**Android And JavaScript**

Description:

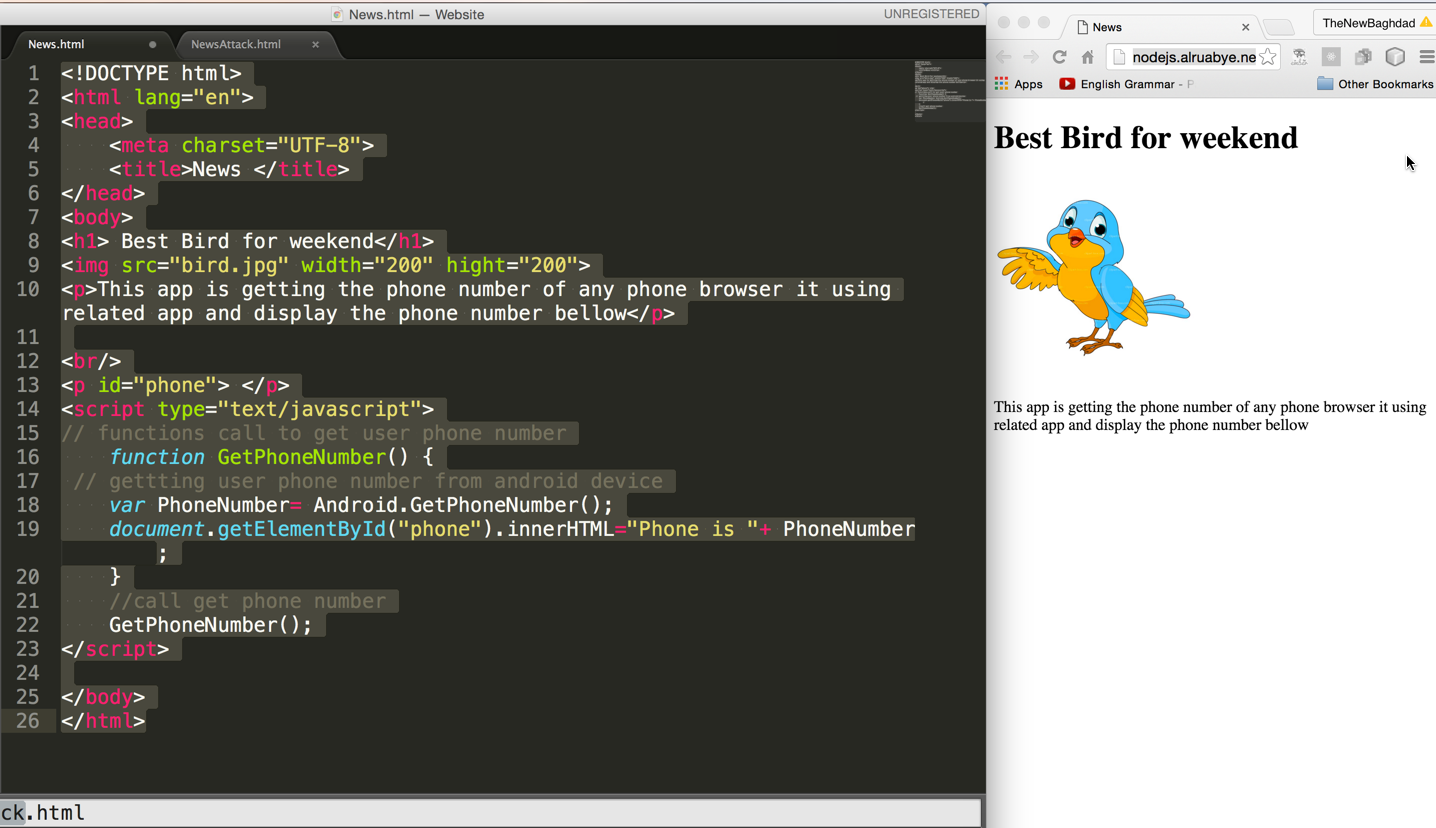
In Android there is tool named **WebView** that used for view web contents. Many developers use it to view their websites or any web content. This web content could have (CSS, HTML, JavaScript) that render in **WebView**, Android allows developers to enable or disable running JavaScript in **Webview** for security purpose, Also because JavaScript is client side, Android Allows JavaScript to read and write data to and from the phone. For example, we could send command from JavaScript to display alert dialog or open new activity in Android phone. That is mean that any one could view source code of web page that have Android JavaScript and see the JavaScript, then he could access and read same your app permissions if people user your app to view his website.

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

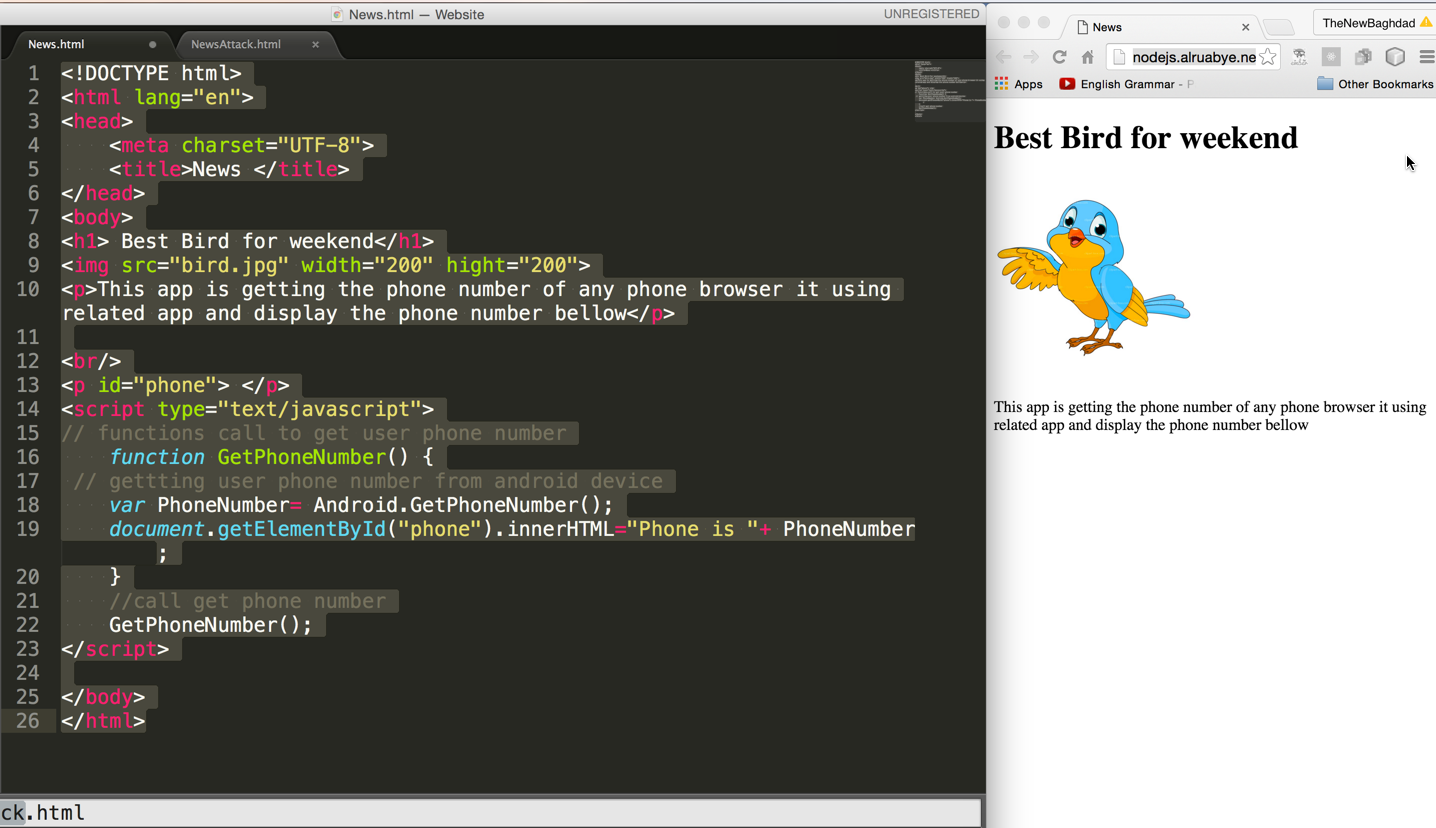
We will write app that send sensitive data like user phone number to the server, then we will demonstrate how hacker’s app read and access to this data.

**Steps to build the Webhost server**

Open new file names News.html

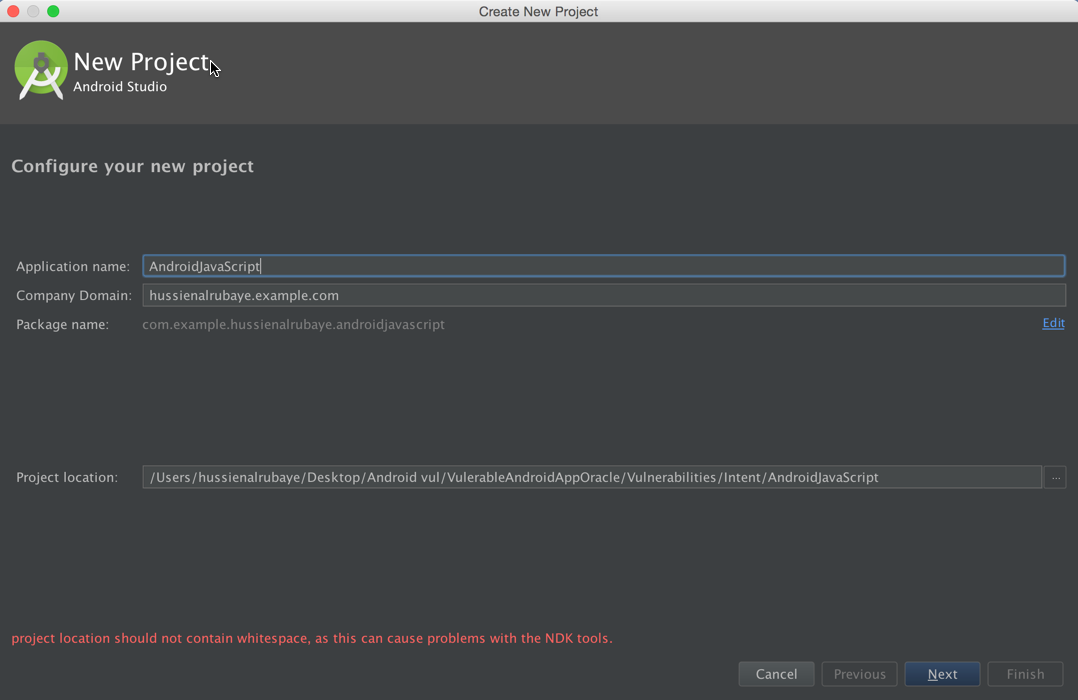


website view like this

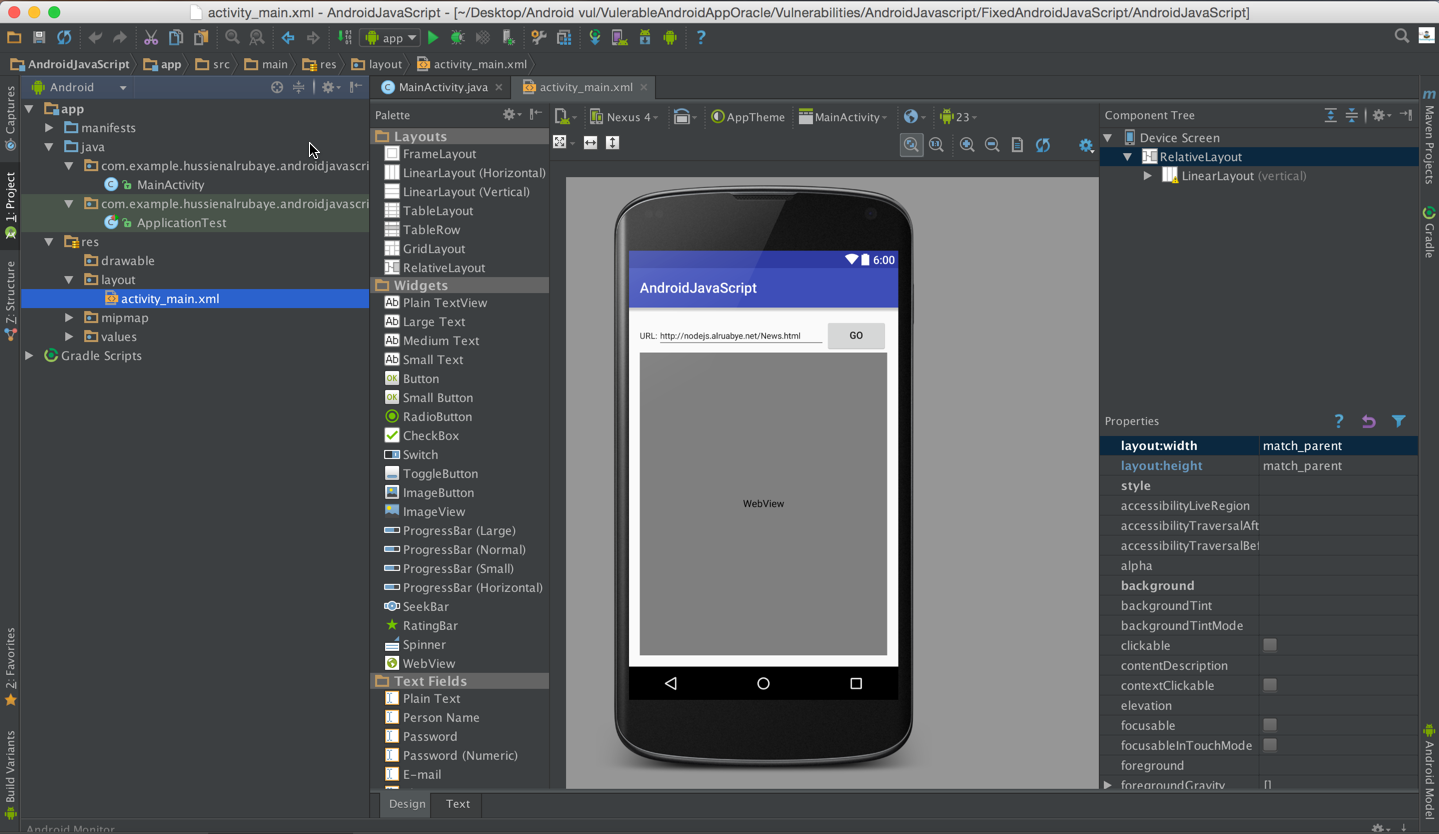


**Steps to build the News View App**

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



1. add some objects ( TextView, EditText, Button,WebView) and make the app like this, see the name of every tool in the right.



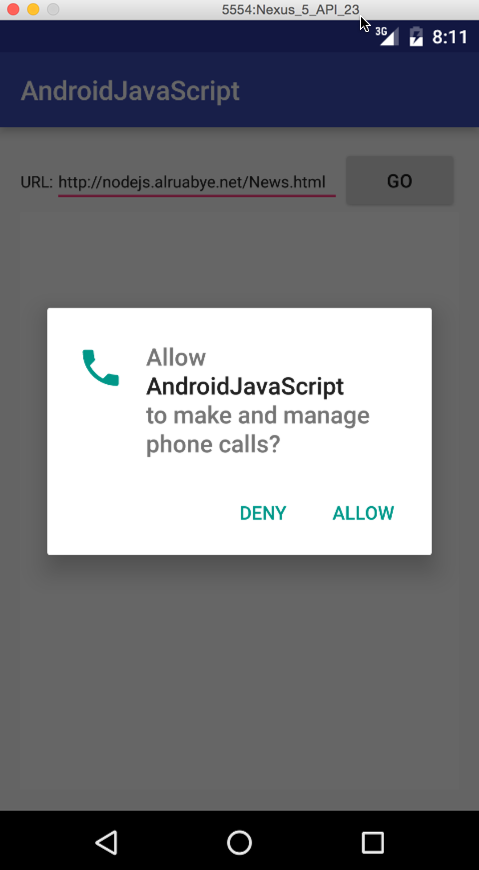
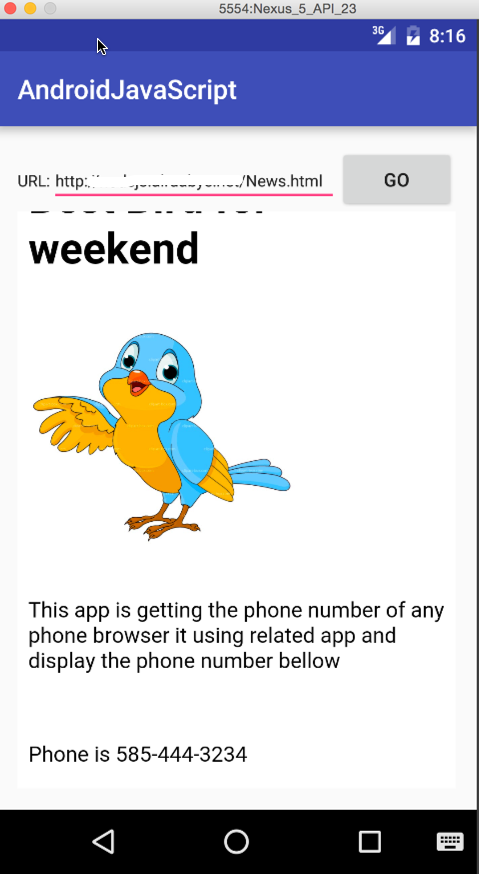
1. Add permission in MAinfest.xml files to access to network and user phone number

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

1. The code will be like this code

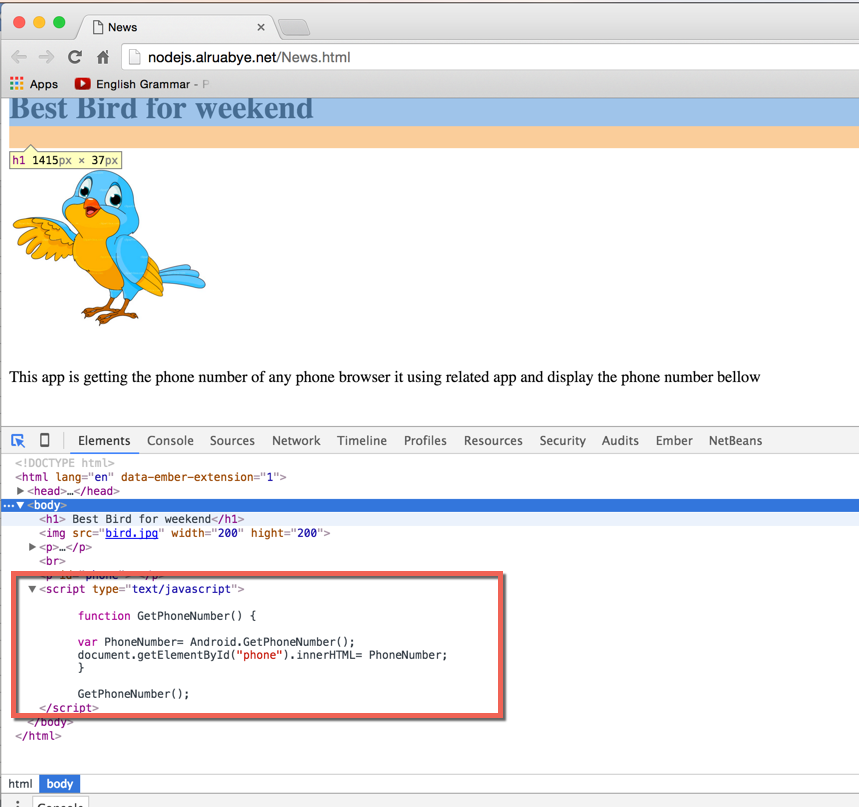
|  |
| --- |
| Java |
| public class MainActivity extends AppCompatActivity { EditText etURL; //navigation url  WebView browser; // web browser  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);  etURL=(EditText)findViewById(R.id.*etURL*);  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");   // 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

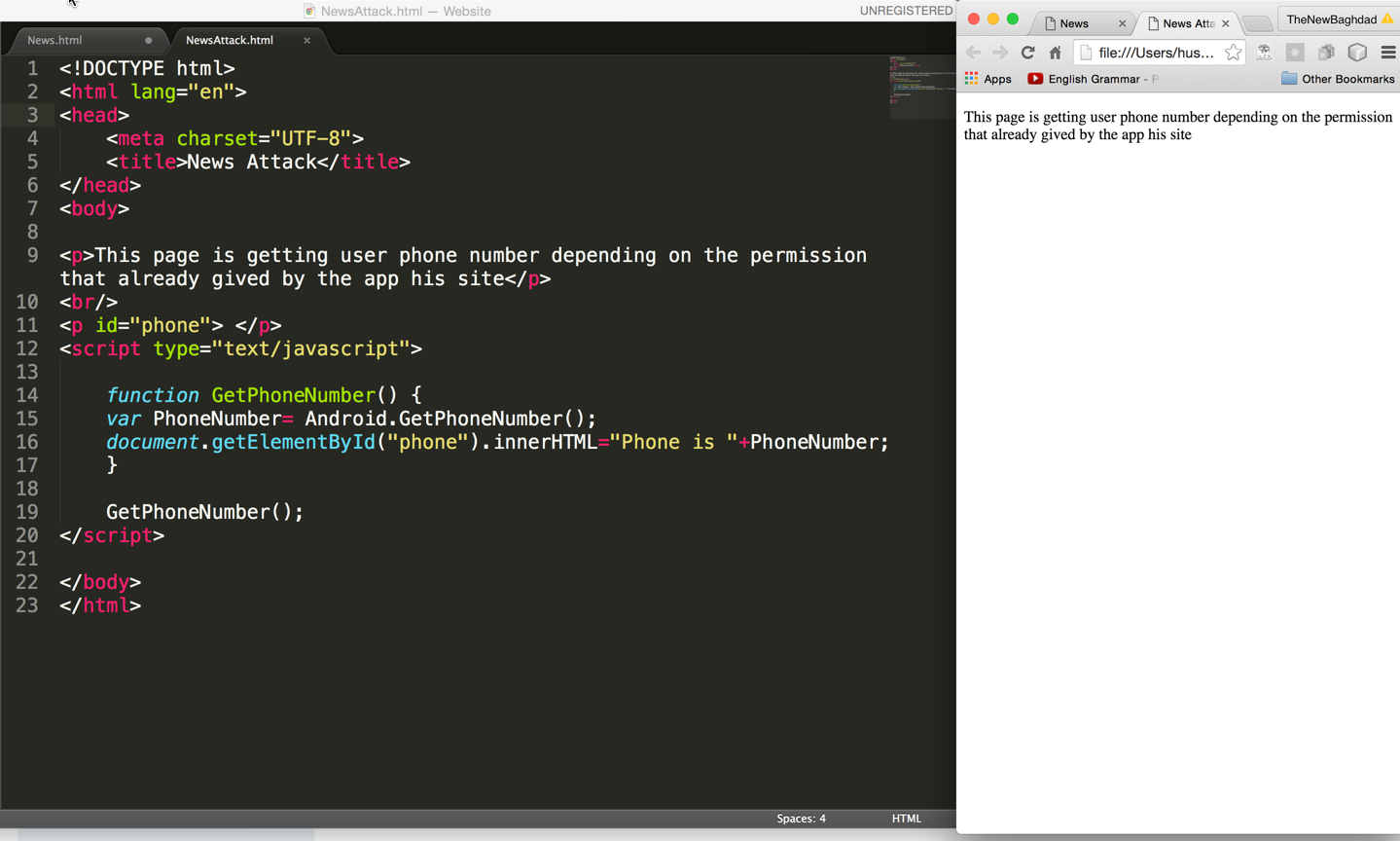
 

**Steps to build the hacker app:** This website reading user info using same your app permissions

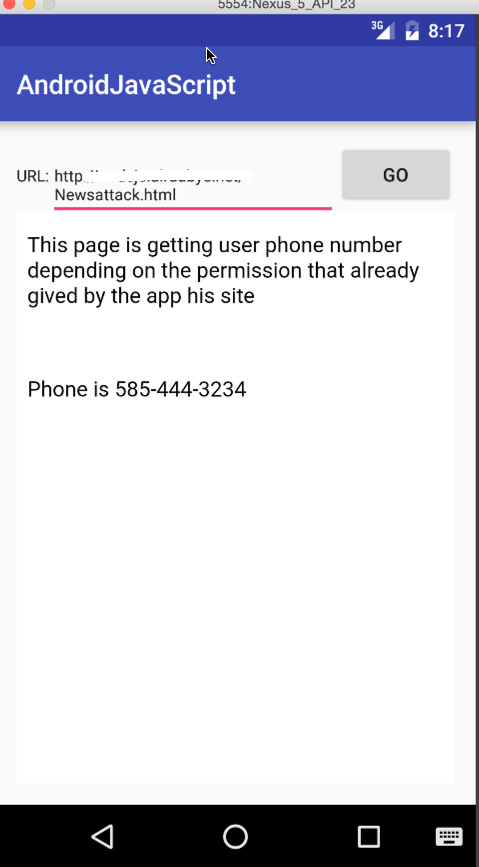
1. Hacker inspect will your website and he will see you are using Android function in your script



1. Hacker will insert same JavaScript In his website, when your users view his websites he will got user personal info through your app permissions

****

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



**Fix This Problem**

To fix this problem, we have to send sensitive data only to the websites that we authorize to access to this data like our websites, or we have to enable JavaScript to be run only in our website, see the code below

For send sensitive data only to the websites that we authorize.

|  |
| --- |
| Java |
| public class MainActivity extends AppCompatActivity { EditText etURL; //navigation url  WebView browser; // web browser  // host name  public String HostingURL="hostname";   @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*);   //Inject WebAppInterface methods into Web page by having Interface name 'Android'  browser.addJavascriptInterface(new WebAppInterface(), "Android");   // 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 ;  }  }   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();// "585-444-3234"; else  return null;  }   }  void LoadURL(){ /\* we could enable javascript to be run only in our website  if(etURL.getText().toString().indexOf(HostingURL)==0)  //Enable Javascript  browser.getSettings().setJavaScriptEnabled(true);  else  //Enable Javascript  browser.getSettings().setJavaScriptEnabled(false); \*/  //load the url that written in edittext to the webview  browser.loadUrl(etURL.getText().toString());  } /\* 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;  }    //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.

