# Windows Phone Security Model and Platform

## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Phone Platform</td>
<td>2</td>
</tr>
<tr>
<td>Windows Phone Security Model -1</td>
<td>5</td>
</tr>
<tr>
<td>Windows Phone Security Model -2</td>
<td>6</td>
</tr>
<tr>
<td>Windows Phone Security Model -3</td>
<td>8</td>
</tr>
<tr>
<td>Windows Phone Security Model -4</td>
<td>9</td>
</tr>
<tr>
<td>Windows Phone Security Model -5</td>
<td>11</td>
</tr>
<tr>
<td>Windows Phone Security Model -6</td>
<td>14</td>
</tr>
<tr>
<td>Windows Phone Security Model -7</td>
<td>16</td>
</tr>
<tr>
<td>Windows Phone Security Platform</td>
<td>18</td>
</tr>
<tr>
<td>Notices</td>
<td>19</td>
</tr>
</tbody>
</table>
Windows Phone Platform

Successor to Windows Mobile

Part of Windows CE family

Primarily aimed at consumer market

- Enterprise support limited to only Exchange and ActiveSync
- Enterprise management of third-party apps is not supported

Heavy support for cloud-based services

**091 Windows Phone Platform. It's a successor to Windows Mobile. The last Windows Mobile update was 6.5.something. And then they decided we're getting too far behind our competitors. We need to come out with a new platform.

So Windows Phone 7 was released. And Windows Phone 7 was an entire rewrite of how their devices were actually going to work.

It is still part of the Windows CE family. Going forward that might not be the case. With Windows 8--normal Windows 8, so desktop, which is coming out in 2012--the rumors
are saying that the phones might move over to the same code base that's being used by the desktop and the tablets.

Microsoft has not confirmed whether that's the case or not. It wouldn't surprise me if they are. They do like - the smaller number of code bases they have to create for, the more commonality between the different code bases, the easier it is to make applications; the easier it is to secure all of the devices running it.

But we'll find out probably later this summer. The release candidate is supposed to come out in the summer of 2012 that says what are the phones going to be running going forward.

Primarily aimed at the consumer market. There are very few enterprise applicable controls on these devices; and we'll discuss which ones are available going forward.

Exchange and ActiveSync. However ActiveSync, it supports somewhere between 10 and 20 of the 100 ActiveSync policies. So it's a very limited subset of ActiveSync policies that it supports.

Again, there's a very good Wikipedia article on ActiveSync that shows exactly which version of phone you have, which ActiveSync policies it supports.
Again, Wikipedia can be a great source; but you never can tell whether it’s true or not. So you want to do additional research before you take what Wikipedia has as gospel.

Enterprise management of third-party apps is not supported. So you cannot control the user installing third-party apps on the phone. Google, Apple, both allow you to disallow other apps being installed. You just can’t do it with Windows Phone 7 or 7.5 at this point.

There is very heavy support for cloud-based services; just like with the other companies. Google has their cloud services, Apple has their cloud services, Windows--Microsoft has their cloud services; and there is very tight integration with their cloud-based services.
Windows Phone Security Model -1

Designed for the protection of confidential data and communications

Security Architecture is divided into four sections:

1. Chambers
2. Capabilities
3. Sandbox
4. Application Deployment

**092 The phone itself is designed to protect confidential data and communication, to a point. The security is nowhere near the amount of security that a BlackBerry device connected to a BEZ has.

We’re not talking about full communication path encryption between the phone and the Exchange Server. It doesn’t get to that level. But just like with any phone, with any mobile device, there is some security inherent in the product.

The security architecture is divided into four main sections: Chambers, Capabilities, Sandbox and Application
Deployment. Using these four sections is how they maintain application security on their devices.

**Windows Phone Security Model -2**

**Windows Phone Security Model -2**

Chambers

- Windows Phone processes and applications execute in isolated chambers that are each configured with their own security policy.
- Four chamber types:
  1. Trusted Computer Base (TCB) Chamber
     - Configured with the greatest set of privileges; allows processes to have unrestricted access to system resources.
     - Ex: Kernel, Windows Phone drivers
  2. Elevated Rights Chamber (ERC)
     - Processes have access to all system resources except the security policy.
     - Ex: phone application services, user mode drivers

**093 So Chambers. Windows Phone processes and applications execute in isolated chambers that are configured for their own security policy.

And they have four separate chamber types. They have a Trusted Computer Base Chamber. These have the greatest set of privileges. These-- if the app is loaded into a TCB chamber, it has the widest range of privileges. It allows processes to have unrestricted access to system resources.
It says here "Kernel, Windows Phone drivers." Microsoft severely limits applications having the ability to be put into a Trusted Computer Base Chamber.

If an app developer submits something like that, Windows is going to turn around, or Microsoft is going to turn and say, "You probably need to develop this app to work in a lower chamber. We don’t want to give you this much access to our kernel."

Elevated Rights Chamber. Processes have access to all system resources except the security policy. Again, we’re talking about stuff like drivers and services. Most apps, other than apps Microsoft develops for a certain purpose for the phone, will not have any access to using this chamber.
Chambers (con’t)

- Four chamber types:
  3. Standard Rights Chamber (SRC)
     - For processes and apps that do not provide a “device-wide” service; default chamber for pre-installed Windows Phone applications.
     - Ex: Microsoft Outlook Mobile 2010
  4. Least Privileged Chamber (LPC)
     - Security policy is defined based on stated capabilities of third-party apps installed from the Windows Phone Marketplace.
     - Ex: Facebook, Twitter

**094 Standard Rights Chamber. This is where most apps are going to run from. Stuff like Microsoft Outlook Mobile-- it says up there-- antivirus programs. These things need more permission than a Least Privileged Chamber but they don’t need as much-- they don’t need full access to the whole operating system files.

Least Privileged. This is where things like games are going to be installed; Facebook, Twitter. They don’t need very many permissions in order to run. They need explicit permissions, and that’s it. And this is the level most apps will be written at.
**095** Capabilities. Windows Phone resources used by apps that need to be controlled via dynamic access control policies. Capabilities are stuff like the GPS on the device; the camera on the device; the microphone on the device; access to Wi-Fi; access to Bluetooth.

A capability-- basically when you write a program you say, "My program needs to be able to--" And then you have like checkboxes. "I need to be able to do camera, GPS and this." You're going to have to tell Microsoft why you want to have these capabilities if your application doesn't seem you'll need it.
Why does Angry Birds need access to the camera? And there might be a valid reason. Maybe they're adding a feature in Angry Birds where they take your picture when you reach a high score. I don't know.

But during the app review process again, Microsoft will go and say, "Why do you need the capabilities you've said- or you've listed in order for your program to run?"

Explicitly stated by the application at installation time; and cannot be changed during run-time. So again we're going back to the Android model.

When I first install an app on Android, it tells me exactly which items it needs to be able to access. Well Windows does the same thing. And once it's installed, unless there's an update that I install, or that my phone updates automatically because I've allowed it to, if a permission changes I have to be notified as a user that the permission level changed.

So that's Microsoft's way of preventing a bait and switch where you install an app and then five days later the app comes out and suddenly it needs access to GPS, Wi-Fi and Bluetooth when it didn't before. The user still has to sign off on it.

Again, ultimately it's the user's responsibility to make sure that they're securing their phone. We as users own it. We're the primary
people that are installing stuff on the phone. It falls on us to make sure we're not doing something stupid with the phone.

**Windows Phone Security Model -5**

**Windows Phone Security Model -5**

**Sandbox**

- Each Windows Phone app runs in its own isolated chamber configured with access to stated capabilities.
- Each app is granted access to its own isolated storage container.
- There are no methods to communicate between running apps on a Windows Phone except via the cloud.
- Third-party apps cannot remain active in the background.
  - Gracefully shutdown when user switches to another app
  - Prevents potentially unwanted access to resources

---

**CERT | Software Engineering Institute | Carnegie Mellon**

096 So along with the chambers which severely limit the permissions, the app also gets sandboxed while it's running. The sandboxing works almost exactly the same as the other operating systems we've talked about.

It gets placed in a sandbox in a virtual machine where it has its own access to memory. It's not-- it doesn't have any access to the system kernel; doesn't have access to
other programs. If this app fails for some reason, the app will crash, it should not affect the usage of the rest of the apps on the phone; it should not crash the whole system if the app crashes.

Also it should not be able to talk to other apps and get data from other apps. It should not be able to go to your email and be able to send your email to somebody else. It shouldn't be able to look at your call history and send your call history to somebody else. That's the whole purpose of sandboxing. We want to keep everything in its own separate box so we can control what it can and cannot do.

There are no methods to communicate between running apps on a Windows Phone except via the cloud.

So on Android, if you're the developer you could actually sign the app to be able to communicate with each other. So you could have one app that was able to access data from the other app if you developed both applications.

Windows, there's no way to do that except via the cloud. So if for whatever reason your app is storing stuff in the cloud, a secondary application might be able to access that same data that's up in the cloud.

Again, the security concerns with the cloud. Once it's up in the cloud on somebody else's server, I as a user
have no idea how well protected it is in the cloud environment. They could be doing anything with that data and I would have no idea that something potentially malicious was going on with the data that’s been uploaded to the cloud.

Third-party apps cannot remain active in the background. They have limited multitasking. So when I switch to another app, the application I was in gracefully shuts down; and this prevents potentially unwanted access to the resources.

So unlike Google or Android, it allows multiple apps to be running—third-party apps or first-party apps to be running. On the Windows Phone currently only one app, only one third-party app is running at a time.

Now that doesn’t mean that first-party apps such as the Zoon Music Player or other Microsoft apps aren’t running. It just means that my bank app and my Facebook app are not running concurrently; it’s only running one at the same time— one, it’s only running one at a time, since it can’t run them at the same time.
**Windows Phone Security Model -6**

**Application Deployment**

- All Windows Phone developers must be actively registered with Microsoft before an app can be submitted for review.
- All applications are code-signed by VeriSign on behalf of the developer.
  - Apps that are not code-signed will not run on Windows Phone.
- Windows Phone apps are only available from the Windows Phone Marketplace.

**Removable Storage**

- Windows Phone 7 does not support removable storage of any kind.

**097 Application Deployment. All Windows Phone developers must be registered with Microsoft before it can be submitted for review. So if I'm developing apps for the Windows Phone, Microsoft has to know I'm creating them; because I have to register with them first.**

All applications are code-signed by VeriSign on behalf of the developer. If it's not signed, it's not going to run. There's no way around it.

And as far as I know, I haven't seen any way to root a Windows Phone, at this point, to run unsigned code. Probably because the user base is
much more limited than Android and Google; so-- I mean Android and Apple, iOS. So people just haven't targeted to try to find out if there's a way to root it.

Windows Phone apps are only available currently from the Windows Phone Marketplace. That being said, Windows- or Microsoft's model is they want enterprises to be able to have their own app store. So if I'm a Dell employee, and Dell is a Windows- has Windows Phones as part of their network environment, Dell can have their own application store where users can go to that store and download applications to run on the phone.

And it does not support any type of removable storage. So you don't even have to worry about removable storage encryption because it's all internal memory. You don't have to worry about a removable media card.
Windows Phone Security Model -7

Device Passwords

- Alphanumeric passwords are not supported.
- Only weak passwords can be defined.

Encryption

- Device encryption is not supported.

Now we start getting into one of the major issues of the Windows Phone. Alphanumeric passwords are not supported. Only weak passwords can be defined; and by weak passwords I'm talking about a four-digit numeric pass code. No support for alphanumeric whatsoever; can't do more than four characters.

Now with Windows Phone 8, again from what I've read, they're going to actually go with a BitLocker implementation for the encryption-for encrypting the back side. If they go with BitLocker, they're going to have to have complex passwords.
So going forward hopefully we'll have it. But the question is, will older phones be upgradeable to Windows 8, with all functionality?

BitLocker usually requires a special chip on the device it's running on. So I don't know whether older phones are going to be able to run it or not; because they probably do not have the required chip on the device. But we'll see.

Currently no encryption is supported. So if I get a Windows Phone 7, the back end is not encrypted. So if I can get to the storage space, and I know how to read the storage space, I can get everything from the storage space in plain text.

It's just like a drive in most computers. If I take a drive out of 90 percent of the computers out there, the user probably wasn't using any sort of encryption, and I'll be able to read it with other forensic tools.

So these are two things that make Windows Phone not very desirable at the enterprise - for the enterprise user. Can't secure it with strong passwords; can't encrypt the back end if it goes missing. This makes it a no-go for most organizations.
Windows Phone Security Platform

Authentication
- Exchange Server to enforce password policies
- Uses digital certificates for network authentication

Data Protection
- Allows file encryption using vetted third-party apps

Application Security
- Limited ability to verify individual applications

**099 Authentication. The Exchange Server to enforce password policy; and basically the only password is you have to have the four-digit numeric pass code because there's no complex pass codes. But you can use digital certificates for network authentication.

Again, with the weakness to passwords and encryptions, the encryption-- the non-existence of encryption I should say-- if I'm a CEO or a CIO, I'm not going to allow these devices on my network. So it's great that they can use digital certificates. But I wouldn't let it within 100 feet of my network. So--
but that's my personal opinion. I'm sure there's CIOs out there that have; and if you're listening, I apologize.

Data Protection. Allow file encryption using vetted third-party apps. So there are third-party apps you can use to encrypt specific files on the phone. It just doesn't encrypt the whole back end.

And Application Security. It's limited ability to verify individual applications. So it does do some code signing review to see- to verify that this application is what it says it is.

**Notices**

Copyright 2013 Carnegie Mellon University

This material has been approved for public release and unlimited distribution except as restricted below.

This material is distributed by the Software Engineering Institute (SEI) only to course attendees for their own individual study. Except for the U.S. government purposes described below, this material SHALL NOT be reproduced or used in any other manner without requesting formal permission from the Software Engineering Institute at permission@sei.cmu.edu.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose this material are restricted by the Rights in Technical Data-Noncommercial Items clauses (DFAR 252-227.7013 and DFAR 252-227.7013 Alternate I) contained in the above identified contract. Any reproduction of this material or portions thereof marked with this legend must also reproduce the disclaimers contained on this slide.

Although the rights granted by contract do not require course attendance to use this material for U.S. Government purposes, the SEI recommends attendance to ensure proper understanding.

NO WARRANTY. THE MATERIAL IS PROVIDED ON AN “AS IS” BASIS, AND CARNEGIE MELLON DISCLAIMS ANY AND ALL WARRANTIES, IMPLIED OR OTHERWISE (INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, RESULTS OBTAINED FROM USE OF THE MATERIAL, MERCHANTABILITY, AND/OR NON-INFRINGEMENT).

CERT® is a registered mark of Carnegie Mellon University.