WINDOWS PHONE 8 APPLICATION SECURITY

HackInParis 2013

Dmitriy Evdokimov
Andrey Chasovskikh

About us

Dmitriy ‘D1g1’ Evdokimov

- Security researcher at ERPScan
    - Mobile security, RE, fuzzing, exploit dev etc.
- Editor of Russian hacking magazine
- DEFCON Russia (DCG #7812) co-organizer

Andrey Chasovskikh

- Software developer
- Windows Phone addict

Agenda

- Intro
- Security model
- First steps in Windows Phone 8
- Applications
- Application security
- Conclusion

INTRO

(ROOT@DRONE USER)# EOF 2011-01-13 15:01:58 (244 MB/S) - 'FONDO.PNG' SAVED (9045)

Intro

- 29 Oct 2012 – Windows Phone 8 released
- Based on Windows 8 core
  - ARM architecture
- Market share: 3,2% (Q1 2013, IDC)
- 145 000+ applications in Windows Phone Store

[ROOT@DRONE USER]# EOF 2011-01-13 15:01:58 (244 MB/S) - 'FONDO.PNG' SAVED [9045]

SECURITY MODEL

Chambers

- Trusted Computing Base (TCB)

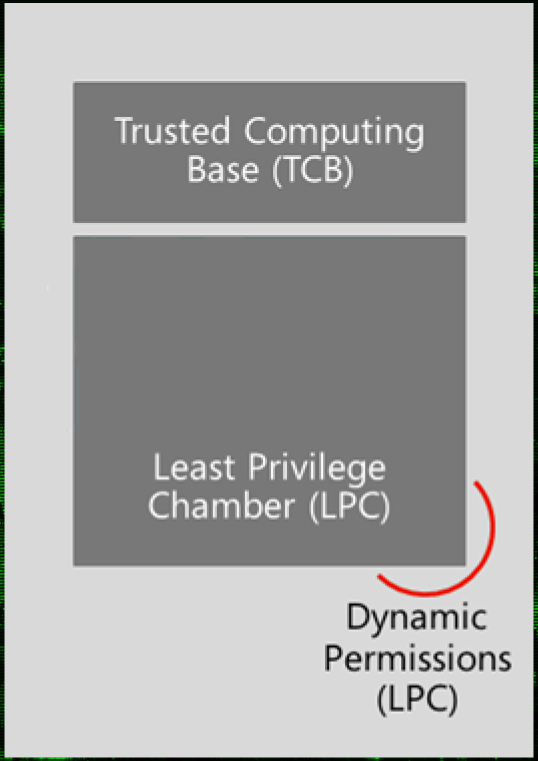
  Kernel, kernel-mode drivers

- Least Privileged Chamber (LPC)

  All other software: services,

  pre-installed apps,

  application from WP store


Trusted Computing Base (TCB)
Least Privilege Chamber (LPC)
Dynamic Permissions (LPC)

Capabilities

WMAppManifest.xml

Developers

- Network
- Camera
- NFC
- SD card access
- Wallet
- Speech recognition
- Front camera

Etc.

Total 27

OEM Developers

- Cell API
- Device management

Etc.

Total 39

System

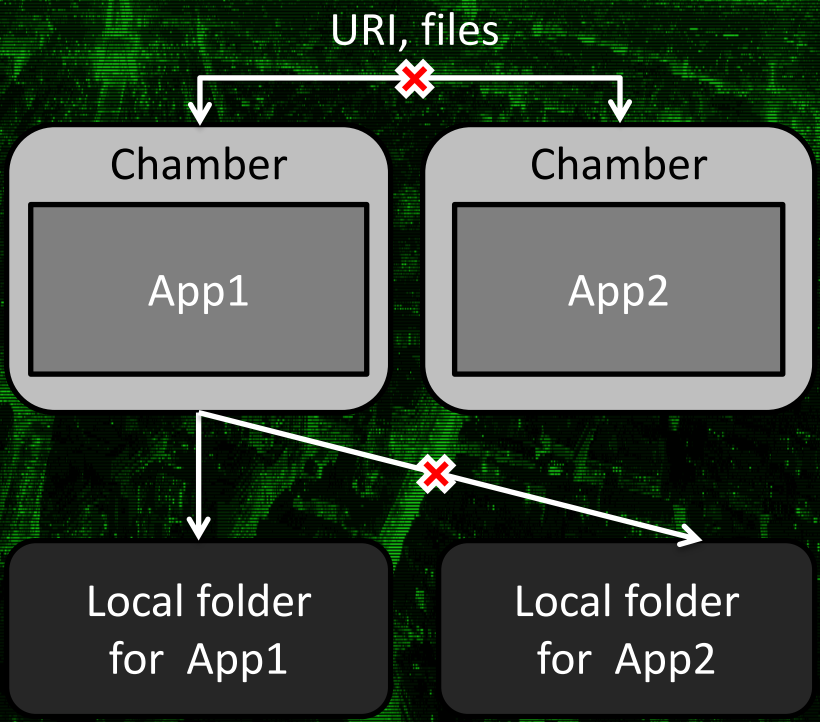
- Debug
- SMS API
- Live ID
- SIM API

Etc.

Total 350+

Sandboxing

- File system structure is hidden
- Local folder
  - Former isolated storage
- Limited app-to-app communication

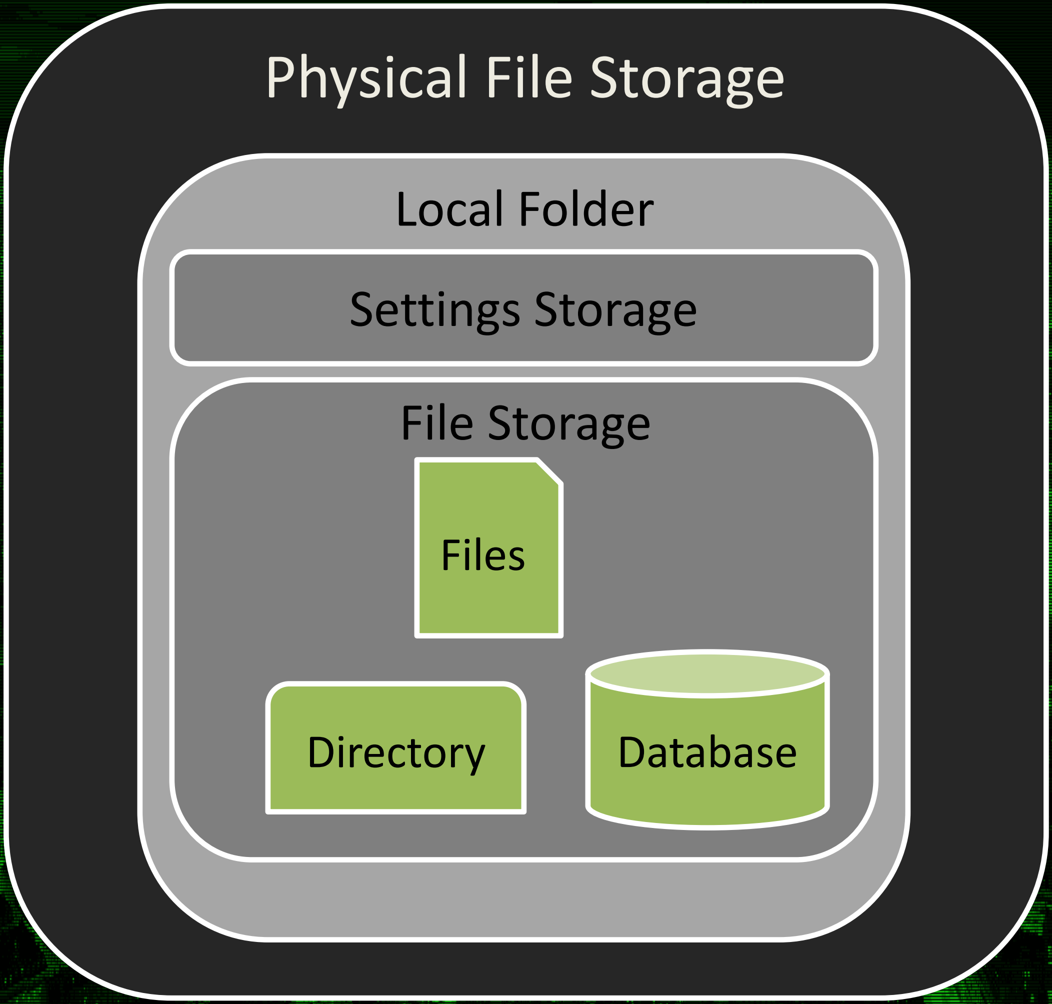


App-to-app communication

- File types associations
  - LaunchFileAsync()
  - Reserved: xap, msi, bat, cmd, py, jar etc.

- URI associations
  - LaunchUriAsync()
  - Reserved: http, tel, wallet, LDAP, rlogin, telnet etc.
  - Proximity communication using NFC

Local folder



Physical File Storage

Local Folder

Settings Storage

File Storage

Files

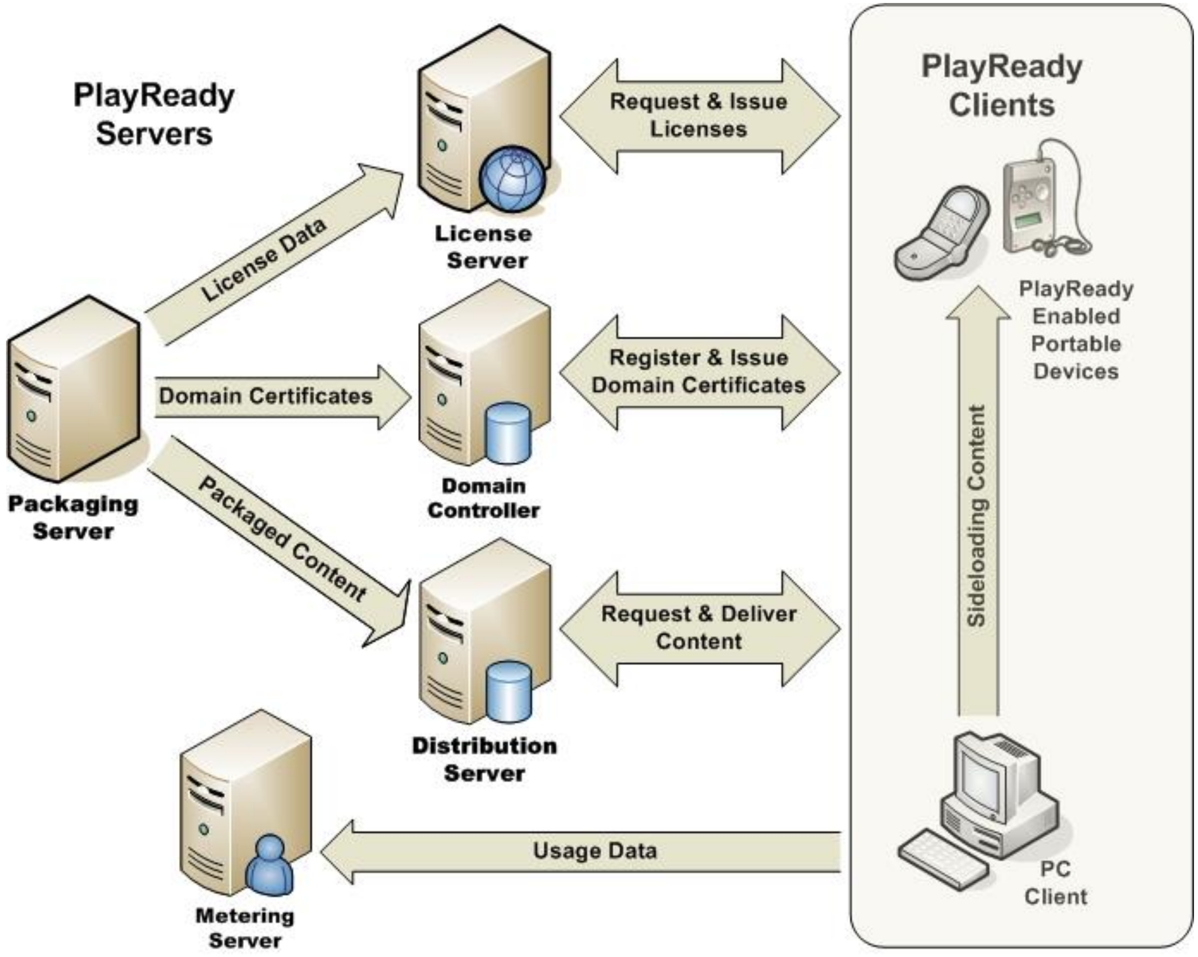
Directory

Database

Application protection

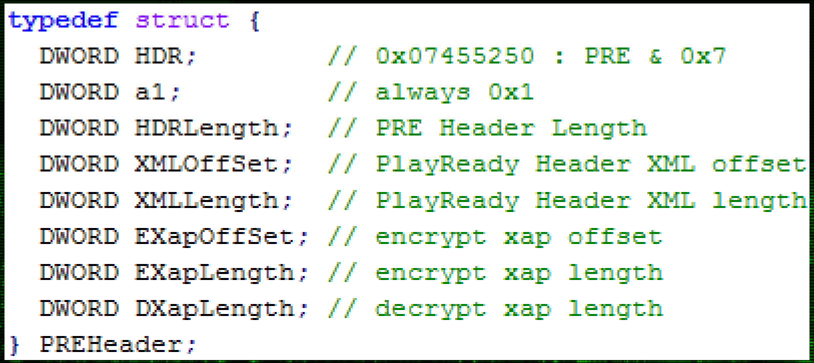
- All binaries are signed
- Application file is signed
  - Kind of checksum file is put into applications
- Certificate pinning for Store
- XAP file has DRM key

The Microsoft PlayReady Ecosystem

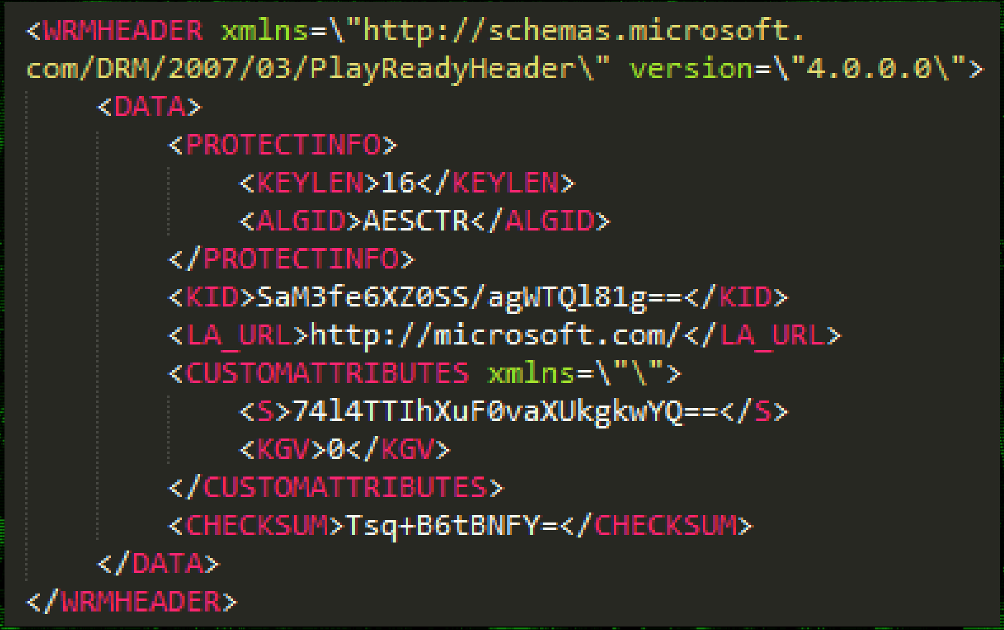


XAP file protection

- Before august 2012
  - ZIP archive
  - Sign
- After august 2012
  - New file format
  - PlayReady Header
  - AESCTR algorithm



```
typedef struct {
  DWORD HDR;          // 0x07455250 : PRE & 0x7
  DWORD a1;           // always 0x1
  DWORD HDRLength;    // PRE Header Length
  DWORD XMLOffSet;    // PlayReady Header XML offset
  DWORD XMLLength;    // PlayReady Header XML length
  DWORD EXapOffSet;   // encrypt xap offset
  DWORD EXapLength;   // encrypt xap length
  DWORD DXapLength;   // decrypt xap length
} PREHeader;
```



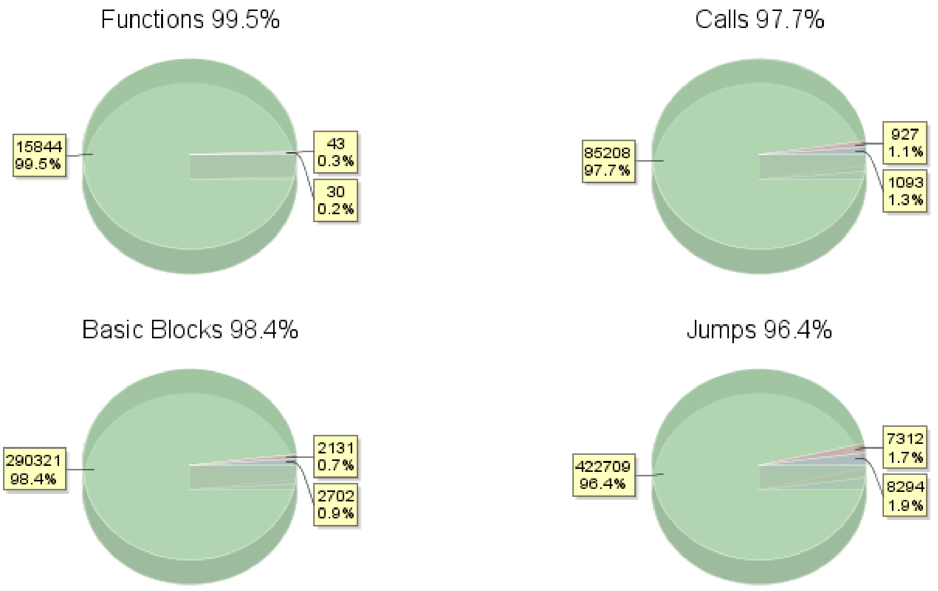
```
<WRMHEADER xmlns=\"http://schemas.microsoft.
com/DRM/2007/03/PlayReadyHeader\" version=\"4.0.0.0\">
    <DATA>
        <PROTECTINFO>
            <KEYLEN>16</KEYLEN>
            <ALGID>AESCTR</ALGID>
        </PROTECTINFO>
        <KID>SaM3fe6XZ0SS/agWTQl81g==</KID>
        <LA_URL>http://microsoft.com/</LA_URL>
        <CUSTOMATTRIBUTES xmlns=\"\">
            <S>74l4TTIhXuF0vaXUkgkwYQ==</S>
            <KGV>0</KGV>
        </CUSTOMATTRIBUTES>
        <CHECKSUM>Tsq+B6tBNFY=</CHECKSUM>
    </DATA>
</WRMHEADER>
```

FIRST STEPS IN WINDOWS PHONE 8

[ROOT@DRONE USER]# EOF 2011-01-13 15:01:58 (244 MB/S) - 'FONDO.PNG' SAVED [9045]

Windows 8 vs Windows Phone 8

- WP8 is migrating from the WinCE core to the WinNT core
- Win8/emulator (x86)
- WinRT/device (ARM)



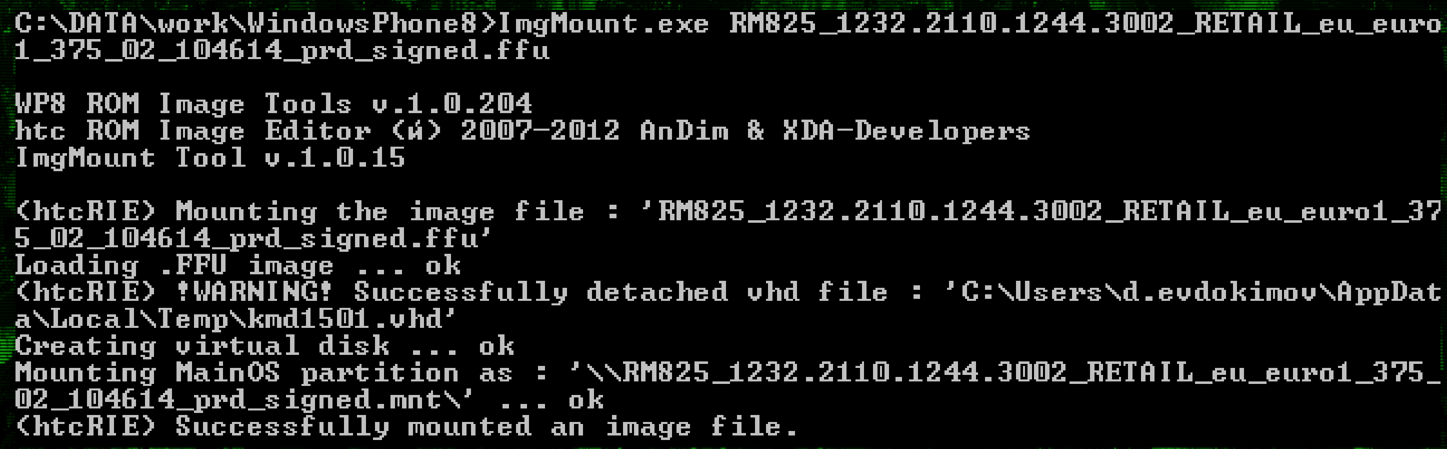
http://intrepidusgroup.com/insight/2012/12/windows-phone-8-and-windows-8-similarity/

WP8 emulator

- **Hyper-V images**
  - %ProgramFiles(x86)%\Microsoft SDKs\Windows Phone\v8.0\Emulation\Images\
- **Emulator vs. Device**
  - x86
  - Fake binaries
    - FakeLed.sys, Fakevibra.sys, FakeModem.dll etc.
  - Different user-agent
  - Prohibited to install apps from the Store

WP8 device

- Windows Phone 8 has standardized bootloader
  – Full flash images are available
- ImgMount tool
  – FFU Image file as a virtual hard drive



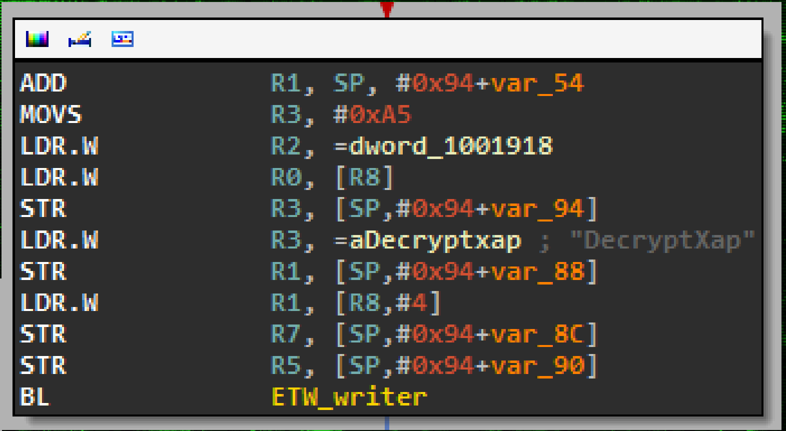
```
C:\DATA\work\WindowsPhone8>ImgMount.exe RM825_1232.2110.1244.3002_RETAIL_eu_euro
1_375_02_104614_prd_signed.ffu

WP8 ROM Image Tools v.1.0.204
htc ROM Image Editor (ώ) 2007-2012 AnDim & XDA-Developers
ImgMount Tool v.1.0.15

(htcRIE) Mounting the image file : 'RM825_1232.2110.1244.3002_RETAIL_eu_euro1_37
5_02_104614_prd_signed.ffu'
Loading .FFU image ... ok
(htcRIE) !WARNING! Successfully detached vhd file : 'C:\Users\d.evdokimov\AppDat
a\Local\Temp\kmd1501.vhd'
Creating virtual disk ... ok
Mounting MainOS partition as : '\\RM825_1232.2110.1244.3002_RETAIL_eu_euro1_375_
02_104614_prd_signed.mnt\' ... ok
(htcRIE) Successfully mounted an image file.
```

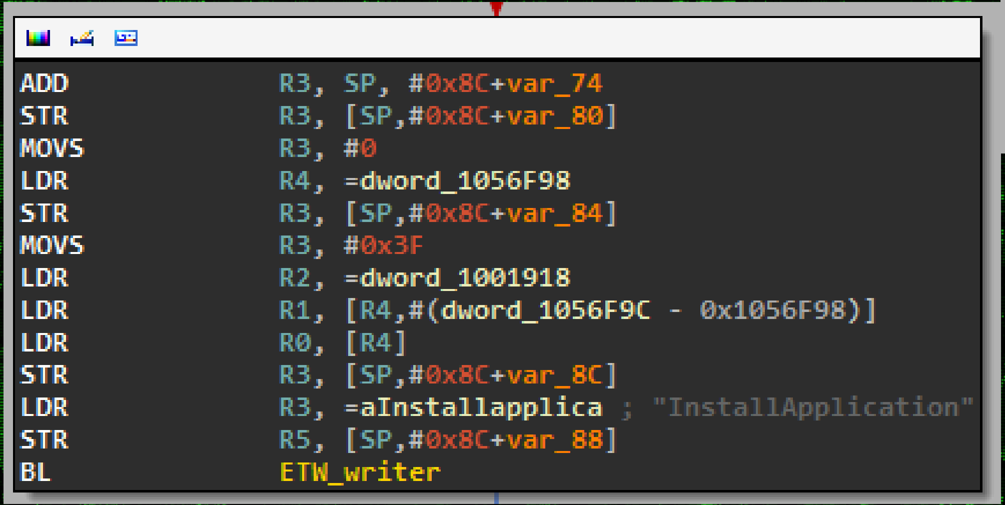
Reversing WP8 internals

- No debug symbols
- Tip: restore information from Event Tracing for Windows (ETW)
- Use IDAPython



```
ADD             R1, SP, #0x94+var_54
MOVS            R3, #0xA5
LDR.W           R2, =dword_1001918
LDR.W           R0, [R8]
STR             R3, [SP,#0x94+var_94]
LDR.W           R3, =aDecryptxap ; "DecryptXap"
STR             R1, [SP,#0x94+var_88]
LDR.W           R1, [R8,#4]
STR             R7, [SP,#0x94+var_8C]
STR             R5, [SP,#0x94+var_90]
BL              ETW_writer
```

*InstallerWorker.exe



```
ADD             R3, SP, #0x8C+var_74
STR             R3, [SP,#0x8C+var_80]
MOVS            R3, #0
LDR             R4, =dword_1056F98
STR             R3, [SP,#0x8C+var_84]
MOVS            R3, #0x3F
LDR             R2, =dword_1001918
LDR             R1, [R4,#(dword_1056F9C - 0x1056F98)]
LDR             R0, [R4]
STR             R3, [SP,#0x8C+var_8C]
LDR             R3, =aInstallapplica ; "InstallApplication"
STR             R5, [SP,#0x8C+var_88]
BL              ETW_writer
```

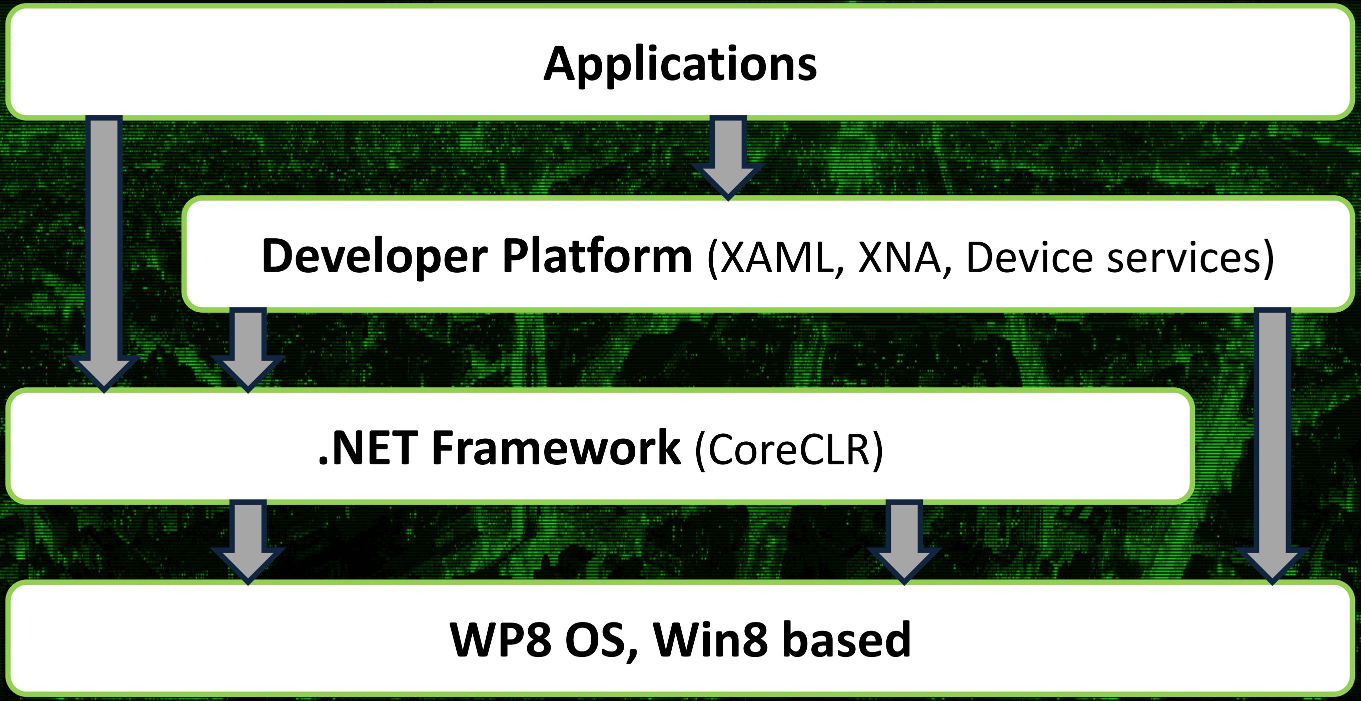
Windows API calls

- Full Windows API is not available by default
- Originally posted on XDA for WindowsRT apps
  - Find kernerbase.dll address (“MZ”) -> Get “LoadLibraryA” and “GetProcAddress” functions -> call any function you want
  - http://bit.ly/Uw2Gk6
- Works for Windows Phone 8

APPLICATIONS

(ROOT@DRONE USER)# EOF 2011-01-13 15:01:58 (244 MB/S) - 'FONDO.PNG' SAVED (9045)

.NET and CLR



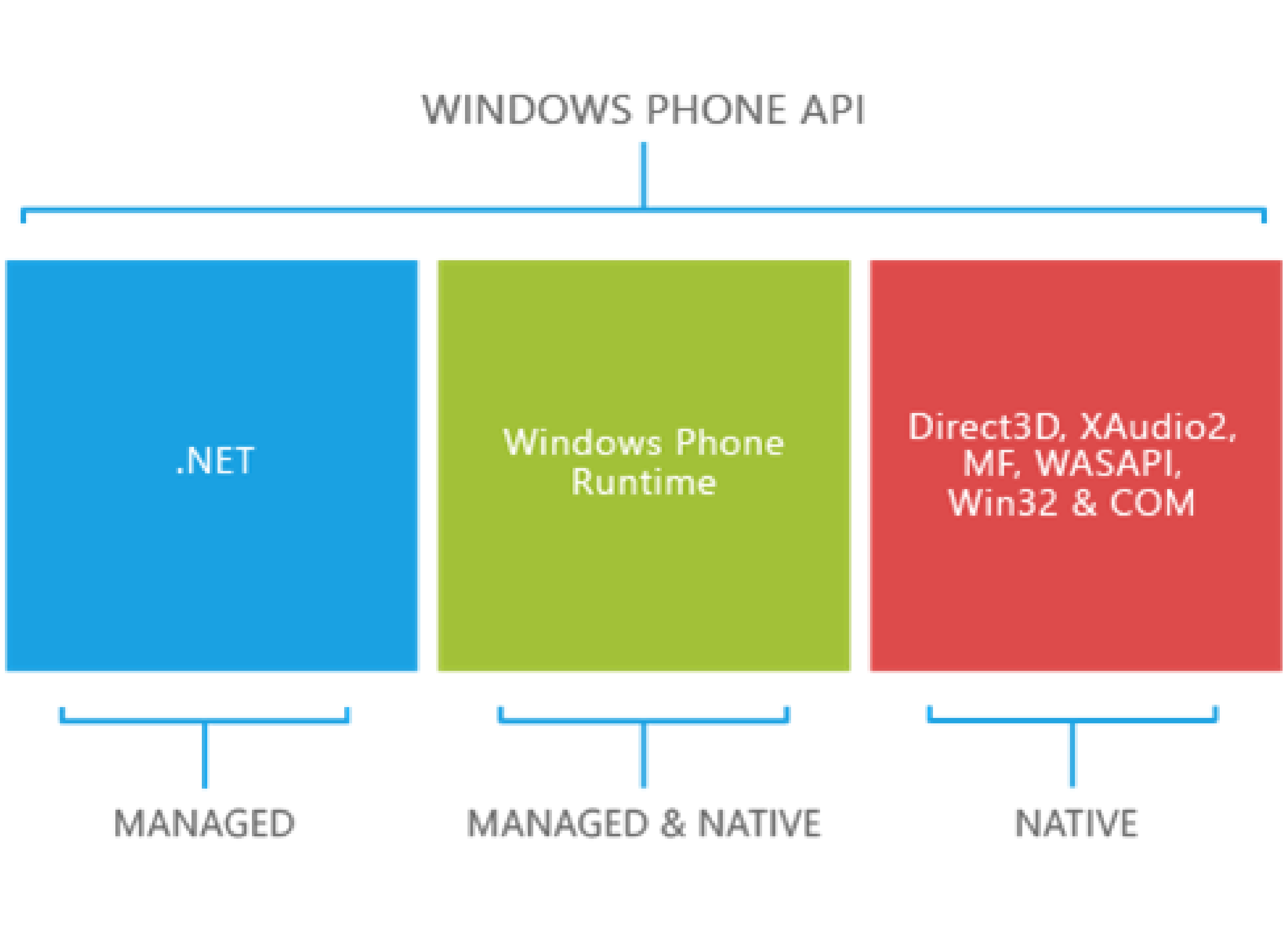
Applications

Developer Platform (XAML, XNA, Device services)

.NET Framework (CoreCLR)

WP8 OS, Win8 based

Frameworks



WINDOWS PHONE API

.NET

Windows Phone Runtime

Direct3D, XAudio2, MF, WASAPI, Win32 & COM

MANAGED

MANAGED & NATIVE

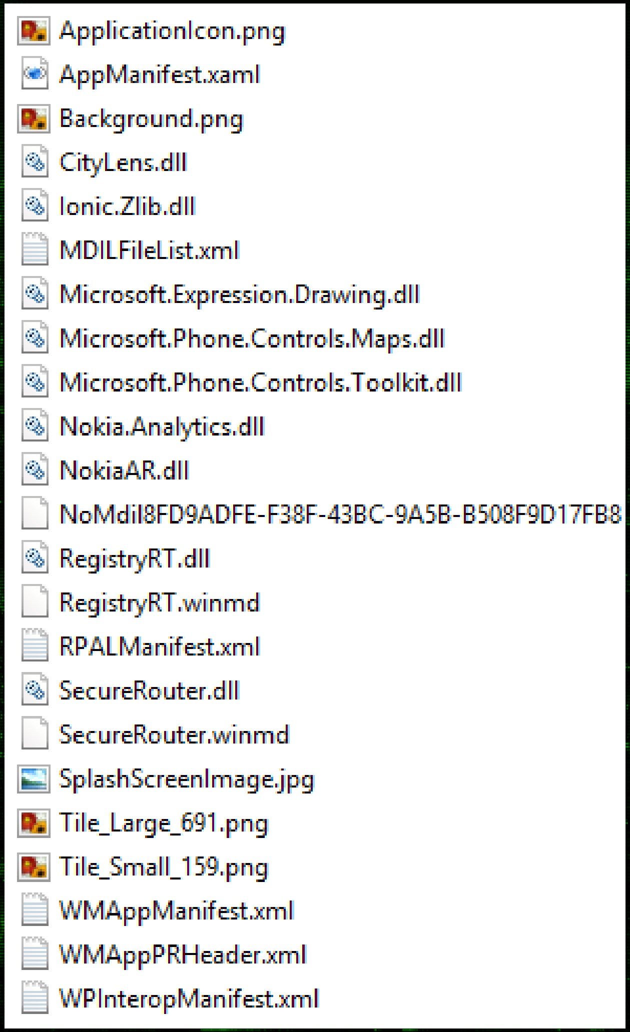
NATIVE

Application kinds

- Microsoft
- OEM
  - XAP files are not encrypted (~ZIP)
  - C:\PROGRAMS\CommonFiles\Xaps\
- Windows Phone Store apps
  - C:\Data\Programs\{ProductID}\Install\
- Company applications
  - XAP files are not encrypted (~ZIP)
  - Company hubs
- Developer applications
  - Need developer unlock

Application file structure

- Application assemblies (in various formats)
- Resources
- AppManifest.xaml
- WMAppManifest.xml



ApplicationIcon.png
AppManifest.xaml
Background.png
CityLens.dll
Ionic.Zlib.dll
MDILFileList.xml
Microsoft.Expression.Drawing.dll
Microsoft.Phone.Controls.Maps.dll
Microsoft.Phone.Controls.Toolkit.dll
Nokia.Analytics.dll
NokiaAR.dll
NoMdil8FD9ADFE-F38F-43BC-9A5B-B508F9D17FB8
RegistryRT.dll
RegistryRT.winmd
RPALManifest.xml
SecureRouter.dll
SecureRouter.winmd
SplashScreenImage.jpg
Tile_Large_691.png
Tile_Small_159.png
WMAppManifest.xml
WMAppPRHeader.xml
WPInteropManifest.xml

APPLICATION SECURITY

(ROOT@DRONE USER)# EOF 2011-01-13 15:01:58 (244 MB/S) - 'FONDO.PNG' SAVED (9045)

Security?!

“One of the goals of the Windows Phone app platform is to foster the creation of apps that are *secure by design and secure by default*.”

Security for Windows Phone

Application entry points

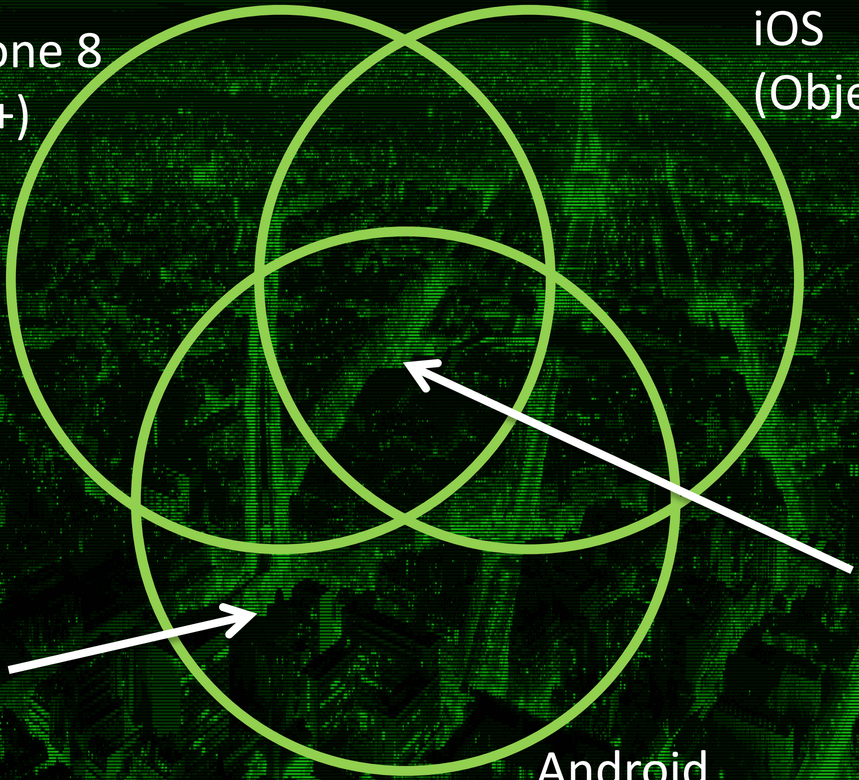
- User input
- SD card
- Sockets
- URI

- Web
- Bluetooth
- NFC
- Speech2Text

Green – Windows Phone 7

White – Windows Phone 8

Vulnerabilities



Windows Phone 8
(C#/VB/C/C++)

iOS
(Objective-C)

Android
(Java)

Platform specific vulnerabilities

Platform independent vulnerabilities

Note: Main programming languages in brackets

Work with SD card

- WP8 allows only read operations
- Only registered file types
- Files on SD cards are not encrypted

OS	Details
iOS	Work with SD card is absent
Android	READ/WRITE

Privacy

- Device Unique ID
  - Requires ID_CAP_IDENTITY_DEVICE
  - DeviceExtendedProperties.GetValue(“DeviceUniqueId”)
- Windows Live Anonymous ID
  - Requires ID_CAP_IDENTITY_USER
  - UserExtendedProperties.GetValue(“ANID2”)
- Both identifiers are per-publisher

OS	Details
iOS	UDID (apps that use UDIDs are no longer accepted, from May 1, 2013)
Android	telephonyManager.getDeviceId()

Privacy, part 2

- Device name, manufacturer, firmware versions
  - – Requires ID_CAP_IDENTITY_DEVICE
  - – DeviceStatus class
- Location tracking
  - – ID_CAP_LOCATION
  - – GeoCoordinateWatcher class

OS	Details
iOS	UDID (apps that use UDIDs are no longer accepted, from May 1, 2013)
Android	telephonyManager.getDeviceId()

Secure storage

- Device can be encrypted (not for all countries)
  - BitLocker 2.0/TPM
  - Available only in business settings
- Data Protection API (DPAPI)
- System.Security.Cryptography
- Algorithms: AES, HMACSHA1, HMACSHA256, Rfc2898DeriveBytes, RSA, SHA1, SHA256

OS	Details
iOS	Keychain, /System/Library/Frameworks/Security.framework
Android	android.security.KeyChain (from 4.0)

Data leak

- Keyboard cache is isolated per-application
- Cache for applications that access internet
  - Controlled by OS

OS	Details
iOS	plist, Custom created documents, Preferences, Logs, Cache data, Keyboard cache, Pasteboard cache, Cookies
Android	shared_preference, logs, external storage, MODE_WORLD_READABLE or MODE_WORLD_WRITETABLE

Work with URI

- Handling function: MapUri()

- Filter user input
- Exclude critical arguments from URI
  - Ex.: prgrm://command?request=data&role=admin

OS	Details
iOS	openURL(), handleOpenURL()
Android	android.net.Uri class

Cross-site scripting (XSS)

- WebBrowser control (based on IE10)
- JavaScript is disabled by default
- To see if enabled:
  - WebBrowser.IsScriptEnabled = true
  - <WebBrowser IsScriptEnabled = “True” />

OS	Details
iOS	UIWebView Class + stringByEvaluatingJavaScriptFromString() shouldStartLoadWithRequest()

Android	WebView.getSettings().setJavaScriptEnabled();
	WebView.getSettings().setPluginsEnabled();

Directory traversal

- Local folder API accepts paths with traversal
  - IsolatedStorageFile class (WP7)
  - StorageFolder class
- Win32 storage API

OS	Details
iOS	contentsAtPath, fileHandleForReadingAtPath, _fopen etc.
Android	ContentProvider + incorrect or missing rights, files functions

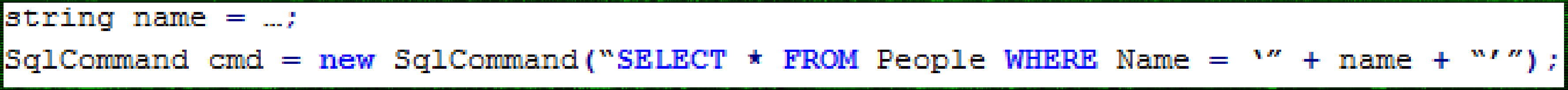
XML External Entity (XXE)

- System.Xml namespace
  - Entity resolving is prohibited by default
- Entities can be resolved by using custom XmlResolver for XmlDocument

OS	Details
iOS	libXML2 + _xmlParseMemory, NSXMLParser + setShouldResolveExternalEntities:YES
Android	setFeature(external-general-entities, True)

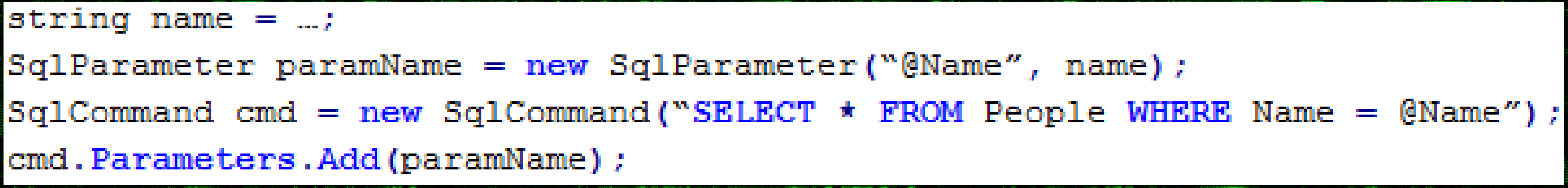
SQL injection

- Bad:



```
string name = …;
SqlCommand cmd = new SqlCommand(“SELECT * FROM People WHERE Name = ‘” + name + “’”);
```

- Good:



```
string name = …;
SqlParameter paramName = new SqlParameter(“@Name”, name);
SqlCommand cmd = new SqlCommand(“SELECT * FROM People WHERE Name = @Name”);
cmd.Parameters.Add(paramName);
```

OS	Details
iOS	sqlite3_exec()
Android	query(), rawQuery()

Memory corruption bugs

- Developers can use native code
- Format string, BoF, use-after-free etc.
  - C/C++ functions
- Compilation flags: /sdl, /GS, /DYNAMICBASE, /NXCOMPAT

OS	Details
iOS	–fPIE, –fstack-protector-all, -fobjc-arc
Android	Only in native libs, -fstack-protector, -Wformat-security, NX, ASLR, PIE

CONCLUSION

```
(ROOT@DRONE USER)# EOF 2011-01-13 15:01:58 (244 MB/S) - 'FONDO.PNG' SAVED (9045)
```

Conclusion

- Windows Phone 8 is pretty secure
- Greater attack surface
- Security-related API
  - More flexible than in iOS
  - More simple than in Android

Q&A

Dmitry ‘D1g1’ Evdokimov

- d.evdokimov@erpscan.com
- @evdokimovds

Andrey Chasovskikh

- http://andreycha.info
- @andreycha