Denial of Service

Background:

A denial of service (DoS) attack is an incident in which a user or organization is deprived of the services of a resource they would normally expect to be able to access. In a distributed denial-of-service (DDoS), large numbers of compromised systems (sometimes called a botnet) attack a single target.

The problem is that Android has this problem in very command way. This is because the number of automatic requests that are received by the server come from different users' apps, like an automatic request to check for new data, or one to update the user location on the server. Imagine that your app has 100,000 users. Consequently, there are many automatic requests going to the server every second, needing many database instances. This is a problem because servers generally have a limit to the number of requests they can fulfill.

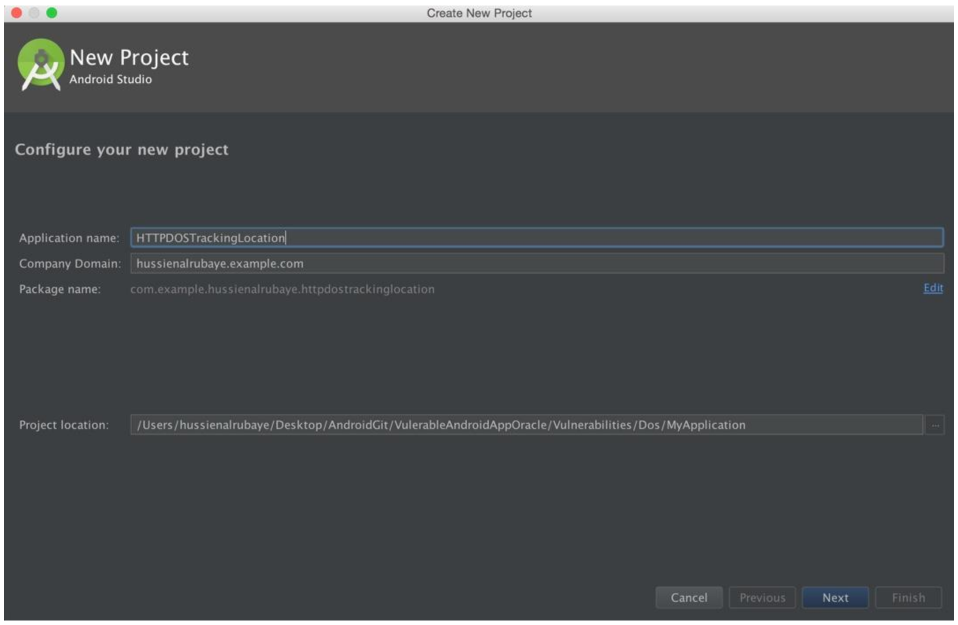
To fix this problem, we must manage the number of requests that are sent by the user app to the server and avoid making any unnecessary requests. For example, instead of sending a request to check the news every ten seconds, we could check every minute to reduce traffic.

We will demonstrate an example here, showing an app that tracks user location by sending the user location over HTTP to the server every time there is a one meter change in location. The problem occurs if many people install the app and use it while travelling by car. The change in user location would then happen very often, and the server would be flooded by requests. This will overwhelm the server, leading many users to be unable to access the services they requested. We will then fix the problem by managing the app requests.

Steps to Build the Tracking Location App:

This app tracks user location by sending user location over HTTP to the server every time there is a one meter change.

1. Create new project in Android Studio named “HTTPDoSTrackingLocation”. Make sure to save the package name somewhere, since you will need it in a later stage of this tutorial. Click next.

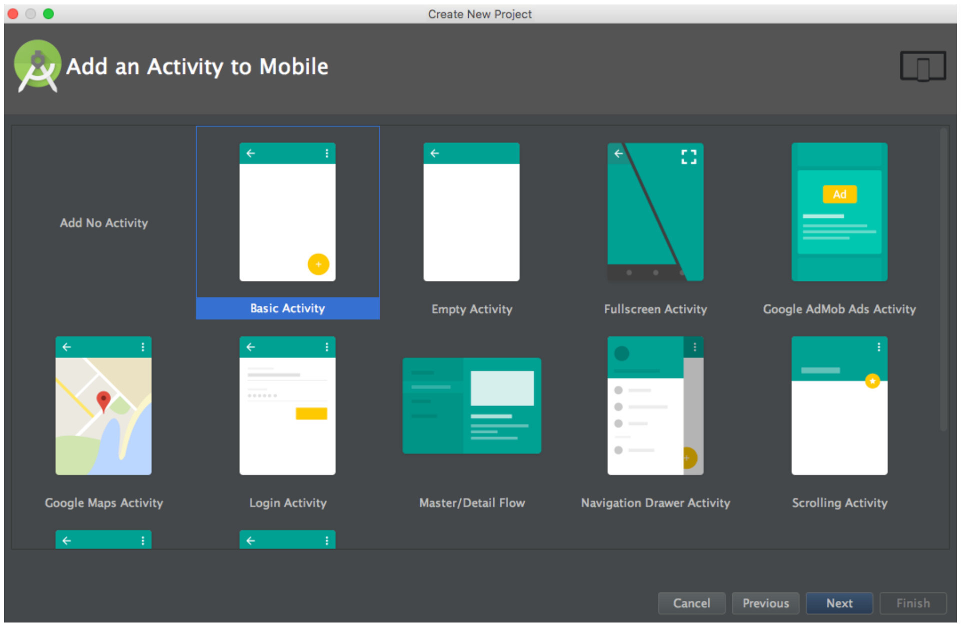


Configure your new project

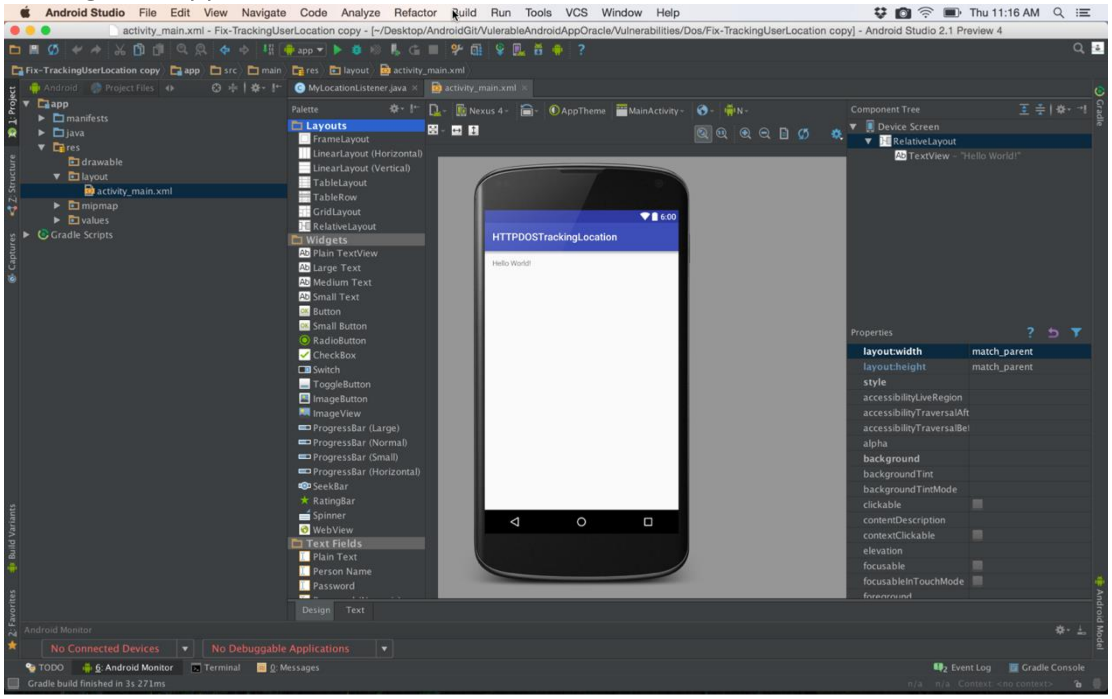
Application name: HTTPDOSTrackingLocation
Company Domain: hussienalrubaye.example.com
Package name: com.example.hussienalrubaye.httpdostrackinglocation

Project location: /Users/hussienalrubaye/Desktop/AndroidGit/VulerableAndroidAppOracle/Vulnerabilities/Dos/MyApplication

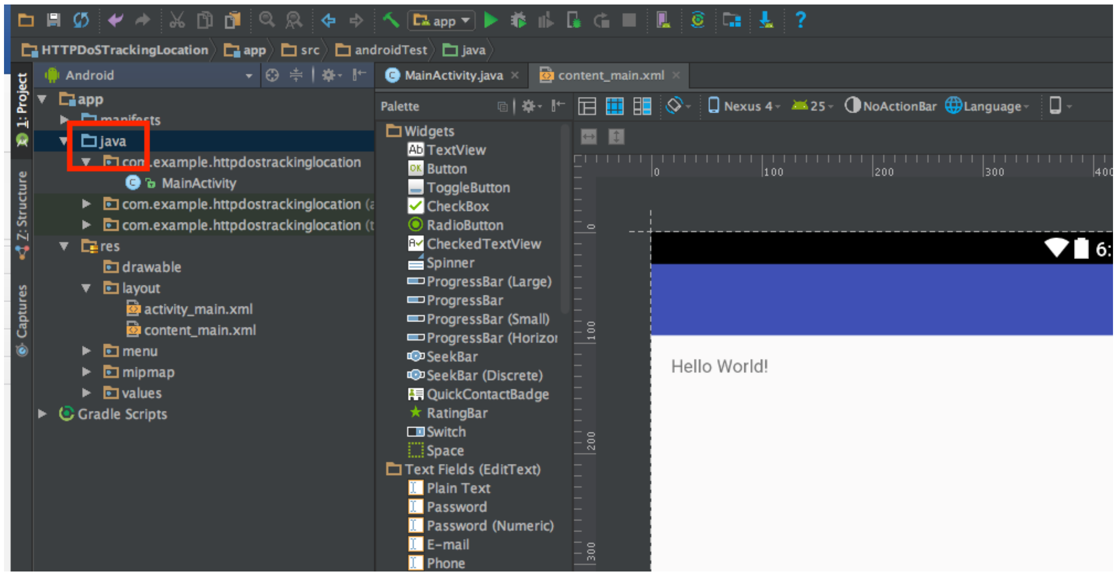
2. Select type Basic Activity.



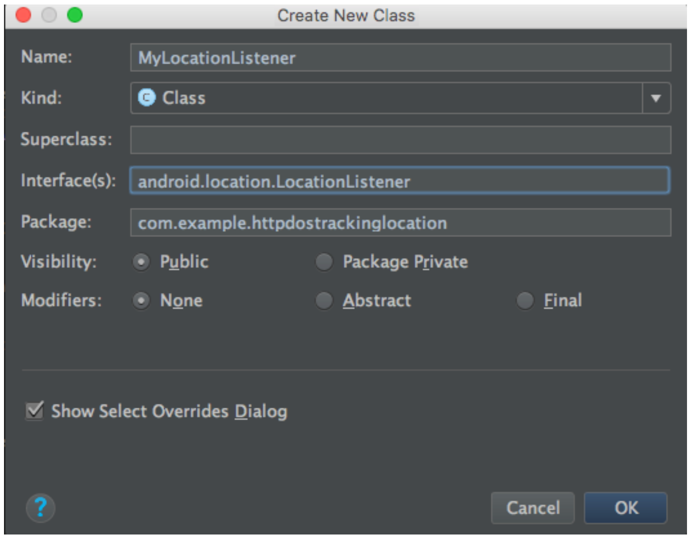
3. Design the app to be like this:



4. Add a class named “MyLocationListener.java” by right-clicking anywhere in the Project View on the right. Select “New Class”. See the icon outlined in red. That is the folder the new class will go into.



Here is how to fill out the prompt that comes up.


Create New Class

Name: MyLocationListener
Kind: Class
Superclass:
Interface(s): android.location.LocationListener
Package: com.example.httpdostrackinglocation
Visibility: [✓ Public] Package Private
Modifiers: [✓ None] Abstract Final
[✓] Show Select Overrides Dialog

The class should look like the code displayed below:

```
package com.example.httpdostrackinglocation;

import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.os.Bundle;
```

```java
import android.widget.Toast;

import java.net.HttpURLConnection;
import java.net.URL;

/**
 * This service listens to changes in user location every meter
 * and sends this data over http to server to save user location
 */
public class MyLocationListener implements LocationListener {
    // previous send date
    Context context;
    public static boolean isActive = false;
    String URL;
    public MyLocationListener(Context context) {
        this.context = context;
    }

    // this method is called for every one meter change in one meter in location
    public void onLocationChanged(Location location) {
        // define the http url to receive the location data
        URL = "http://news.alruabye.net/newfeeds.aspx?Latitude=" + Double.toString(location.getLatitude()) +
"&Longitude=" + Double.toString(location.getLongitude());
        Toast.makeText(context, URL, Toast.LENGTH_LONG).show();

        try {
            // define url for send user location
            URL url = new URL(URL);

            // make http connection to send user location
            HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();

            //set timeout to 5 seconds
            urlConnection.setConnectTimeout(7000);
            // InputStream in = new BufferedInputStream(urlConnection.getInputStream());
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            //urlConnection.disconnect();
        }
    }

    public void onStatusChanged(String s, int i, Bundle b) {

    }

    public void onProviderDisabled(String s) {

    }

    public void onProviderEnabled(String s) {
    }
}
```

5. Update AndroidManifest.xml, which can be found under “app/manifests/”, to be like this:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.httpdostrackinglocation">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

6. Update **MainActivity.java** to look like this:

```
package com.example.httpdostrackinglocation;

import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.LocationManager;
import android.os.Build;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
    // Setting send user location times and distances
    private static final long MINIMUM_DISTANCE_CHANGE_FOR_UPDATES = 1; // in Meters
    private static final long MINIMUM_TIME_BETWEEN_UPDATES = 1000; // in Milliseconds
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // check if the API >= 23 to display runtime request permission
    if (Build.VERSION.SDK_INT >= 23) {
        // check if this permission is not granted yet
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
                PackageManager.PERMISSION_GRANTED) {
            //shouldShowRequestPermissionRationale(). This method returns true

            // if the app has requested this permission previously and the user denied the request.
            if (!shouldShowRequestPermissionRationale(Manifest.permission.ACCESS_FINE_LOCATION)) {
                // the app then requests permission
                requestPermissions(new String[]{Manifest.permission.ACCESS_FINE_LOCATION,
                                Manifest.permission.ACCESS_COARSE_LOCATION},
                        REQUEST_CODE_ASK_PERMISSIONS);
                return;
            }
            return;
        }
    }
    // call start track location
    StartService();
}

// get access to mailbox
final private int REQUEST_CODE_ASK_PERMISSIONS = 123;

// request permission result
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    switch (requestCode) {
        case REQUEST_CODE_ASK_PERMISSIONS:
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // start tracking location
                StartService();

                break;
            }
        default:
            super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    }
}

// This method starts the service for tracking location
void StartService() {
    // run service for one time only
    if (MyLocationListener.isActive == false) {
        MyLocationListener.isActive = true;
```

```
            // start listen service
            LocationManager locationManage = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

            /**
             * Note that the line below requires us to check for permissions OR catch a
             * SecurityException. We opt to catch the SecurityException, simply because we checked
             * if we were granted the permissions upon creation in the onCreate method above (line
             * #25).
             */
            try {
                locationManage.requestLocationUpdates(
                        LocationManager.GPS_PROVIDER,
                        MINIMUM_TIME_BETWEEN_UPDATES,
                        MINIMUM_DISTANCE_CHANGE_FOR_UPDATES,
                        new MyLocationListener(this)
                );
            } catch (SecurityException se) {
                // Let's see what is going on.
                se.printStackTrace();
            }
        }
    }
}
```

How to Fix the Problem:

To fix the problem, we will change how often the user location is sent to the server. We will send updates every minute instead. That way, even if the location changes very quickly, we do not overwhelm the server.

1. Update the class named **MyLocationListener.java**, located under "app/java/com.example.httpdostrackinglocation/". Note that most change is in the method named **onLocationChanged.**

```
package com.example.httpdostrackinglocation;

import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.os.Bundle;
import android.widget.Toast;

import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Date;

/**
 * This service listens to changes in user location every meter
 * and sends this data over http to server to save user location
 */
public class MyLocationListener implements LocationListener {
   // previous send date
   Date lastUpdate = new Date();
   Context context;
   public static boolean isActive = false;
   String URL;
   long diffDate;

   public MyLocationListener(Context context) {
      this.context = context;
   }

   // this method calls every change in one meter in user location
   public void onLocationChanged(Location location) {
      // compute the amount of time elapsed since previous update
      diffDate = (new Date().getTime() - lastUpdate.getTime());

      // check if it has been more than a minute since the last update
      if (diffDate > 60000) {
         lastUpdate = new Date();

         //define the http url that receive location
         URL = "http://news.alruabye.net/newfeeds.aspx?Latitude=" + Double.toString(location.getLatitude()) +
"&Longitude=" + Double.toString(location.getLongitude());
         Toast.makeText(context, URL, Toast.LENGTH_LONG).show();

         try {
            // define url for send user location
            URL url = new URL(URL);

            // make http connection to send user location
            HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
            //set timeout to 5 seconds
            urlConnection.setConnectTimeout(7000);
            // InputStream in = new BufferedInputStream(urlConnection.getInputStream());
         } catch (Exception e) {
            e.printStackTrace();
         } finally {
            //urlConnection.disconnect();
         }
      }


   }

   public void onStatusChanged(String s, int i, Bundle b) {

   }
```

```
    public void onProviderDisabled(String s) {

    }

    public void onProviderEnabled(String s) {

    }
}
```