CSCI 420-03: Analysis Plan

Student Name, Student Name
{student.name,student.name}@email.wm.edu

1 Research Questions

RQ1: *Do apps use all of the permissions they request?*
RQ2: *Are any of the following dangerous permission combinations included?*
`RECORD_AUDIO & INTERNET` *(eavesdropping),*
`ACCESS_FINE_LOCATION & RECEIVE_BOOT_COMPLETED` *(tracking),*
`CAMERA & INTERNET` *(stalking),*
`SEND_SMS & WRITE_SMS` *(use phone as spam bot)*
RQ3: *Do apps verify certifications incorrectly (Incorrect overrides of trust managers or error handlers methods)?*
RQ4: *Do apps use improper Hostname verification?*
RQ5: *Are apps using deprecated or vulnerable SSL protocols?*
RQ6: *Do apps have mixed SSL use/are vulnerable to SSL Stripping attacks?*
RQ7: *Are sensitive data or mutable objects used in any implicit intents?*
RQ8: *Is trusted content loaded within any webviews? If displaying user-provided content, is data loaded into webviews sanitized?*
RQ9: *Are applications that use webviews and* `addJavascriptInterface()` *correctly using the* `@JavascriptInterface` *annotation?*

2 Hypotheses

questions, along with a list of hypotheses.

3 Evaluation Plan

3.1 Permissions Misuse Experiment

This experiment tests whether apps use all the permissions they request, whether dangerous permission combinations are used, and whether users are prompted and give consent to every permission used.

3.1.1 Experimental Setup

We will scrape the MediaStore documentation for what permissions are associated with what constants (eg. CAMERA permission with ACTION_IMAGE_CAPTURE). After populating this dictionary, we will check to make sure that for every permission there is a corresponding constant found in the code, indicating that the permission has been used with MediaStore.

We will use simple string searches in the Manifest to look for the permission combinations.

We will check for lines with .requestPermissions and then the name of each permission, to verify that requests are happening for each permission.

3.1.2 Expected Results

Criteria for success include: a MediaStore API call exists for all permissions requested, no dangerous combinations are found, and a requestPermissions call exists for every permission.

False positives would be encountered if a dangerous combination is used, but there is no misuse of data, and both/all permissions in the combination are necessary to the app's core functionality, or non-MediaStore APIs are used with proper permissions.

3.2 Trust Managers and Error Handlers Experiment

This experiment is to test whether an app overrides a built in trust manager or error handler to forgo correct verification of certificates.

3.2.1 Experimental Setup

Will parse class files to see if trust manager or error handler methods are overridden, and then what is contained within the newly defined method.

3.2.2 Expected Results

The experiment will succeed if the app contains overridden trust manager or error handler methods when using SSL APIs, or app's overridden trust manager or error handler incorrectly handles certificates.

We will see some false positives if there are overridden methods that correctly handle certificates.

3.3 addJavascriptInterface Experiment

This experiment tests whether or not the application is being exposed to potentially dangerous outside sources depending on if the application uses a webview.

3.3.1 Experimental Setup

Source code will be parsed for the use of addJavascriptInterface(), and if successfully found, the use of the @JavascriptInterface annotation.

3.3.2 Expected Results

The experiment will succeed if the app contains no use of addJavascriptInterface, and if it uses addJavascriptInterface to only allow methods marked with it to be accessed by Javascript.

There will be false positives if the app includes addJavascriptInterface without any @JavascriptInterface annotations, however it is being used in a safe manner.

3.4 Mixed use SSL Experiment

This experiment tests whether an app uses SSL on and off. Ex: Switching between HTTP and HTTPS.

3.4.1 Experimental Setup

For this test, will parse through the class files to see instances of using libraries such as "HttpURLConnection" and check to make sure all are configured to use https consistently.

3.4.2 Expected Results

The app will succeed if it consistently (all cases) uses SSL when it communicates with http (uses https).

There may be false positives if an app makes an http request (not https) to a local server within a secure and controlled environment. While best practice is to use https, due to the nature of where this request was made, we can say that it is not a violation.

3.5 AllowAllHostnameVerifier Experiment

This experiment is to test whether an app uses the string AllowAllHostnameVerifier to allow all hostnames.

3.5.1 Experimental Setup

Will parse class files to see if this line is used

3.5.2 Expected Results

The experiment will succeed if the app contains no use of AllowAllHostnameVerifier.

There will be no false positives unless this exact string is used randomly in the program.

References