidx.appcompat.app.AppCompatActivity;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
import android.os.StrictMode.ThreadPolicy.Builder;
import android.os.StrictMode;
import android.os.Bundle;
import android.util.Log;
```

Let’s dev

- Now inside onCreate of MainActivity.java insert this code:

```
StrictMode.setThreadPolicy(new Builder().permitAll().build());
```

- What is it?

StrictMode is a special class for verifying that your Android application is not doing things like disk I/O, Network access from the UI thread.
We are making http connection via UrlConnection class, which get executed on UI thread, therefore we will get “NetworkOnMainThreadException”

To avoid this, we have to override this thread policy to allow the network access via the line above!

Let’s dev

- Next, we defined the url we wan’t to access, in this case: https://tsug0d.com/present/tsu.txt

```
String url = "https://tsug0d.com/present/tsu.txt";
StringBuilder url_holder = new StringBuilder();
url_holder.append(url);
```

Let’s dev

- Now we will make the connection (remember to put in try – catch statement):

```
URLConnection conn = new URL(url_holder.toString()).openConnection();
```

- Then we set some header field:

```
conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
conn.setRequestProperty("charset", "utf-8");
conn.setUseCaches(false);
```

- Create buffer to get the response:

```
BufferedReader buffer = new BufferedReader(new InputStreamReader(conn.getInputStream()));
```

Let’s dev

- Finally, we read the buffer response line by line:

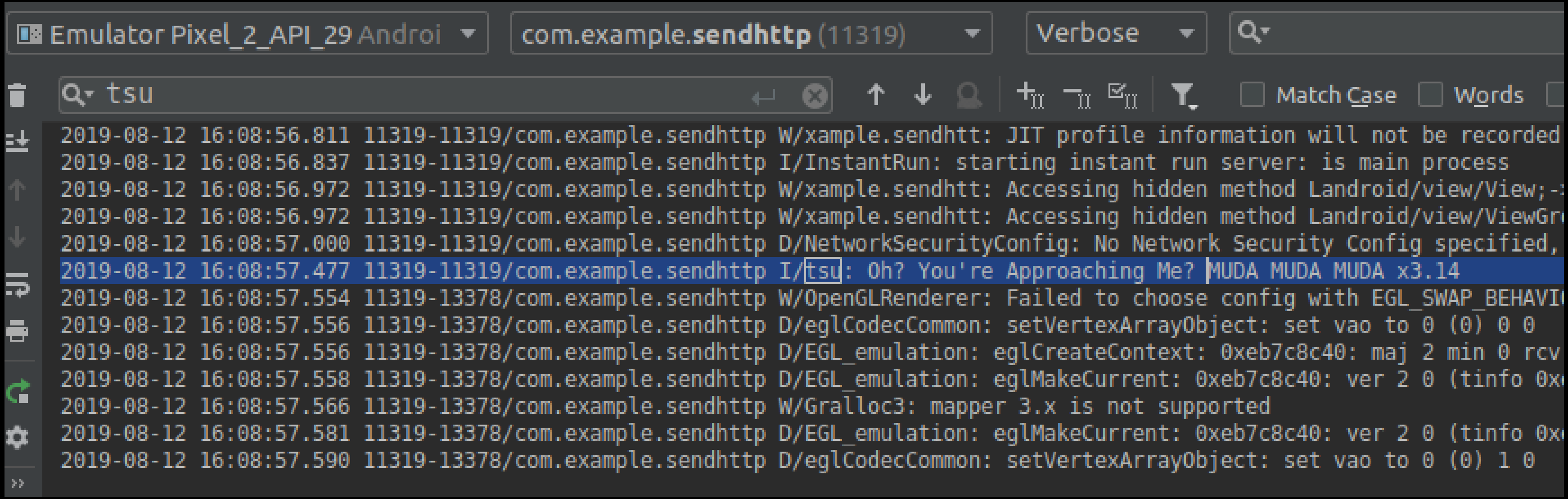
```
String response;
String data_from_stream;
for (response = new String(); true; response += data_from_stream)
{
        String stream = buffer.readLine();
        data_from_stream = stream;
        if (stream == null)
        {
                break;
        }
}
```

- Then log our result ☺

```
Log.i("tsu", response);
```

Let’s dev

- Run the app, here the result ☺



Emulator Pixel_2_API_29 Androi | com.example.sendhttp (11319) | Verbose

tsu | Match Case | Words

```
2019-08-12 16:08:56.811 11319-11319/com.example.sendhttp W/xample.sendhtt: JIT profile information will not be recorded
2019-08-12 16:08:56.837 11319-11319/com.example.sendhttp I/InstantRun: starting instant run server: is main process
2019-08-12 16:08:56.972 11319-11319/com.example.sendhttp W/xample.sendhtt: Accessing hidden method Landroid/view/View;-
2019-08-12 16:08:56.972 11319-11319/com.example.sendhttp W/xample.sendhtt: Accessing hidden method Landroid/view/ViewGr
2019-08-12 16:08:57.000 11319-11319/com.example.sendhttp D/NetworkSecurityConfig: No Network Security Config specified,
2019-08-12 16:08:57.477 11319-11319/com.example.sendhttp I/tsu: Oh? You're Approaching Me? MUDA MUDA MUDA x3.14
2019-08-12 16:08:57.554 11319-13378/com.example.sendhttp W/OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVI
2019-08-12 16:08:57.556 11319-13378/com.example.sendhttp D/eglCodecCommon: setVertexArrayObject: set vao to 0 (0) 0 0
2019-08-12 16:08:57.556 11319-13378/com.example.sendhttp D/EGL_emulation: eglCreateContext: 0xeb7c8c40: maj 2 min 0 rcv
2019-08-12 16:08:57.558 11319-13378/com.example.sendhttp D/EGL_emulation: eglMakeCurrent: 0xeb7c8c40: ver 2 0 (tinfo 0x
2019-08-12 16:08:57.566 11319-13378/com.example.sendhttp W/Gralloc3: mapper 3.x is not supported
2019-08-12 16:08:57.581 11319-13378/com.example.sendhttp D/EGL_emulation: eglMakeCurrent: 0xeb7c8c40: ver 2 0 (tinfo 0x
2019-08-12 16:08:57.590 11319-13378/com.example.sendhttp D/eglCodecCommon: setVertexArrayObject: set vao to 0 (0) 1 0
```

- Grab full code at:

https://github.com/tsug0d/AndroidMobilePentest101/blob/master/lab/AndroidManifest.xml_http
https://github.com/tsug0d/AndroidMobilePentest101/blob/master/lab/MainActivity.java_http