



# BeerTalk – Windows Phone 8.1

Alexandre Herzog  
Cyrill Bannwart

Compass Security Schweiz AG  
Werkstrasse 20  
Postfach 2038  
CH-8645 Jona

Tel +41 55 214 41 60  
Fax +41 55 214 41 61  
[team@csnc.ch](mailto:team@csnc.ch)  
[www.csnc.ch](http://www.csnc.ch)

# Agenda



## Introduction

## The Windows View

- ◆ Windows Environment
- ◆ Attack Surface
- ◆ Breaking Out

## The Mobile View

- ◆ Sandboxing & Encryption

## Findings

- ◆ MDM Integration
- ◆ WiFi Sense
- ◆ Low Level Storage API

## Conclusion

Third player, investing in it

Microsoft is a major player on the business desktop, servers and software

- ◆ Just missing the mobile part
- ◆ But attempting to catch up with the acquisition of Nokia
- ◆ Still understands / answers best companies' needs

Global convergence

- ◆ Business & private
- ◆ Mobile & fix

Something new to look at (and maybe break? ;-)

- ◆ Our focus was the Windows Phone platform itself

Personal feeling: Microsoft is never as good as when it's challenged

Vaudois exilé d'abord en Valais, then Wellington (NZ) und jetzt Zürich

- ◆ CTO of Compass Security Schweiz AG
- ◆ Former sysadmin & developer for banks

Strong interest in Windows security

- ◆ MAS thesis about "Crypto-based security mechanism in Windows and .NET"
- ◆ One of the first to publish about the Group Policy Preferences (GPP) flaw
- ◆ Security advisory about serialization in the .NET framework (CVE-2013-1330 patched in MS13-067 (SharePoint), MS13-105 (Outlook Web Access) etc)
- ◆ Invited by Microsoft to BlueHat back in December 2013
- ◆ ...

More of a server / workstation than mobile guy

- ◆ But always ready to break (out of) Windows / Microsoft technologies!

Joined Compass Security in 2013

- ◆ IT Security Analyst
- ◆ Security trainings teacher
- ◆ Mobile apps developer

Electrical Engineer with a strong interest in

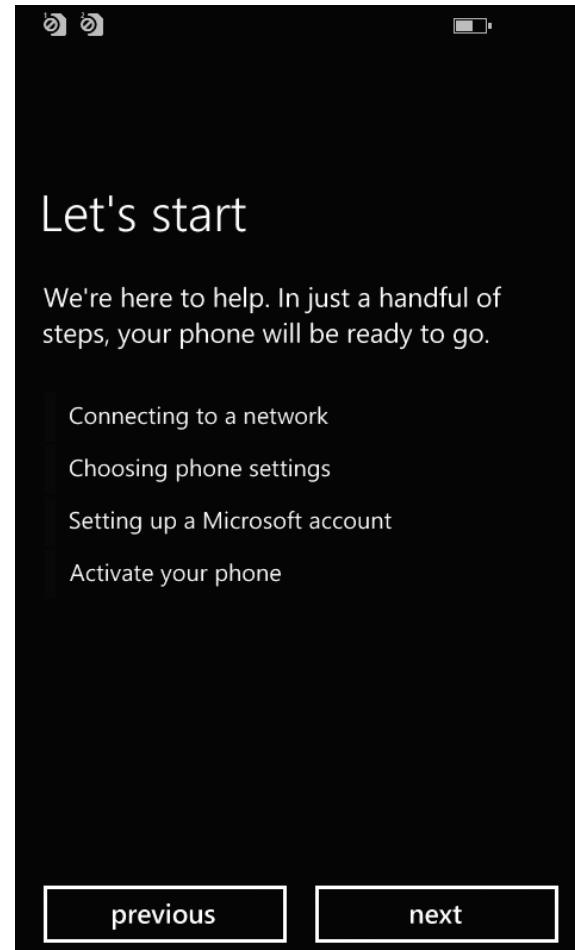
- ◆ Embedded devices
- ◆ Network communications

Mostly dealing with Network, Unix and (iOS) Mobile apps

Giving Security Trainings for

- ◆ Secure Mobile Apps
- ◆ iPhone & iPad Security

# Let's get started



Crash dumps are always useful and a good start...

```
ALLUSERSPROFILE=C:\Data\ProgramData
APPDATA=C:\Data\Users\DefApps\AppData\Roaming
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=Windows Phone
ComSpec=C:\windows\system32\cmd.exe
FP_NO_HOST_CHECK=NO
LOCALAPPDATA=C:\Data\Users\DefApps\APPDATA\Local\Packages\8dd8d60d-8b28-4e52-b113-
c2aac34b9ac3_yhxz8gp8y0q0t\AC
NUMBER_OF_PROCESSORS=4
OS=Windows_NT
Path=C:\windows\system32;C:\windows;C:\Programs\CommonFiles\System;C:\wtt;C:\data\test\bin;
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE=ARM
...
ProgramData=C:\Data\ProgramData
ProgramFiles=C:\Program Files
PUBLIC=C:\Data\Users\Public
SystemDrive=C:
SystemRoot=C:\windows
TEMP=C:\Data\Users\DefApps\APPDATA\Local\Packages\8dd8d60d-8b28-4e52-b113-
c2aac34b9ac3_yhxz8gp8y0q0t\AC\Temp
TMP=C:\Data\Users\DefApps\APPDATA\Local\Packages\8dd8d60d-8b28-4e52-b113-
c2aac34b9ac3_yhxz8gp8y0q0t\AC\Temp
USERDOMAIN=Windows Phone
USERNAME=DefApps
USERPROFILE=C:\Data\Users\DefApps
windir=C:\windows
```

We focused on 3 aspects for the Windows part:

## (Ab)Use of Windows Utilities and Features

- ◆ Can I gather information or perform undesired actions using built-in features?

## Application Attack Surface

- ◆ Or how can I misuse Internet Explorer to run unwanted code?

## Development and APIs

- ◆ List the documented APIs and see what a developer might run as code

A Windows desktop is user (and attacker) friendly

- ◆ Lots of information (eventlogs, detailed error messages, ...)
- ◆ Lots of settings to influence (Control Panel, file & registry access, ...)
- ◆ Built-in programs and features (notepad, sticky keys for accessibility, ...)
- ◆ Various ways to execute code (bat, vbs, WMI, PowerShell, compilers, ...)

This regardless of the target (workstation, app / Citrix server, ATM, ...)

Windows Phone exposes only

- ◆ Very little information or settings are available
- ◆ No interesting default app (you have to download e.g. app «files» separately)
- ◆ No possibility to «run» stuff
- ◆ No sticky keys
- ◆ It's so impossible to e.g. get the UEFI settings details of the phone...

Internet Explorer is the most interesting app on the phone



We can't download this file, because  
Windows Phone doesn't support this file  
type.

**Can't complete**

Can't open file Ink\_DriveC.Ink.  
Error code: -2147024809. You can mention this  
code when providing feedback.

ok

All failed abuse scenarios (so good for the security)

- ◆ Run VBScript within the browser
- ◆ Browse the local file system using file:///
- ◆ SMB connect-back from the phone to the attacker
- ◆ No way to download and execute e.g. .bat, .exe, .vbs, ... files
- ◆ The VB/XSLT <msxsl:script> bypass of Spartan[VB\_XSLT] (but crashes IE when the page is shared)
- ◆ Link files (.lnk) are not executed as well...

If no app provide me the desired feature, let's code my own!

The C++ and .NET APIs are trimmed down & restricted,  
preventing breaking out / unwanted actions

All failed abuse scenarios (so good for the security)

- ◆ Controlling processes or threats to fork new content within an application
- ◆ Running arbitrary commands using Shell.Execute
- ◆ Accessing WMI (Windows Management Instrumentation) to gather information and execute arbitrary commands
- ◆ Running PowerShell for the same reasons

Of course, not all options have been explored so far, e.g.

- ◆ Is arbitrary execution of commands possible, via e.g. Lambda expressions?
- ◆ Can the restricted APIs be misused?  
(e.g. attempt to load an assembly not present within the Windows Phone SDK)
- ◆ In-depth audit of the Protected Data / Vault feature
- ◆ Study of the AppContainer and SIDs separation
- ◆ Understand the steps involved in the application signing process  
(and their capabilities restrictions)
- ◆ Subversion of accorded capabilities  
(capabilities seem to be labels assigned to a given process).
- ◆ Content of C:\WTT
- ◆ Corruption via the video driver e.g. within the browser (WebGL)
- ◆ ...

So Windows guru, what did you actually achieve on this device?

*...really not much...*

Some information leaks from application crash dumps, e.g.

- ◆ User running the app (or let's call it rather the App Container context)
- ◆ List of defined drives:
  - ◆ C:\
  - ◆ D:\ (probably SD card)
  - ◆ U:\ (probably a mapping to C:\data\).
  - ◆ PATH variable contains unknown folder C:\WTT\.
  - ◆ Data seems shared via C:\Data\Share
  - ◆ C:\windows\system32\cmd.exe does not exist

# Agenda



## Introduction

## The Windows View

- ◆ Windows Environment
- ◆ Attack Surface
- ◆ Breaking Out

## The Mobile View

- ◆ Sandboxing & Encryption

## Findings

- ◆ MDM Integration
- ◆ WiFi Sense
- ◆ Low Level Storage API

## Conclusion

# The Mobile View



Create your own App



CREATE >



PUBLISH >

Marketplace



<http://www.microsoft.com/silverlight/windows-phone/>

## Sandboxing

- ◆ Attack Surface Reduction (Least Privilege)
- ◆ User consent and control (Capabilities)
- ◆ Isolation (AppContainer, dedicated SIDs)

## Malware Resistance

- ◆ UEFI, Trusted / Secure Boot
- ◆ System and App Integrity (Code Signing)
- ◆ Windows Phone Store (Automated Malware Scan)

## Exploit Mitigation

- ◆ Address Space Layout Randomization (ASLR)
- ◆ Data Execution Prevention (DEP)

## Encryption

- ◆ BitLocker (AES-128, TPM)

## AppContainer

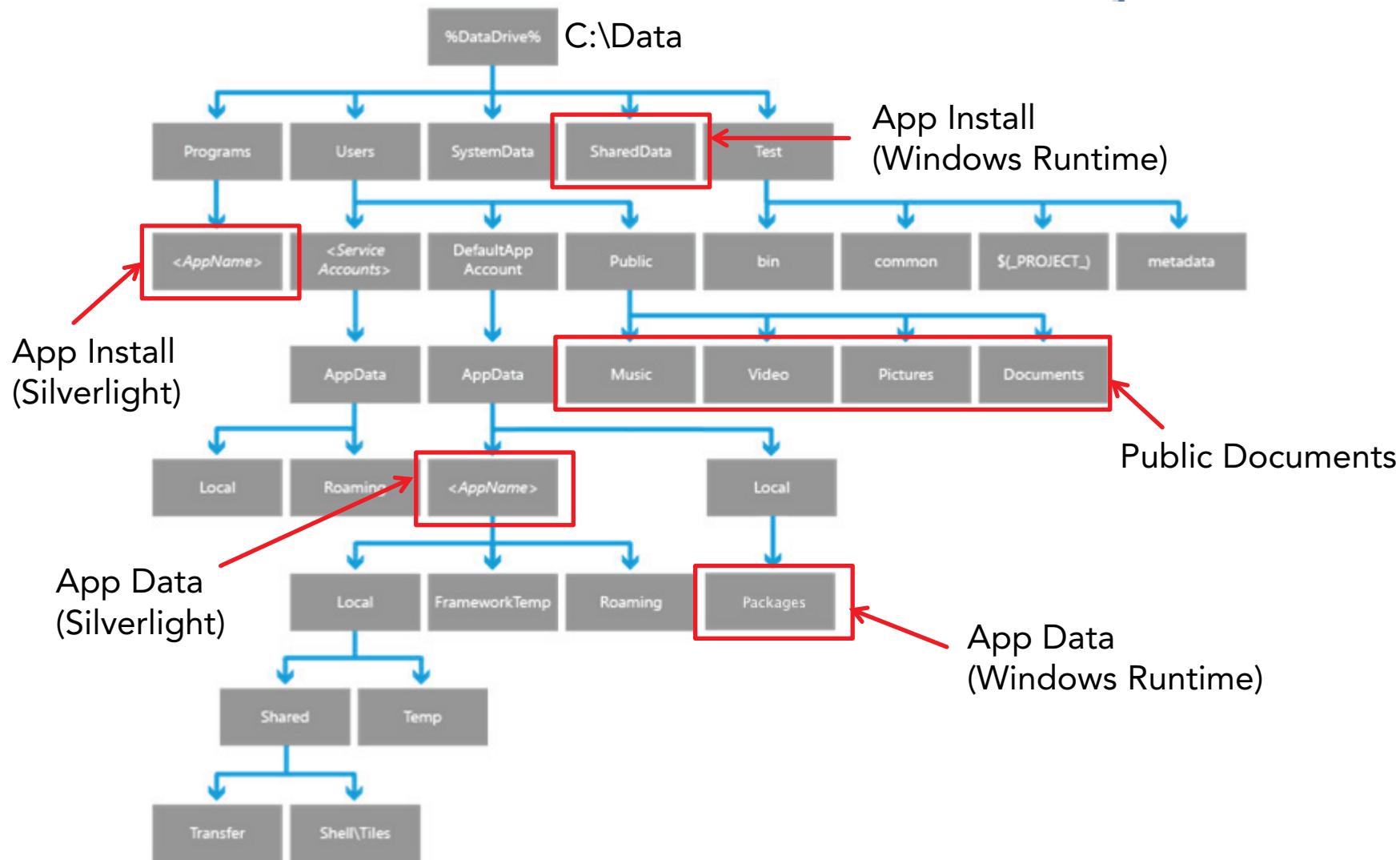
- ◆ Isolation
- ◆ Data Access
- ◆ Credentials
- ◆ Roaming
- ◆ Sharing Data
- ◆ Encrypting data

## Capabilities

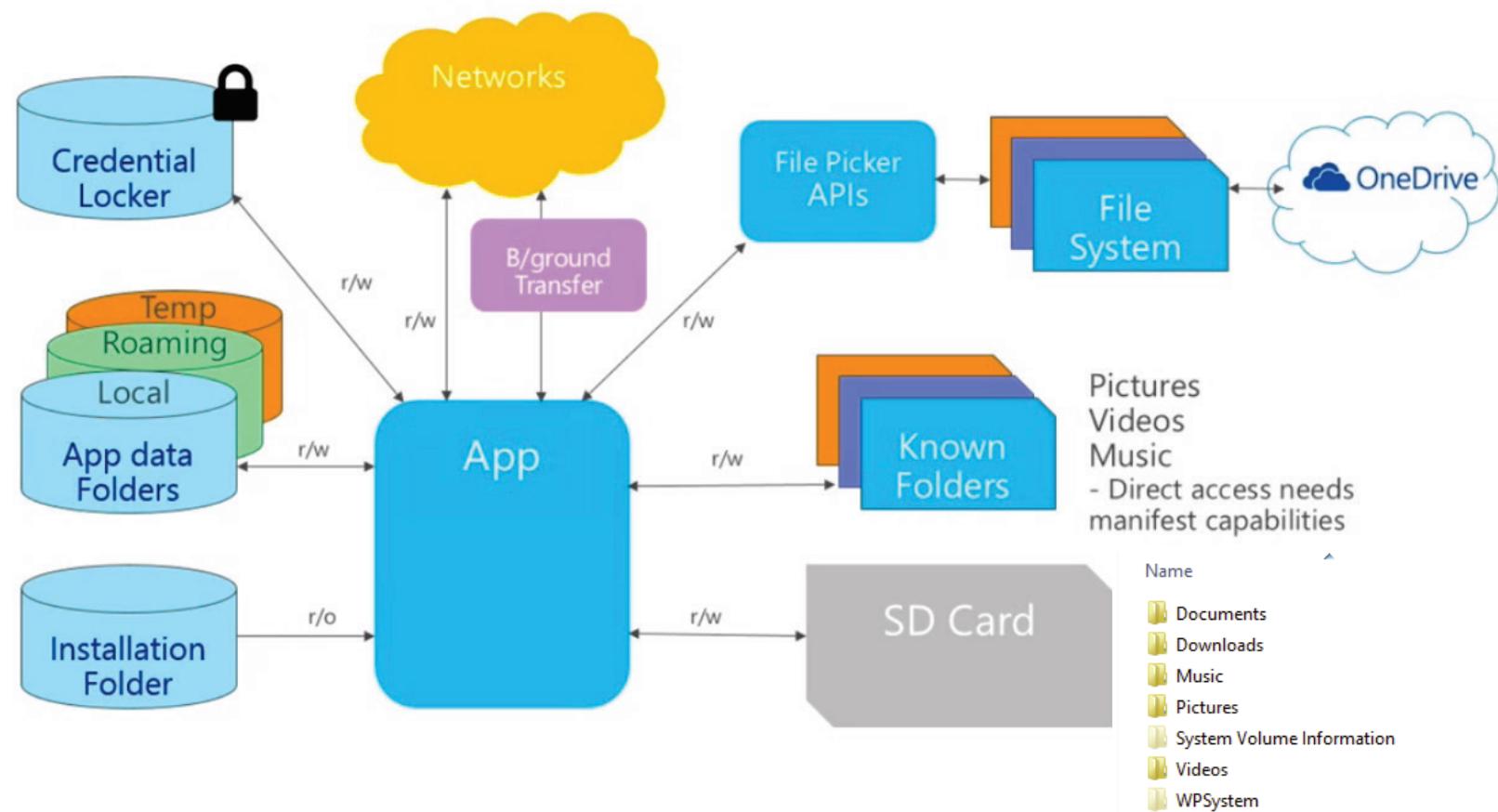
## Restricted APIs

- ◆ Isolated Storage

# File System Overview



## Locations where apps can access data



<http://channel9.msdn.com/Series/Building-Apps-for-Windows-Phone-8-1/09>

# Storing Credentials

Secure Storage & Roaming of Credentials



## Isolation

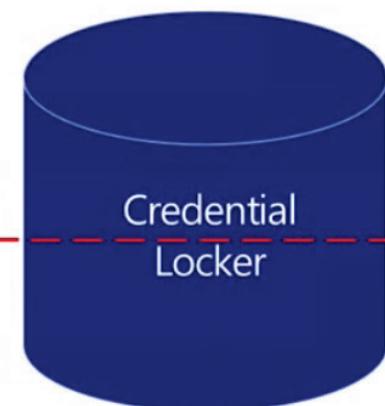
Apps can only access their own credentials

username / password  
pairs only

App A

App Sandbox Boundary

App B



### Example:

```
var vault = new PasswordVault();
PasswordCredential cred = new PasswordCredential("account", username, password);
vault.Add(cred);
```

# Roaming

Sharing data e.g. credentials across devices



## Roaming

Credentials roam across trusted devices



# Sharing Data

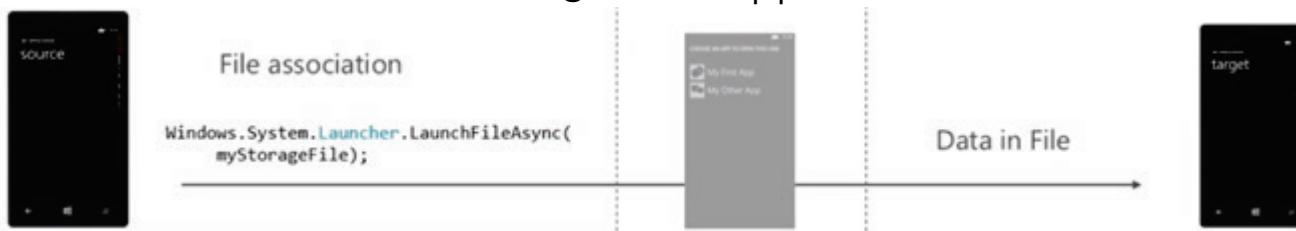


Sharing data between apps works using:

- URI Association, where the registered app obtains the data stored in the URI



- File Association, where the registered app obtains the file content



- Share Contract, allowing custom DataPackages to be shared



Disk Encryption using BitLocker is disabled by default.

- ◆ End-user cannot enable or disable encryption.
- ◆ Can only be activated through ActiveSync or MDM Policy.

Applications can use DPAPI to protect confidential data.

- ◆ DPAPI (Data Protection API) generates and stores a cryptographic key by using the user and device credentials.
- ◆ Every app gets its own decryption key, which is created when the app is run for the first time.
- ◆ The keys will persist across updates to the app.

<https://msdn.microsoft.com/en-us/windows/apps/hh487164.aspx>

## Software capabilities

- ◆ Capability elements are entries in the manifest file that notify the user while installing the app of special software capabilities that your app receives.
- ◆ E.g. Provide access to location services

## Hardware requirements

- ◆ A requirement element is an optional entry in the app manifest file that is used to specify hardware requirements and limit the exposure of an app to users that have a phone with the necessary hardware to run the app.
- ◆ E.g. Requiring Near Field Communication (NFC)

## Functional capabilities

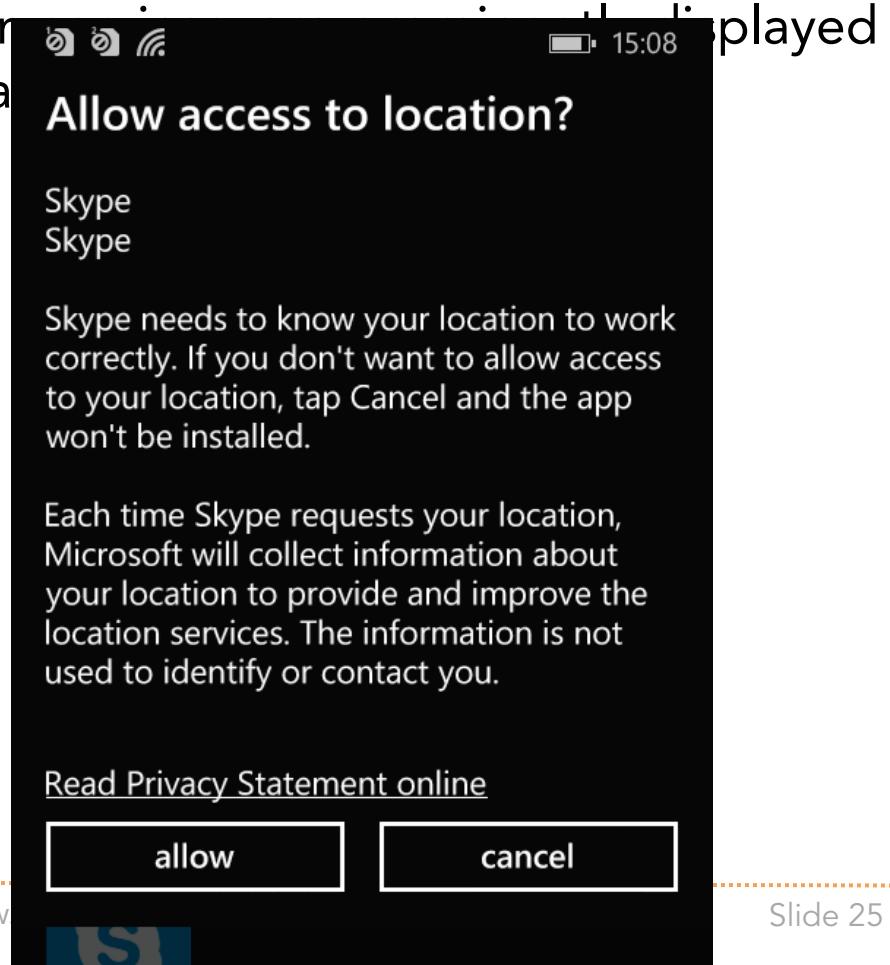
- ◆ A functional capability is an optional entry in the app manifest file that indicates that your app is requesting a hardware capability of the phone which is present, but not automatically granted.
- ◆ E.g. Requesting higher memory limits (in Windows Phone 8.0 only)

# Software capabilities

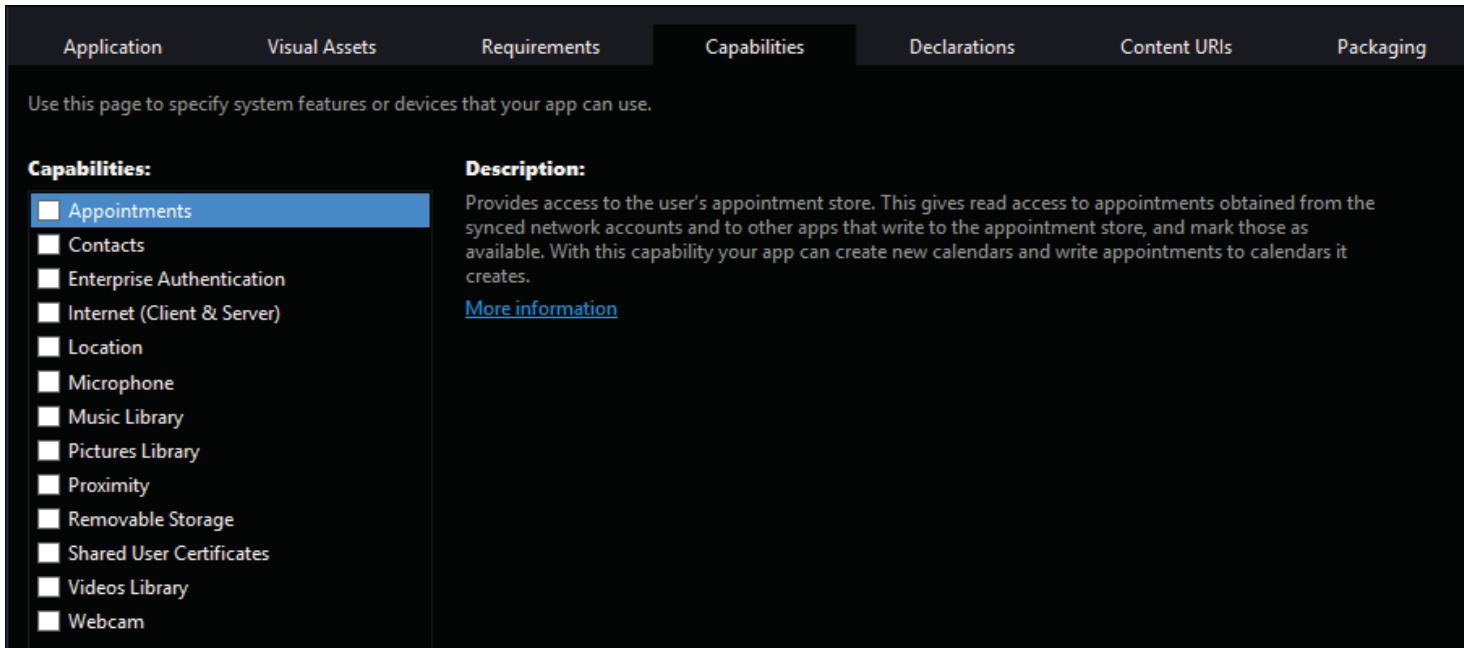


The capabilities listed in the app manifest are disclosed to a user when they view an app for purchase in Windows Phone Store.

Some capabilities, such as location, are displayed so the user is fully aware that an app requests location information.



## Setting capabilities using Microsoft Visual Studio 2013 Express:

A screenshot of the Microsoft Visual Studio 2013 Express interface showing the "Capabilities" tab selected in the top navigation bar. The page title is "Setting capabilities using Microsoft Visual Studio 2013 Express:". Below the title is a sub-instruction: "Use this page to specify system features or devices that your app can use." On the left, there is a list of "Capabilities" with checkboxes next to them. The checkbox for "Appointments" is checked and highlighted with a blue background. The other capabilities listed are: Contacts, Enterprise Authentication, Internet (Client & Server), Location, Microphone, Music Library, Pictures Library, Proximity, Removable Storage, Shared User Certificates, Videos Library, and Webcam. To the right of the list is a "Description" section for the selected "Appointments" capability. The description states: "Provides access to the user's appointment store. This gives read access to appointments obtained from the synced network accounts and to other apps that write to the appointment store, and mark those as available. With this capability your app can create new calendars and write appointments to calendars it creates." Below the description is a link labeled "More information".

Capabilities:	Description:
<input checked="" type="checkbox"/> Appointments	Provides access to the user's appointment store. This gives read access to appointments obtained from the synced network accounts and to other apps that write to the appointment store, and mark those as available. With this capability your app can create new calendars and write appointments to calendars it creates. <a href="#">More information</a>
<input type="checkbox"/> Contacts	
<input type="checkbox"/> Enterprise Authentication	
<input type="checkbox"/> Internet (Client & Server)	
<input type="checkbox"/> Location	
<input type="checkbox"/> Microphone	
<input type="checkbox"/> Music Library	
<input type="checkbox"/> Pictures Library	
<input type="checkbox"/> Proximity	
<input type="checkbox"/> Removable Storage	
<input type="checkbox"/> Shared User Certificates	
<input type="checkbox"/> Videos Library	
<input type="checkbox"/> Webcam	

Note: When testing apps using the Windows Phone emulator the capabilities are granted automatically, even when not included in the app manifest.

# Capabilities Overview



## Appointments

Allows an app to access the calendar and appointment info.

## Camera

Allows an app to access the built-in camera.

## Compass

Allows an app to access the built-in compass, if available.

## Contacts

Allows an app to access the contact info.

## Data services

Your phone's cellular data or Wi-Fi connection.

## Gyroscope

Allows an app to access the built-in gyroscope, if available.

## Location services

The approximate location.

## Libraries

Allows an app to access all photos, music, and videos on your phone.

## Microphone

Allows an app to record audio and to use Speech features.

## Movement sensor

Allows an app to access the motion sensor.

## Proximity

Allows access to the Bluetooth, Wi-Fi, and near field communication (NFC) capabilities.

## Owner identity

An anonymous identifier that allows an app to distinguish one person from another, but provides no personal info.

## Phone identity

A unique device identifier that allows an app to distinguish one phone from another.

## Push notification service

Notifications that an app automatically sends to your phone.

## Ringtones

Allows an app to access the ringtones.

## SD card

Allows an app access to the SD card.

## Speech recognition

Allows an app to access Speech features.

## Wallet

Allows an app to access items in your Wallet or to make payments.

## Web browser

Allows an app to access the web browser.

## Xbox

Allows an app to access the Xbox service or your account info.

# Capabilities – WhatsApp



social

## WhatsApp



Free

★★★★★  
29190 reviews

By installing you agree to the Terms of Use and  
other terms

App requires

- appointments
- contacts
- phone identity
- owner identity
- video and still capture
- location services
- maps
- music library
- photos library
- media playback
- microphone
- data services
- phone dialer
- push notification service
- movement and directional sensor
- VOIP calling
- web browser component
- HD720P (720x1280)
- WVGA (480x800)
- WXGA (768x1280)
- appointments
- Proximity
- SD card
- internet connection
- videos library
- photo, music, and video libraries
- camera

Locations all apps can access:

## Application install directory

- ◆ The folder where your app is installed on the user's system. (read only)

## Application data locations

- ◆ The folders where your app can store data. These folders (local, roaming and temporary) are created when your app is installed.

## Removable devices (SD Card)

- ◆ Access is limited to specific file types

## User's Downloads folder

Locations requiring additional capabilities in the app manifest:

## Libraries

- ◆ Documents
- ◆ Music
- ◆ Pictures
- ◆ Videos

Removable devices (SD Card)

Homegroup libraries

Media server devices (DLNA)

Universal Naming Convention (UNC) folders

# Agenda



## Introduction

## The Windows View

- ◆ Windows Environment
- ◆ Attack Surface
- ◆ Breaking Out

## The Mobile View

- ◆ Sandboxing & Encryption

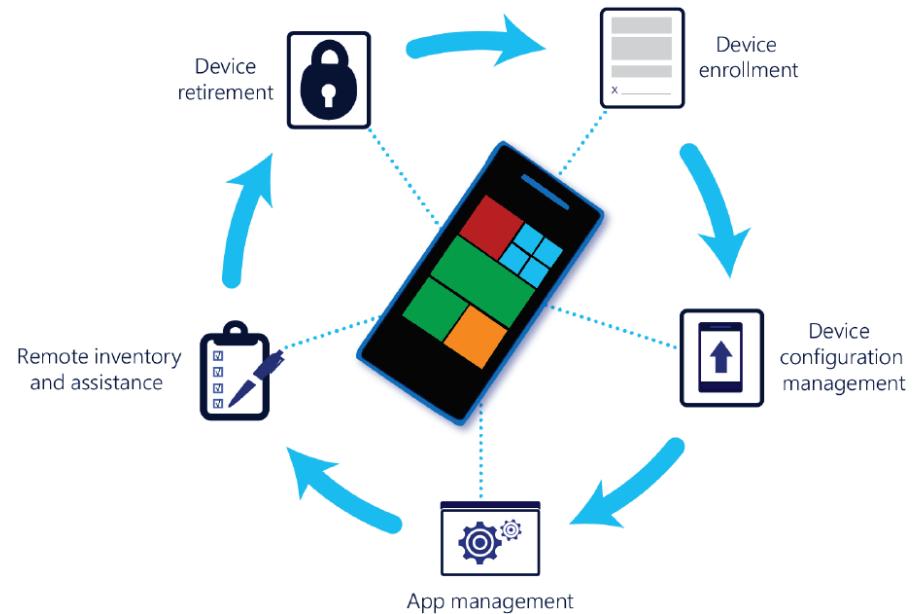
## Findings

- ◆ MDM Integration
- ◆ WiFi Sense
- ◆ Low Level Storage API

## Conclusion

Windows Phone 8.1 introduces push communication to distribute policies and configuration standards.

Periodically contacts the MDM server to download configurations, apps, updates and to upload asset information.



## Microsoft: Windows Phone 8.1 Mobile Device Management Overview

## Device Configuration Management

- ◆ Configuration Policies
- ◆ Access Management
- ◆ Storage Management
- ◆ Wi-Fi Network / VPN / Certificate Management
- ◆ Email Account / Message Management

## App Management

- ◆ Windows Phone Store Apps
- ◆ Sideloaded Apps
- ◆ Allow / Deny Apps

## Remote Inventory

- ◆ Remote Inventory / Assistance (Lock / PIN Reset)

## Device Retirement

# MDM Integration – Policies



Policies that MDM and EAS support	Policies that only MDM supports
Simple password	Disable cellular data roaming
Alphanumeric password	Disable Location
Minimum password length	Disable NFC
Minimum password complex characters	Disable Microsoft Account
Password expiration	Disable roaming between Windows devices
Password history	Disable custom email accounts
Device wipe threshold	Disable screen capture
Inactivity timeout	Disable copy & paste functionality
Device encryption	Disable share and save as
Disable removable storage card	App Allow/Deny list
Disable Camera	Disable Microsoft Store
Disable Bluetooth	Disable development unlock (side loading)
Disable Wi-Fi	Disable Internet Explorer
Disable Sync via USB	Disable Internet Sharing over Wi-Fi
	Disable Wi-Fi Off loading
	Disable Manual Configuration of Wi-Fi Profiles
	Disable Wi-Fi Hotspot reporting
	Disable VPN when Roaming over Cellular
	Disable VPN over Cellular
	Disable mdm un-enrollment and soft factory reset
	Disable Wi-Fi credential sharing
	Lock screen notification controls
	Disable telemetry data submission

## Microsoft: Windows Phone 8.1 Mobile Device Management Overview

Automatically connects you to Wi-Fi networks around you.

- ◆ Open Wi-Fi networks known by crowdsourcing e.g. other Windows Phone users have connected to.
- ◆ Accept the Terms of Use on your behalf.
- ◆ Provide additional information such as e-mail address or phone number on your behalf. (In some countries generic info will be used by default)
- ◆ Shares your Wi-Fi credentials with your Facebook friends, Outlook.com or Skype contacts.

Can I prevent my users from sharing their credentials?

- ◆ Yes, if you don't mind adding "\_optout" to your Wi-Fi SSID
- ◆ What about Google's "\_nomap" suffix then?

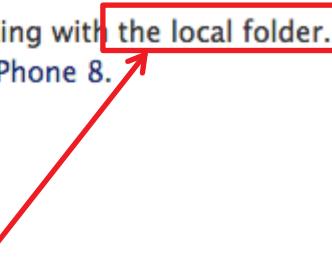
<https://www.windowsphone.com/en-us/how-to/wp8/connectivity/wi-fi-sense-faq>

## Win32 storage APIs supported on Windows Phone 8

Windows Phone 8 supports the following Win32 storage APIs for working with the local folder. For the full list of supported Win32 APIs, see [Supported Win32 APIs for Windows Phone 8](#).

- [CopyFile2](#)
- [CreateDirectoryW](#)
- [CreateFile2](#)
- [DeleteFileW](#)
- [FindClose](#)
- [FindFirstFileExW](#)
- [FindNextFileW](#)
- [FlushFileBuffers](#)

Local folder only?



# Demo



Device has to be registered / developer unlocked to deploy apps locally.

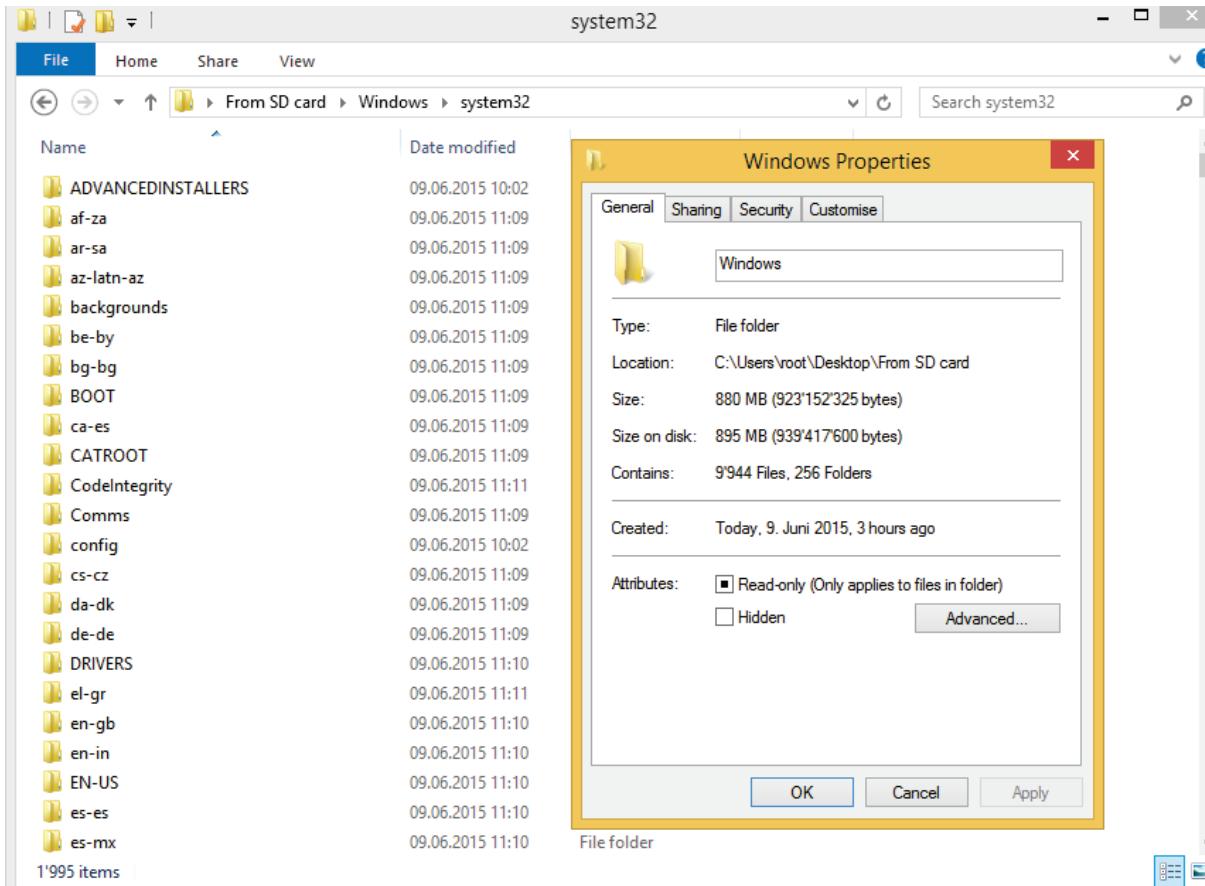
Our test app can now access files / pipes etc. outside of the «official» folders.

The app can also access documents stored by another app when knowing the path.

Does not work anymore when app is signed and distributed via Windows App Store.

We managed to brick our test phone and had to perform a full reset...

Extracted ~10'000 operating system files



Some documents seem to be encrypted ...



The screenshot shows a hex editor window titled "HxD - [C:\Users\root\Desktop\From SD card\Windows\Packages\RegistryFiles\Microsoft.Comms.rga]". The menu bar includes File, Edit, Search, View, Analysis, Extras, Window, and ?. The toolbar has icons for file operations and displays "16" and "hex" modes. The status bar at the bottom shows "Offset: 0", "Block: 0-9", "Length: A", and "Overwrite".

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	00000000 1F 8B 08 00 00 00 00 04 0B C5 56 6B 6B 13 41 .<.....AVkk.A	00000010 14 3D 9F 05 FF 43 E9 27 85 22 3E 36 BB 6A F0 43 .=Ý.ýCé'...">6»j8C	00000020 9A D4 B6 68 62 30 55 11 23 65 49 A3 86 DA 4D C8 šÔ¶hb0U.#eIftÜME	00000030 A6 0F FF BC 7A 66 B6 9B 7B C6 EC 88 62 41 96 36 .!y4zfq>{Æi`bA-6	00000040 33 73 1F 73 CE 9D 3B 73 EF 8F EF EF 30 43 81 13 3s.sÍ.;si.iiOC..	00000050 CC 71 89 12 5B 78 8D 29 3E 73 AD C4 0A 4B 7C E3 iq¾. [x.)>s.ä.K ä	00000060 CA 1E A5 33 CE E6 9C 6F E1 2D E5 4B 4A 67 9C 17 È.¥3íæmoá-åKJgœ.	00000070 9C B7 70 0F F7 F9 DD C6 2D 7C C0 01 5E 50 FF 3D œ·p_-ùÝE- À.^Pý=	00000080 8E F1 12 AF D0 45 87 BF C7 E8 F3 B7 4B D9 21 06 žñ. ÐE‡çéó-KÜ!.	00000090 94 8E 31 A2 ED 27 7A BC 44 4E 5F 53 AE F4 E9 6F "Žicí'z4DN_Sødéo	000000A0 C2 F1 9C 9E 2B D9 18 BF 22 1B E0 88 9A 5D 9C 53 Åñež+Ù.ç".à^š]œS	000000B0 CF 59 15 F4 10 A2 71 9E 2F E8 E7 C0 FB 59 E1 A3 IY.ð..cqž/èçÅÙYáf	000000C0 47 D5 A6 56 8E AF B4 9B E2 29 ED E7 38 E3 57 A2 GÖ;VZ" >å)iç8åWö	000000D0 C7 3D 9D 9F 09 FD 74 F8 BF 44 1B 43 AE 9D 11 E7 Ç=.Ý.ytøëD.C@..ç	000000E0 09 AE 88 29 E7 CC ED D2 C6 1B 4A 1D EF 1E D7 56 .@^)çliòE.J.i.xV	000000F0 FC 1B F9 D9 85 47 3D F5 F2 82 63 17 33 17 A5 E9 ü.ùÜ.G=öö,c.3.¥é	00000100 86 DC 76 ED 70 8F 1A 5F DB A3 71 3B E4 1E 45 49 tÜvip.._ÛEq;ä.EI	00000110 86 73 9C D2 BE B8 66 52 69 54 78 FB 5C 2F F9 E5 tseÖ%,fRITxû/ùå	00000120 3C 1F B7 83 63 DA 26 CA 92 F3 91 67 74 C4 BD 73 <..fcÜ&È'ó'gtÄms	00000130 DA 96 58 78 1C 0E 79 25 3F A4 7C 53 EA CE 6C 9B Ú-Xx..y%?x Séíl>	00000140 71 1E 4A 54 42 66 25 E5 CF F0 85 FB 5D E1 0E 32 q.JTBf%åïß..ù)å.2	00000150 DC 65 04 13 3C C2 8E 3F F3 1D A4 78 2E E3 5E E3 Üe..<ÅZ?ó..xx.ä^ä	00000160 38 13 FD 84 D6 66 FB 64 3D CE F0 70 3D 4E 99 53 8.ý.,Öfûd=fîðp=NMS	00000170 A6 63 7B 65 62 9B E0 41 44 C7 F4 5D 56 D6 7E 5A  c{eb}àADçô]VÖ~z	00000180 32 4E 61 FB A6 3C D1 5A 27 09 D6 OD E7 D8 67 51 2Naû;<ÑZ'.Ö.çøgQ
-----------	---	---	---	---	---	---	--	---	---	--	---	---	--	---	---	---	---	---	--	---	---	--	--	---	---	---

# Open Research

## Analysis of extracted information



The screenshot shows a Windows File Explorer window titled "Manifests" with the path "From SD card > Windows > system32 > Manifests". The left pane lists 122 XML files, and the right pane shows the contents of the selected file, "AccountsManagerPackMan.xml", in Notepad. The XML code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
Copyright (c) Microsoft Corporation. All rights reserved.
-->
<Provisioning xmlns="http://dbPreProvisioning"
RuntimeType="Native"
Title="@AccountSettingsRes.dll,-24001"
Version="1.0"
Genre=""
ProductID="{5B04B775-356B-4AA0-AAF8-6491FFEA5616}"
DisplayOnAppList="false"
SingleInstanceHost="true">
<IconPath xmlns="">res://UIXMobileAssets{ScreenResolution}!allapp.email.png</IconPath>
<ImagePath xmlns="">\Programs\CommsApplications\AccountsManager.exe</ImagePath>
<ImageParams xmlns=""></ImageParams>
<Tasks xmlns="">
    <DefaultTask Name="_default" ActivationPolicy="Replace"
        NavigationPage="res://AccountSettings!AccountsPage.uix#AccountsPage"/>
    <ExtendedTask Name="USW" ActivationPolicy="Replace"/>
    <ExtendedTask Name="CreateAccount" ActivationPolicy="Replace"/>
    <ExtendedTask Name="ShowSettings" ActivationPolicy="Replace"/>
</Tasks>
<Tokens xmlns="">
    <PrimaryToken TokenID="Default"
```

While others are not ...

# Agenda



## Introduction

## The Windows View

- ◆ Windows Environment
- ◆ Attack Surface
- ◆ Breaking Out

## The Mobile View

- ◆ Sandboxing & Encryption

## Findings

- ◆ MDM Integration
- ◆ WiFi Sense
- ◆ Low Level Storage API

## Conclusion

## Windows 8.1 versus Windows 10

- ◆ Windows 10 released end of July this year  
(but probably not with for the phone)
- ◆ New version called Windows 10 Mobile Enterprise
- ◆ Private versus business containers, providing isolation between contexts  
(probably using Microsoft Advanced Threat Analytics – ATA)

## Windows Phone 8.1

- ◆ Is more a phone similar to iOS and Android than a Windows desktop
- ◆ Is based on secure and proven good security technologies
- ◆ Is a first step into a more mature Windows 10 eco-system
- ◆ Is as business ready as your current MDM solution is...

Questions?

