Android and JavaScript

**Background**:

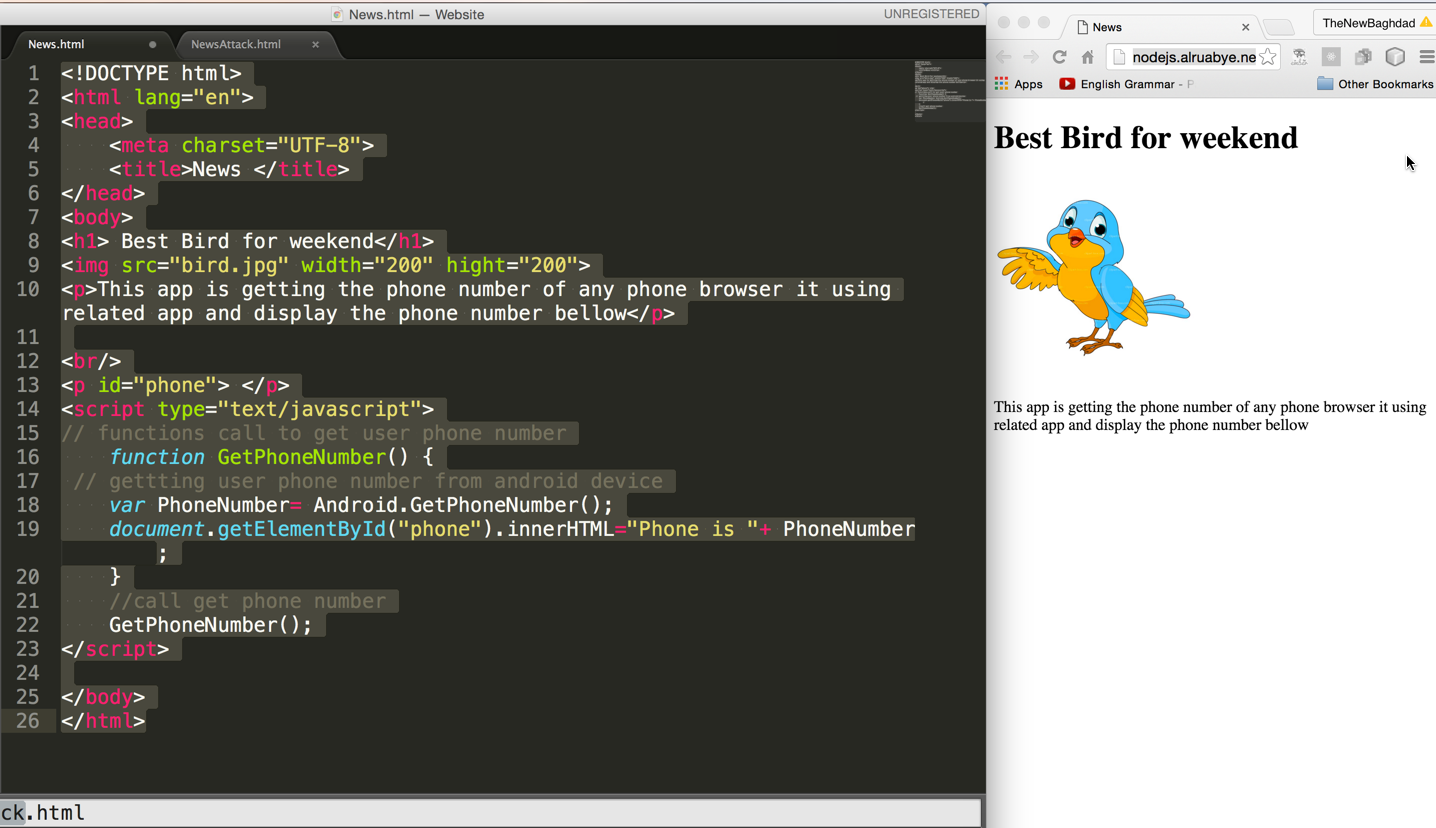
Android has a tool named **WebView** allowing users to visit websites and view other content on the web. This web content normally consists of some HTML, CSS and JavaScript that are rendered in **WebView**. Android allows developers to enable or disable running JavaScript in **WebView** for security purposes. As JavaScript is client side, Android Allows JavaScript to read and write data to and from the device. For example, we could have JavaScript display an alert or open a new activity on the Android device. This means that anyone could view the source code of a web page that has Android JavaScript, get access to the script and use this script (in another website) to access data on the device.

Today we will investigate how sending and receiving sensitive data using JavaScript is not secure.

We will build an app that sends sensitive data like the user’s phone number to the server, and then demonstrate how a hacker’s app can read and get access to this data.

**Creating HTML file**

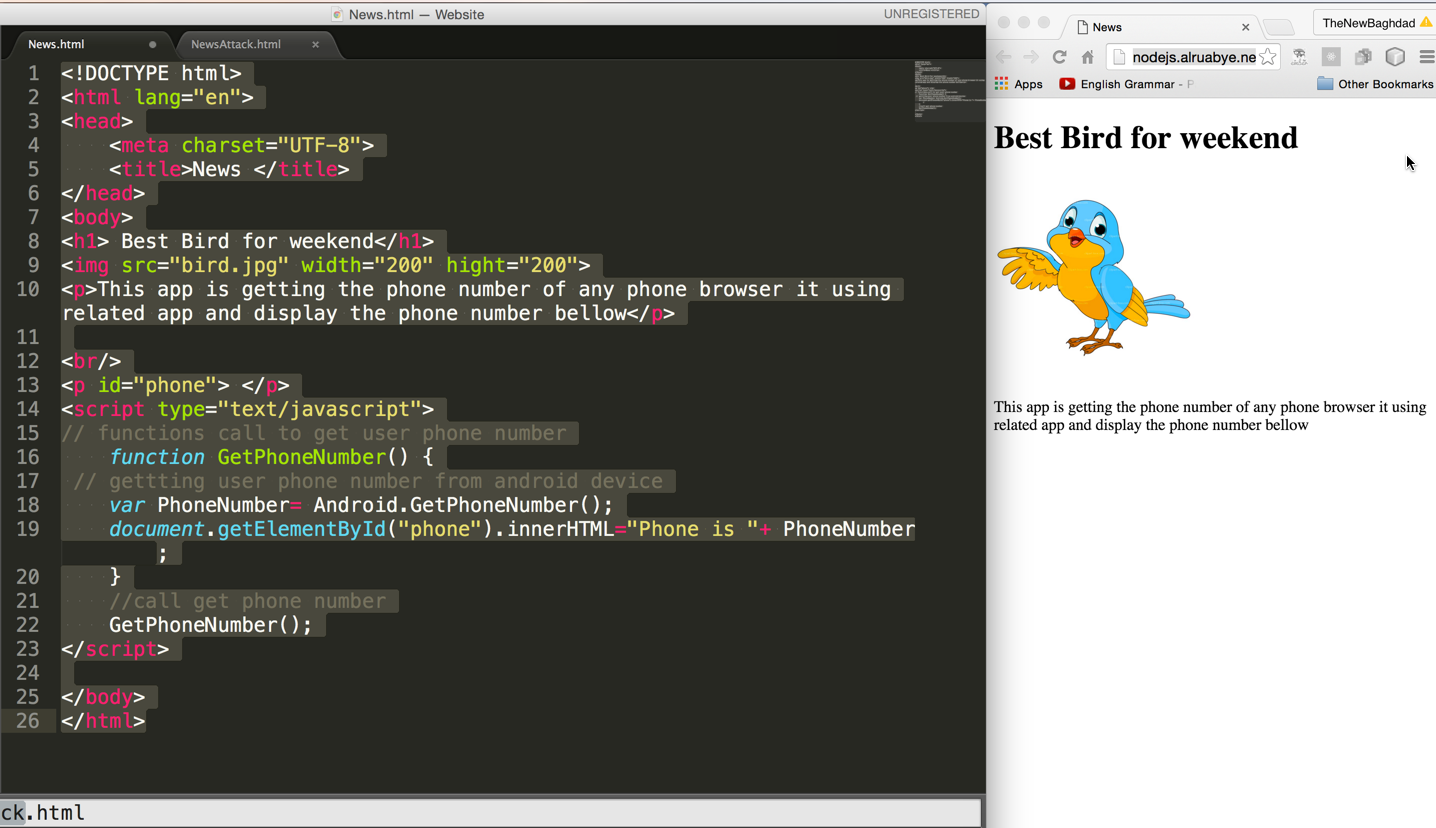
Open new file names News.html



The website should look like this. Then you can use any local webserver to serve the file or

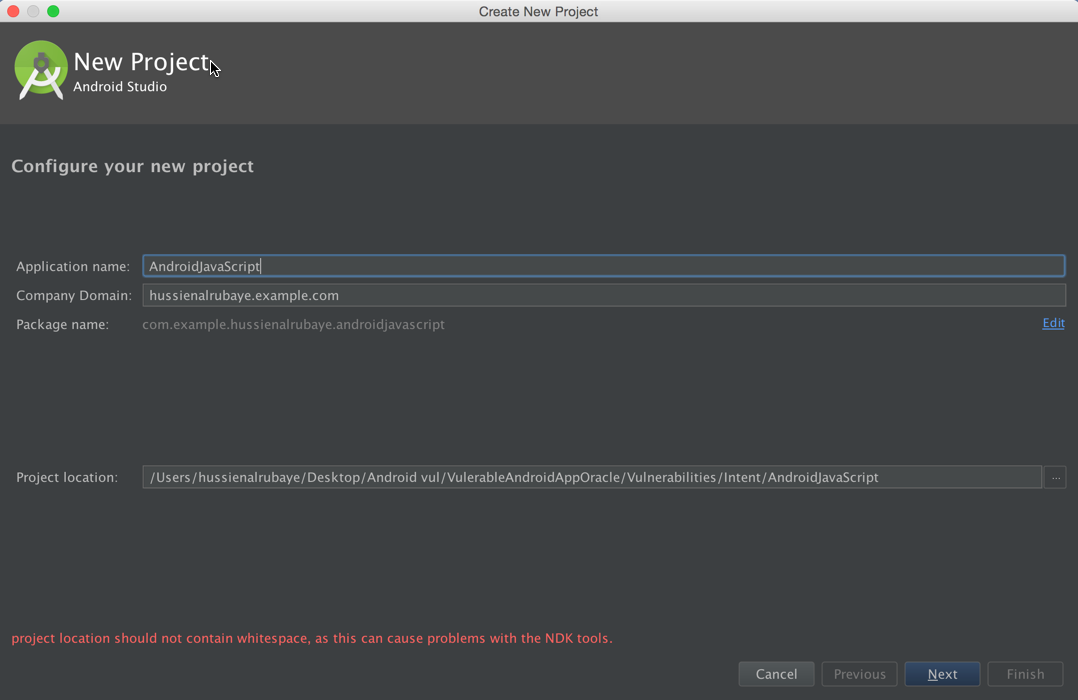
If you do not want to test it with a local server you can browse this url <https://goo.gl/TIGDOb> .

The output should look like the image below



**Steps to build the News View App**

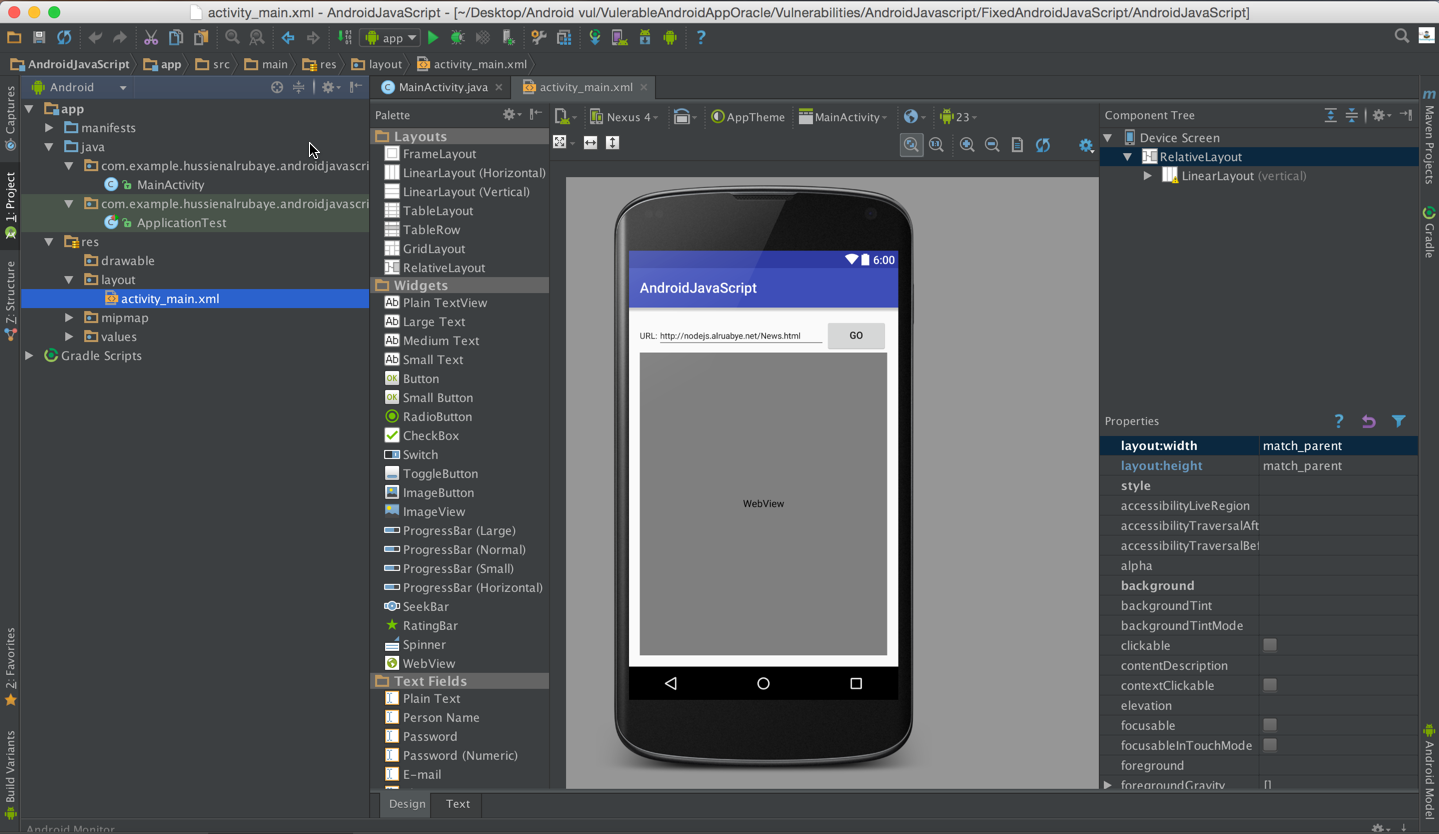
1. Open new project with name “AndroidJavaScript”, save the package name will will need next



1. Paste the following code to activity\_main.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context=".MainActivity"**>   <**LinearLayout  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"**>   <**TextView  android:id="@+id/textView"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="0"  android:text="URL:"  android:textAppearance="?android:attr/textAppearanceLarge"  android:textSize="12dp"** />   <**EditText  android:id="@+id/etURL"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="https://goo.gl/TIGDOb"  android:textSize="12dp"** />   <**Button  android:id="@+id/buGo"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="0"  android:text="Go"** />   </**LinearLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="horizontal"**>   <**WebView  android:id="@+id/wvURL"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_alignParentLeft="true"  android:layout\_alignParentStart="true"  android:layout\_alignParentTop="true"  android:layout\_weight="0"** />  </**LinearLayout**>   </**LinearLayout**> </**RelativeLayout**> |

The result should look like this



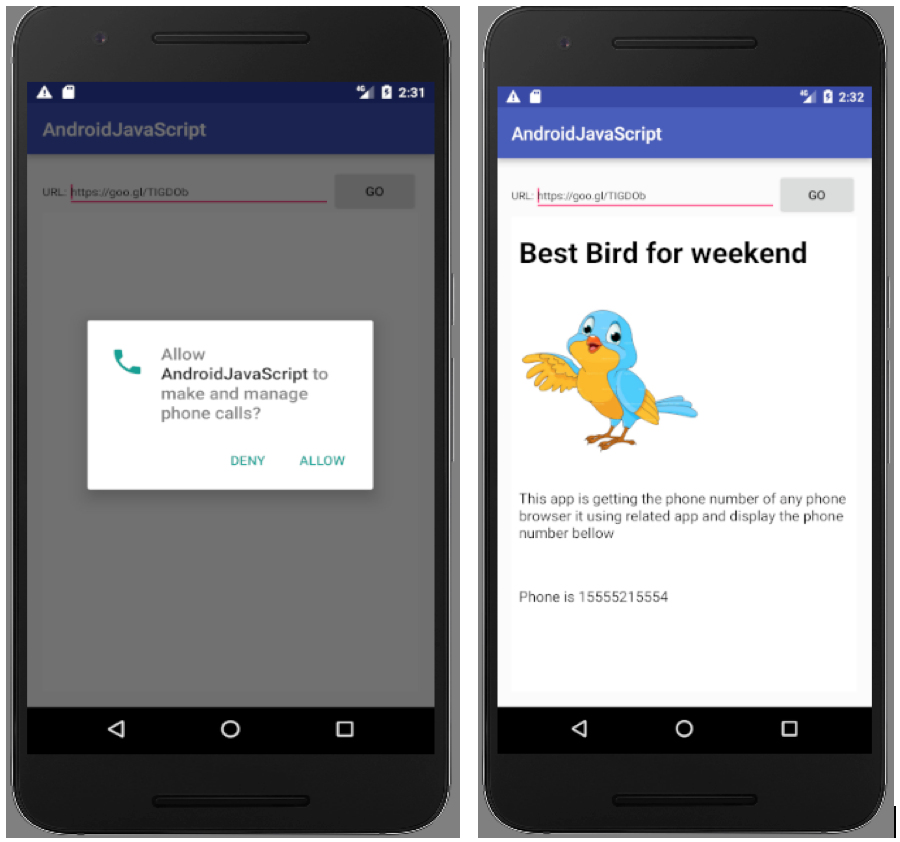
1. Add permission in AndroidManinfest.xml files to access to network and user phone number

|  |
| --- |
| <**uses-permission android:name="android.permission.INTERNET"** /> <**uses-permission android:name="android.permission.READ\_PHONE\_STATE"**/> |

1. The code will be like this code

|  |
| --- |
| **public class** MainActivity **extends** AppCompatActivity {   EditText **etURL**; *//navigation url* WebView **browser**; *// web browser* @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **etURL** = (EditText) findViewById(R.id.***etURL***);  **browser** = (WebView) findViewById(R.id.***wvURL***);   *//Enable Javascript* **browser**.getSettings().setJavaScriptEnabled(**true**);   *//Inject WebAppInterface methods into Web page by having Interface name 'Android'* **browser**.addJavascriptInterface(**new** WebAppInterface(), **"Android"**);   **browser**.setWebViewClient(**new** WebViewClient() {   @Override  **public boolean** shouldOverrideUrlLoading(WebView view, String url) {  view.loadUrl(url);  **return true**;  }   });   *// button that click to go to url* Button buClick = (Button) findViewById(R.id.***buGo***);   *// event to navigate to website* buClick.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *//check if the API>=23 to display runtime request permission* **if** ((**int**) Build.VERSION.***SDK\_INT*** >= 23) {   *// check if this permission is not grated yet* **if** (ActivityCompat.*checkSelfPermission*(getApplicationContext(), Manifest.permission.***READ\_PHONE\_STATE***) !=  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.***READ\_PHONE\_STATE***)) {   *// display request permission* requestPermissions(**new** String[]{Manifest.permission.***READ\_PHONE\_STATE***},  **REQUEST\_CODE\_ASK\_PERMISSIONS**);  **return**;   }   **return**;  }  }   *//load the url that written in edittext to the webview* LoadURL();  }  });  }   *//Class to be injected in Web page* **public class** WebAppInterface {   *//This method return user phone number to the javascript calls from website* @JavascriptInterface *// must be added for API 17 or higher* **public** String GetPhoneNumber() {  **return** GetUserPhoneNumber();*// "585-444-3234";* }   }   */\* this method is getting  user phone number from his device  \*/* String GetUserPhoneNumber() {  TelephonyManager tMgr = (TelephonyManager) getSystemService(Context.***TELEPHONY\_SERVICE***);  String mPhoneNumber = tMgr.getLine1Number();  **return** mPhoneNumber;  }   **void** LoadURL() {   *//load the url that written in edittext to the webview* **browser**.loadUrl(**etURL**.getText().toString());  }   *//get access to mailbox* **final private int REQUEST\_CODE\_ASK\_PERMISSIONS** = 123;   *//request permsion result* @Override  **public void** onRequestPermissionsResult(**int** requestCode, String[] permissions, **int**[] grantResults) {  **switch** (requestCode) {   **case REQUEST\_CODE\_ASK\_PERMISSIONS**:   **if** (grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {   *// load the url data* LoadURL();   } **else** {  *// Permission Denied* }  **break**;  **default**:  **super**.onRequestPermissionsResult(requestCode, permissions, grantResults);  }  } } |

View the page content

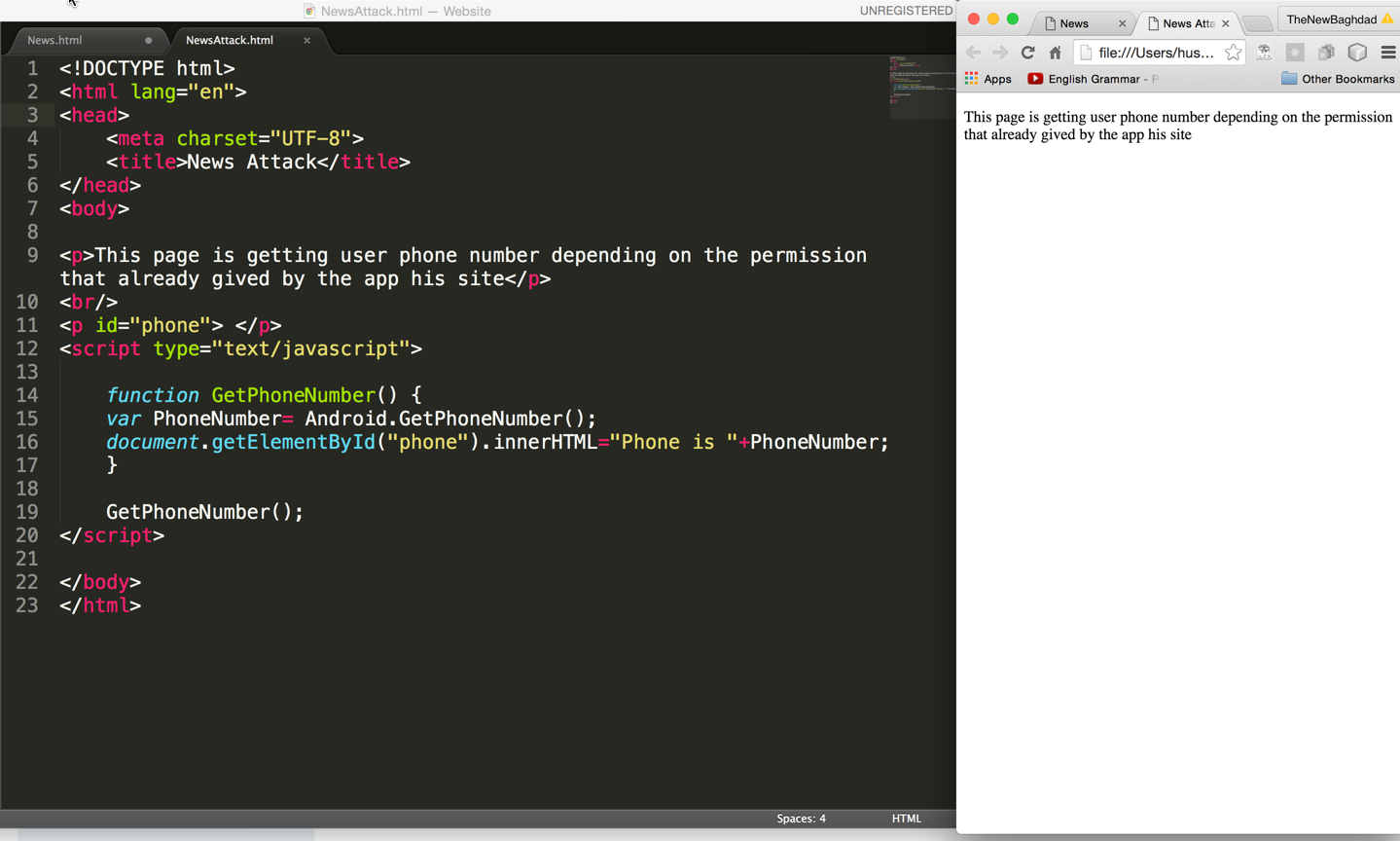


**Steps to build the hacker app:** Another website can embed the same permissions included in your website’s script to gain access to user’s data on the device.

1. A hacker could inspect your website’s code and see that you are using Android function in your script

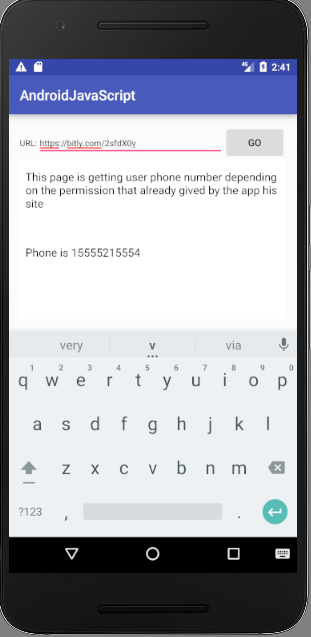


1. Hacker will insert same JavaScript in his website. When your users view this website, he will get user’s personal information through your app’s permissions

****

**Example of the user view hacker website, and the hacker get his phone number**

If you do not want to run local server you can use this url https://bitly.com/2sfdX0v as the hacker url



**Fix This Problem**

To fix this problem, we must send sensitive data only to the websites that we wish to authorize to access this data like our websites, or we could enable JavaScript to be run only in our website. The code below allows for sending sensitive data only to the websites that we authorize. Change the hotingURL if you are using a local server.

|  |
| --- |
| **public class** MainActivity **extends** AppCompatActivity {   EditText **etURL**; *//navigation url* WebView **browser**; *// web browser   // host name* String **HostingURL** = **"https://goo.gl/TIGDOb"**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **etURL** = (EditText) findViewById(R.id.***etURL***);  **browser** = (WebView) findViewById(R.id.***wvURL***);   *//Enable Javascript* **browser**.getSettings().setJavaScriptEnabled(**true**);   *//Inject WebAppInterface methods into Web page by having Interface name 'Android'* **browser**.addJavascriptInterface(**new** WebAppInterface(), **"Android"**);   **browser**.setWebViewClient(**new** WebViewClient() {   @Override  **public boolean** shouldOverrideUrlLoading(WebView view, String url) {  view.loadUrl(url);  **return true**;  }   });   *// button that click to go to url* Button buClick = (Button) findViewById(R.id.***buGo***);   *// event to navigate to website* buClick.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *//check if the API>=23 to display runtime request permission* **if** ((**int**) Build.VERSION.***SDK\_INT*** >= 23) {   *// check if this permission is not grated yet* **if** (ActivityCompat.*checkSelfPermission*(getApplicationContext(), Manifest.permission.***READ\_PHONE\_STATE***) !=  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.***READ\_PHONE\_STATE***)) {   *// display request permission* requestPermissions(**new** String[]{Manifest.permission.***READ\_PHONE\_STATE***},  **REQUEST\_CODE\_ASK\_PERMISSIONS**);  **return**;   }   **return**;  }  }   *//load the url that written in edittext to the webview* LoadURL();  }  });  }   *//Class to be injected in Web page* **public class** WebAppInterface {   *//This method return user phone number to the javascript calls from website* @JavascriptInterface *// must be added for API 17 or higher* **public** String GetPhoneNumber() {   *// only send the phone to authorize website* **if**(**etURL**.getText().toString().indexOf(**HostingURL**)==0)  **return** GetUserPhoneNumber();   **else  return null**;   }   }   */\* this method is getting  user phone number from his device  \*/* String GetUserPhoneNumber() {  TelephonyManager tMgr = (TelephonyManager) getSystemService(Context.***TELEPHONY\_SERVICE***);  String mPhoneNumber = tMgr.getLine1Number();  **return** mPhoneNumber;  }   **void** LoadURL() {   *//load the url that written in edittext to the webview* **browser**.loadUrl(**etURL**.getText().toString());  }   *//get access to mailbox* **final private int REQUEST\_CODE\_ASK\_PERMISSIONS** = 123;   *//request permsion result* @Override  **public void** onRequestPermissionsResult(**int** requestCode, String[] permissions, **int**[] grantResults) {  **switch** (requestCode) {   **case REQUEST\_CODE\_ASK\_PERMISSIONS**:   **if** (grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {   *// load the url data* LoadURL();   } **else** {  *// Permission Denied* }  **break**;  **default**:  **super**.onRequestPermissionsResult(requestCode, permissions, grantResults);  }  } } |

As we see our website could access to phone number while hacker website cannot.

