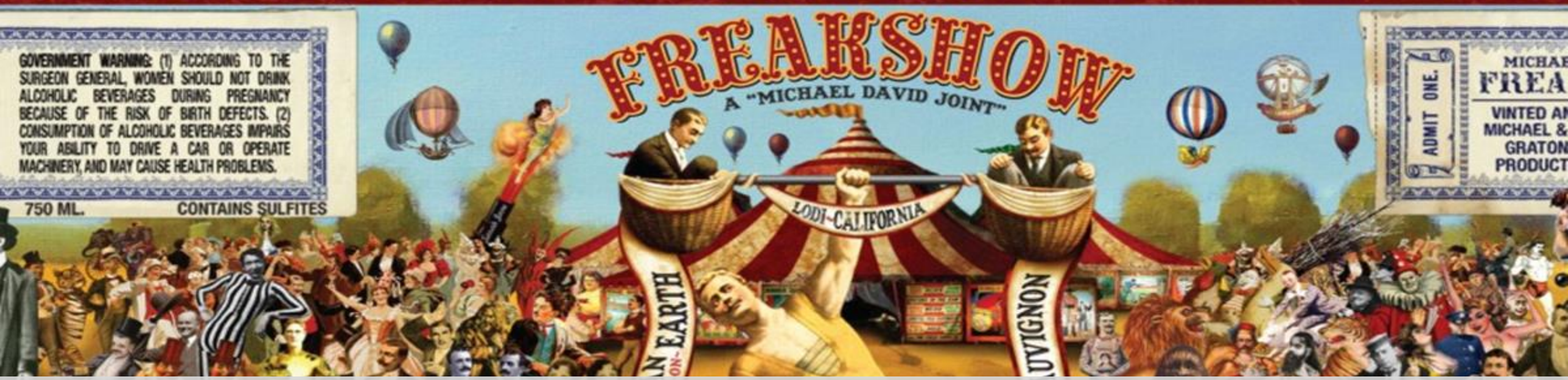
THE WINDOWS PHONE FREAKSHOW

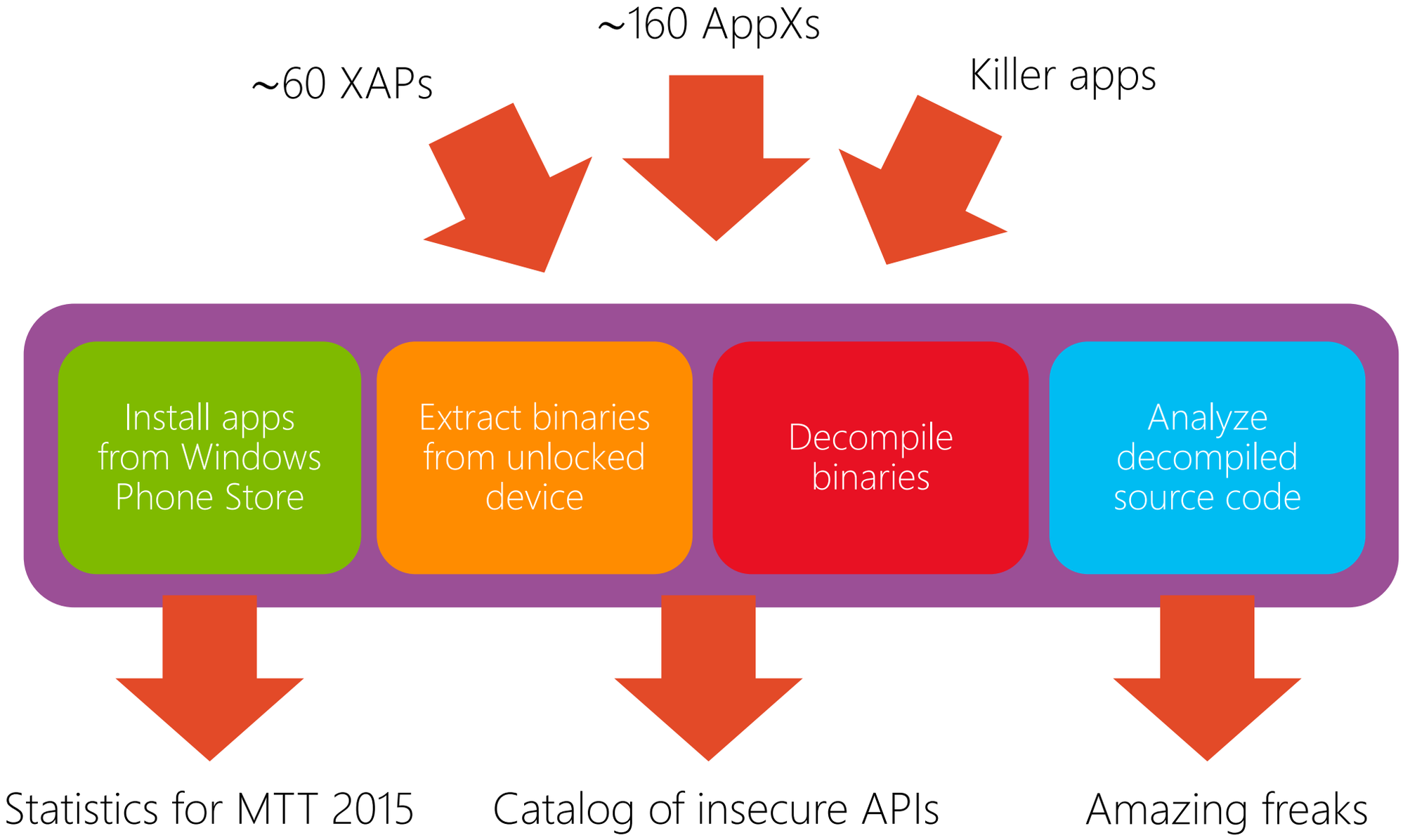
Luca De Fulgentis ~ luca@securenetwork.it

Hack in The Box, Amsterdam ~ 05/29/2015

Luca De Fulgentis
Offensive Security Adept
Chief Technology Officer at Secure Network
Consuming brain power with InfoSec since 2001



INTRODUCING THE FREAK SHOW



freaks*

in terms of vulnerable examples of code

* no *toy code*, real world examples only

THE FIL ROUGE

SELF DEFENDING APPS

"A lack of binary protections results in a mobile app that can be analyzed, reverse-engineered, and modified by an adversary in rapid fashion" [1]

[1] https://www.owasp.org/index.php/Mobile_Top_10_2014-M10

securenetwork
Your Protection, Our Mission

lack of anti-debugging mechanisms
and runtime-tampering detection

*"A lack of binary protections results in a mobile app that can be **analyzed**, reverse-engineered, and modified by an adversary in rapid fashion"* [1]

[1] https://www.owasp.org/index.php/Mobile_Top_10_2014-M10

securenetwork
Your Protection, Our Mission

lack of anti-debugging mechanisms and runtime-tampering detection

lack of code obfuscation and code encryption

"A lack of binary protections results in a mobile app that can be analyzed, reverse-engineered, and modified by an adversary in rapid fashion" [1]

[1] https://www.owasp.org/index.php/Mobile_Top_10_2014-M10

securenetwork
Your Protection, Our Mission

lack of anti-debugging mechanisms
and runtime-tampering detection

lack of code obfuscation
and code encryption

"A lack of binary protections results in a mobile app that can be analyzed, reverse-engineered, and modified by an adversary in rapid fashion" [1]

lack of resources integrity verification
and jailbreak detection mechanisms

[1] https://www.owasp.org/index.php/Mobile_Top_10_2014-M10

securenetwork
Your Protection, Our Mission

intellectual property theft

lack of anti-debugging mechanisms
and runtime-tampering detection

lack of code obfuscation
and code encryption

"A lack of binary protections results in a mobile app that can be analyzed, reverse-engineered, and modified by an adversary in rapid fashion" [1]

lack of resources integrity verification
and jailbreak detection mechanisms

[1] https://www.owasp.org/index.php/Mobile_Top_10_2014-M10

securenetwork
Your Protection, Our Mission

intellectual property theft

lack of anti-debugging mechanisms
and runtime-tampering detection

lack of code obfuscation
and code encryption

"A lack of binary protections results in a mobile app that can be analyzed, reverse-engineered, and modified by an adversary in rapid fashion" [1]

malicious app *clones*
app cracking
frauds

lack of resources integrity verification
and jailbreak detection mechanisms

[1] https://www.owasp.org/index.php/Mobile_Top_10_2014-M10

securenetwork
Your Protection, Our Mission

~95%* of analyzed apps

lack proper binary protections

* 223 out of 235 assessed apps

securenetwork
Your Protection, Our Mission

Weak custom code encryption

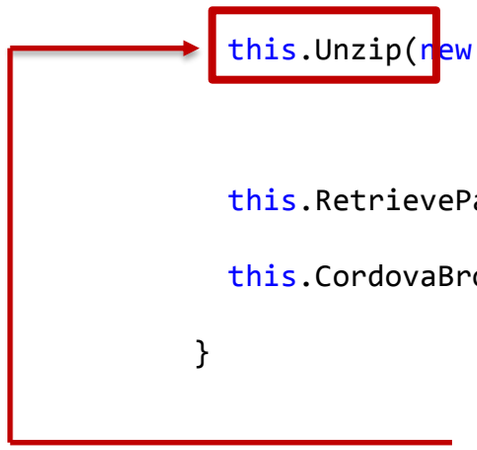
```
private void CordovaBrowser_Loaded(object sender, RoutedEventArgs e)
{
  this.resourceStreamInformation =
  Application.GetResourceStream(new Uri(Resource1.WWWPath, UriKind.Relative));

  // [...]
  string encodedPassword = Resource1.EncodedPassword;

  this.strPasswordDecodingSecond =
    CordovaView.Base64Decode(CordovaView.Base64Decode(
      encodedPassword.Substring(0, encodedPassword.Length - 10)));

  this.passwordLength = this.strPasswordDecodingSecond.Length;
  this.stream = this.resourceStreamInformation.Stream;
  this.filebytes = Convert.FromBase64String(CordovaView.StreamToString(this.stream));

  this.Unzip(new MemoryStream(this.Decrypt(this.filebytes, this.strPasswordDecodingSecond, this.passwordLength)));


  this.RetrievePage(); // CordovaView.uri setting

  this.CordovaBrowser.Navigate(CordovaView.uri); // Navigate unzipped app index.html page

}
```

pathname of the encrypted ZIP file

hardcoded password

Weak custom code encryption

```
private void CordovaBrowser_Loaded(object sender, RoutedEventArgs e)
{
  this.resourceStreamInformation =
  Application.GetResourceStream(new Uri(Resource1.WWWPath, UriKind.Relative));

  // [...]
  string encodedPassword = Resource1.EncodedPassword;

  this.strPasswordDecodingSecond =
    CordovaView.Base64Decode(CordovaView.Base64Decode(
      encodedPassword.Substring(0, encodedPassword.Length - 10)));

  this.passwordLength = this.strPasswordDecodingSecond.Length;
  this.stream = this.resourceStreamInformation.Stream;
  this.filebytes = Convert.FromBase64String(CordovaView.StreamToString(this.stream));

  this.Unzip(new MemoryStream(this.Decrypt(this.filebytes, this.strPasswordDecodingSecond, this.passwordLength)));



  this.RetrievePage(); // CordovaView.uri setting

  this.CordovaBrowser.Navigate(CordovaView.uri); // Navigate unzipped app index.html page

}
```



Unizip() calls the UnzipAndSaveFiles() method

securenetwork
Your Protection, Our Mission

Weak custom code encryption

```
public void UnzipAndSaveFiles(Stream stream)
{
     // [...]

        using (ZipInputStream zipInputStream = new ZipInputStream(stream))
        {
          storeForApplication.CreateDirectory(Resource1.WWWDirectory);

          ZipEntry nextEntry;
          while ((nextEntry = zipInputStream.GetNextEntry()) != null)
          {
           // [...]
                    str1 = Path.Combine(Resource1.WWWDirectory, strArray[index]);
                    if (!storeForApplication.DirectoryExists(str1))
                      storeForApplication.CreateDirectory(str1);
```

unzipped file content is saved in the SANDBOX

On apps encryption

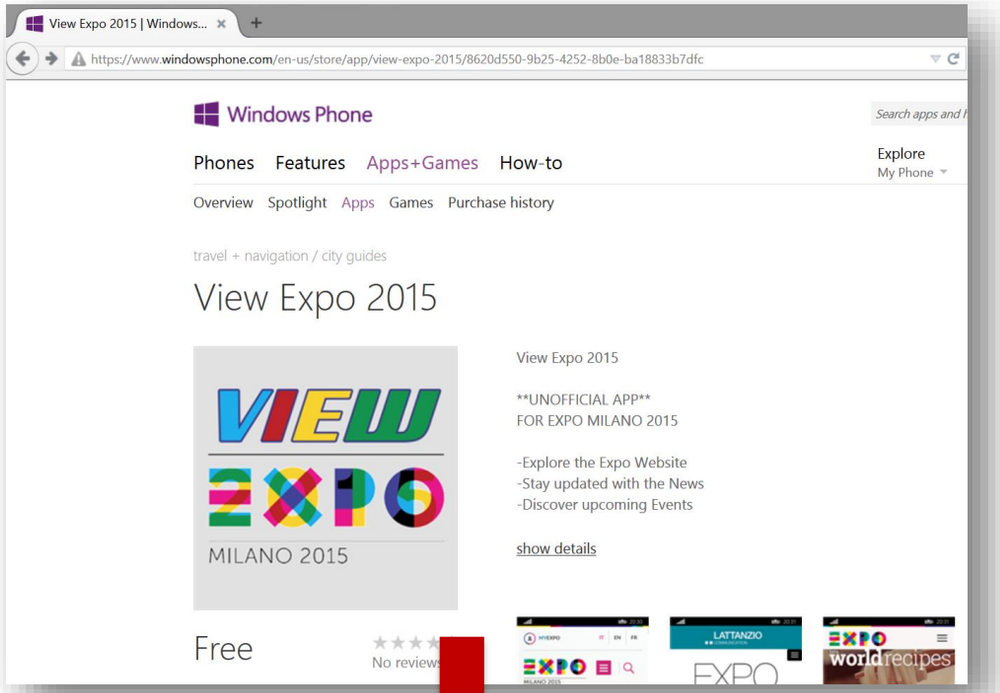
- Windows Phone Store apps are downloaded as encrypted files
  - Packages are then decrypted during the installation phase
- A privileged access to the file system allows binaries extraction
  - Apps' bytecode can be easily decompiled with publicly available utilities
    - ILSpy, .NET Reflector and JetBrains dotPeek are examples of available decompilers
- Code obfuscation and encryption represent solid strategies to mitigate
  - Intellectual theft
  - App behavior analysis while increasing malicious users effort

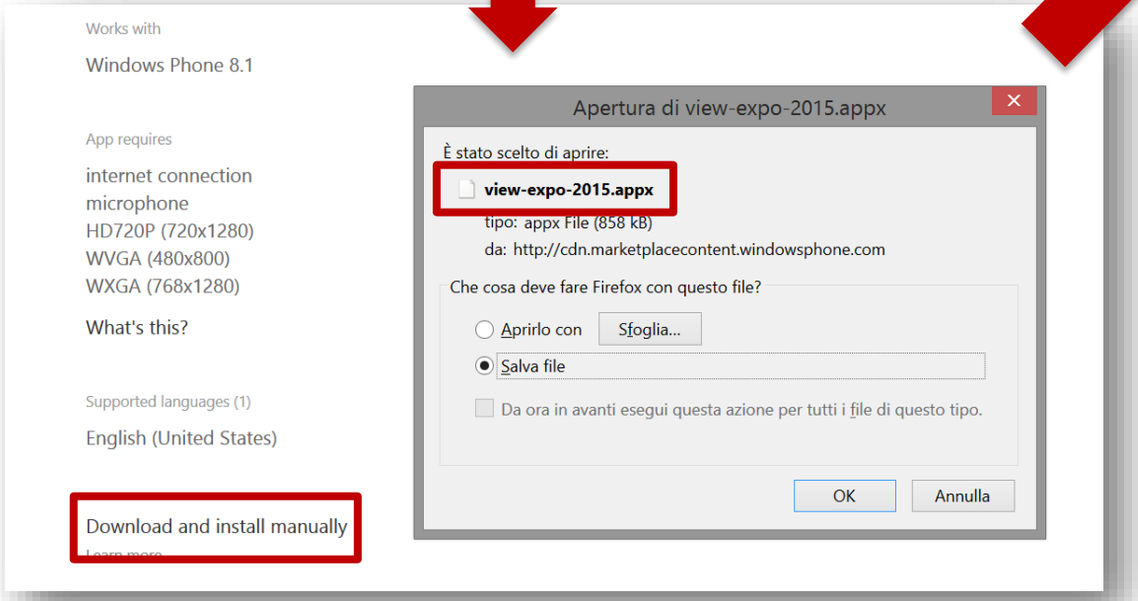
Comparison by feature by package format

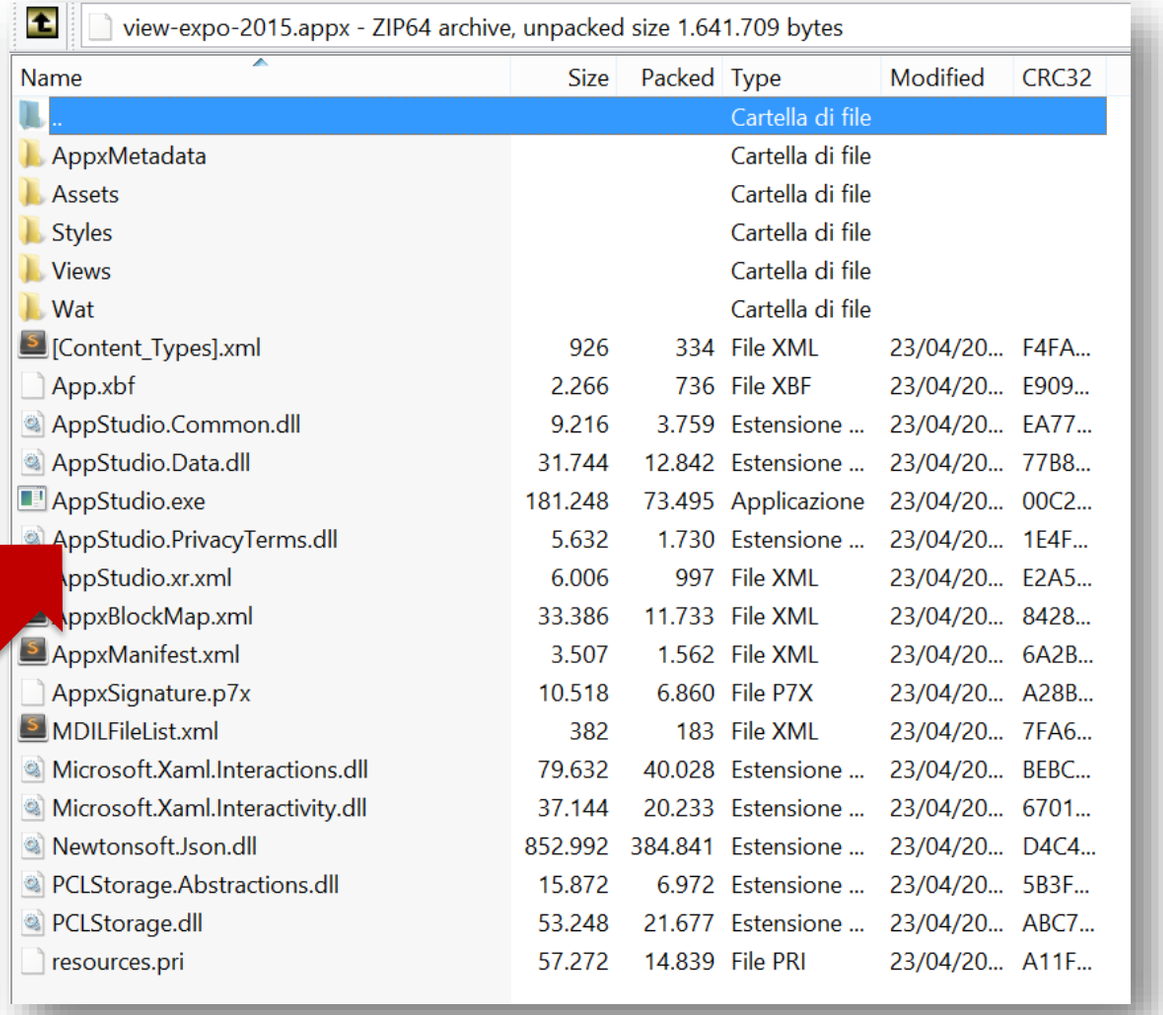
In summary...

Feature	XAP Phone	XAP 8.1 Phone	AppX Phone	AppX Windows
Platform Targeting	7.x and later	8.1 and later	8.1 and later	8.0 and later
Package Encryption	Yes	Yes	No, not yet.	No, not yet.
Package Bundling	No	No	Yes	Yes
Debug Package Signing	No	No	No	Yes
Differential Download/Update	No	No	Yes	Yes
Application File Single Instancing	No	No	Yes	Yes
Formal Versioning Requirements	No	Yes	Yes	Yes
External Volume (SD) App Installation	Yes on 8.1	Yes	Yes	No, not yet.

* Slide taken from a Microsoft's Build presentation







Name	Size	Packed	Type	Modified	CRC32
..			Cartella di file		
AppxMetadata			Cartella di file		
Assets			Cartella di file		
Styles			Cartella di file		
Views			Cartella di file		
Wat			Cartella di file		
[Content_Types].xml	926	334	File XML	23/04/20...	F4FA...
App.xbf	2.266	736	File XBF	23/04/20...	E909...
AppStudio.Common.dll	9.216	3.759	Estensione ...	23/04/20...	EA77...
AppStudio.Data.dll	31.744	12.842	Estensione ...	23/04/20...	77B8...
AppStudio.exe	181.248	73.495	Applicazione	23/04/20...	00C2...
AppStudio.PrivacyTerms.dll	5.632	1.730	Estensione ...	23/04/20...	1E4F...
ppStudio.xr.xml	6.006	997	File XML	23/04/20...	E2A5...
ppxBlockMap.xml	33.386	11.733	File XML	23/04/20...	8428...
AppxManifest.xml	3.507	1.562	File XML	23/04/20...	6A2B...
AppxSignature.p7x	10.518	6.860	File P7X	23/04/20...	A28B...
MDILFileList.xml	382	183	File XML	23/04/20...	7FA6...
Microsoft.Xaml.Interactions.dll	79.632	40.028	Estensione ...	23/04/20...	BEBC...
Microsoft.Xaml.Interactivity.dll	37.144	20.233	Estensione ...	23/04/20...	6701...
Newtonsoft.Json.dll	852.992	384.841	Estensione ...	23/04/20...	D4C4...
PCLStorage.Abstractions.dll	15.872	6.972	Estensione ...	23/04/20...	5B3F...
PCLStorage.dll	53.248	21.677	Estensione ...	23/04/20...	ABC7...
resources.pri	57.272	14.839	File PRI	23/04/20...	A11F...

securenetwork
Your Protection, Our Mission

Secure mindset

- Apps should be securely *designed* to mitigate binary attacks
  - OWASP RE and Code Modification Prevention Project provides secure design principles
- Adopt tools such as **dotFuscator** and **ConfuserEx** to protect binaries
- *Certificate pinning* should be implemented as well – see later
- **Binary protections simply mitigate, but *do not solve*, binary attacks**
  - They represent a further layer of security (obscurity?)
  - Consider that every protection can be bypassed with proper time and motivation
  - The idea is *raising the bar* to increase attacker's effort

DATA TRANSPORT SECURITY

Transport security

- Confidentiality in the app-to-backend communication
  - http-based communication ⊂ WP-supported mechanisms
- Common issues
  - Communication over an unencrypted channel – e.g., http instead of https → MiTM attacks
  - Communication over a poorly encrypted channel – e.g., use of weak encryption mechanisms
  - Issues related to digital certificates handling

Hunting for transport issues

Category	Namespaces	Classes, Methods or Properties	
Http	System.Net.Http.HttpClient	DeleteAsync() GetAsync() PostAsync() PutAsync()	GetByteArrayAsync() GetStreamAsync() GetStringAsync() SendAsync()
	Windows.Web.Http.HttpClient	DeleteAsync() GetAsync() PostAsync() PutAsync()	GetStringAsync() SendRequestAsync() GetBufferAsyn() GetInputStreamAsync()
TCP and UDP Sockets	Windows.Networking.Sockets	StreamSocket.ConnectAsync() SocketProtectionLevel.PlainSocket - property StreamSocket.UpgradeToSslAsync() StreamSocketListener - does not support SSL/TLS DatagramSocket.ConnectAsync()	

Hunting for transport issues

Category	Namespaces	Classes, Methods or Properties
Web	Microsoft.Phone.Controls	WebBrowser.Navigate() WebBrowser.Source property
	Windows.UI.Xaml.Controls	WebView.Navigate() WebView.Source property
	Microsoft.Phone.Tasks	WebBrowserTask.Uri property
	Windows.System	Launcher.LaunchUriAsync(uri)
WebSocket	Windows.Networking.Sockets	MessageWebSocket.ConnectAsync() – with ws:// uri scheme StreamWebSocket.ConnectAsync() – with ws:// uri scheme

Hunting for transport issues

Category	Namespaces	Classes, Methods or Properties
XAML Object Element Usage	-	«Source» property for WebBrowser and WebView «uri» property for WebBrowserTask "NavigateUri" for HyperlinkButton
Push Notifications	Microsoft.Phone.Notification	HttpNotificationChannel(string)
Brutal approach	-	Grep for Uri() and look at http:// instead of https://
Digital Certificates	Windows.Web.Http.Filters	HttpBaseProtocolFilter.IgnorableServerCertificateErrors.Add()
Windows.Web.AtomPub, Windows.Networking.BackgroundTransfer, Windows.Web.Syndication classes/methods should be reviewed as well		

Video

Attacking unencrypted communication – take I
Hijacking CNN's app news

securenetwork
Your Protection, Our Mission

Video

Attacking unencrypted communication – take II
Stealing Instagram app authorization token

Phishing has never been so easy

```
<phone:PhoneApplicationPage
  x:Class="App.MainPage"
     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
     xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
     xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
     mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
     FontFamily="{StaticResource PhoneFontFamilyNormal}"
     FontSize="{StaticResource PhoneFontSizeNormal}"
     Foreground="{StaticResource PhoneForegroundBrush}"
     SupportedOrientations="PortraitOrLandscape" Orientation="Portrait"
     shell:SystemTray.IsVisible="True">

    <phone:WebBrowser Height="Auto" IsScriptEnabled="true" Source="http://m.WONT-SAY.com/login1.html?continua=true"
      HorizontalAlignment="Stretch" Name="WONT-SAY"
      VerticalAlignment="Stretch" Width="Auto"
      Margin="-12,0,0,0" Grid.ColumnSpan="2" />

</phone:PhoneApplicationPage>
```

an attacker can replace
the login page with a malicious one

securenetwork
Your Protection, Our Mission

Phishing has never been so easy

```
public CordovaView()
{
      this.InitializeComponent();
      if (DesignerProperties.IsInDesignTool)
        return;

      // [...]
      if (this.configHandler.ContentSrc != null)
       this.StartPageUri = !Uri.IsWellFormedUriString(this.configHandler.ContentSrc, UriKind.Absolute) ?
          new Uri(CordovaView.AppRoot + "www/" + this.configHandler.ContentSrc, UriKind.Relative) :
          new Uri(this.configHandler.ContentSrc, UriKind.Absolute);
     // [...]
```

```
<!-- ... -->

<link rel="stylesheet" href="style.css" />
<link rel="stylesheet" href="style-icons.css" />

<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?v=3.4&key=ABCDE[...]&libraries=places"></script>

<!-- ... -->
```

www/index.html loads a remote JS via http == XSS

Video

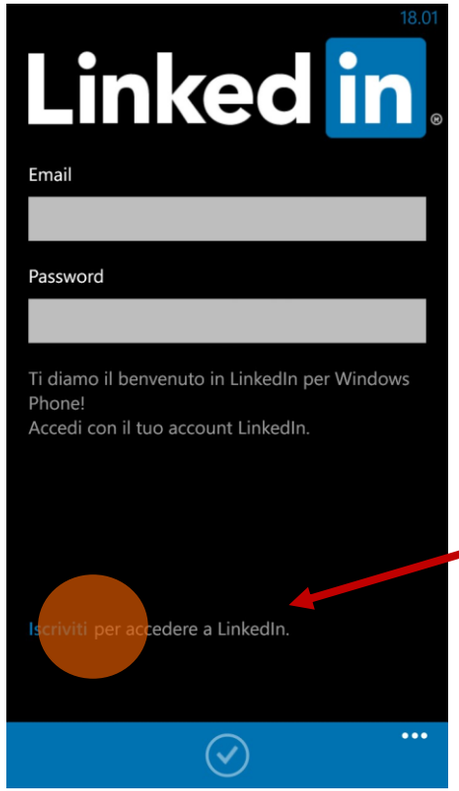
Attacking unencrypted communication – take III

The *Italian job* (or how to manipulate a banking app UI)

securenetwork

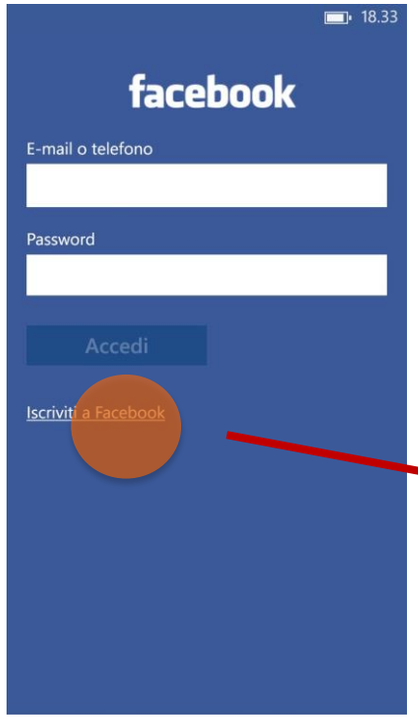
Your Protection, Our Mission

```
<Button x:Name="uxSignUpButton" Margin="12,180,0,240" VerticalAlignment="Bottom" Style="{StaticResource LinkedInPhoneButton}"

   toolkit:TiltEffect.IsTiltEnabled="True" Tag="http://www.linkedin.com/reg/join"   Tap="openHyperlink_Click" >
   <StackPanel Orientation="Horizontal">
      <TextBlock Text="{Binding Path=LocalizedResources.SignIn_SignUp1, Source={StaticResource LocalizedStrings}}"
                 Margin="0,0,0,0" HorizontalAlignment="Left" VerticalAlignment="Top" Style="{StaticResource LinkedInTextAccentStyle}" />
      <TextBlock Text="{Binding Path=LocalizedResources.SignIn_SignUp2, Source={StaticResource LocalizedStrings}}"
                 Margin="6,0,0,0" HorizontalAlignment="Left" VerticalAlignment="Top" Style="{StaticResource PhoneTextSubtleStyle}" />
   </StackPanel>
</Button>
```



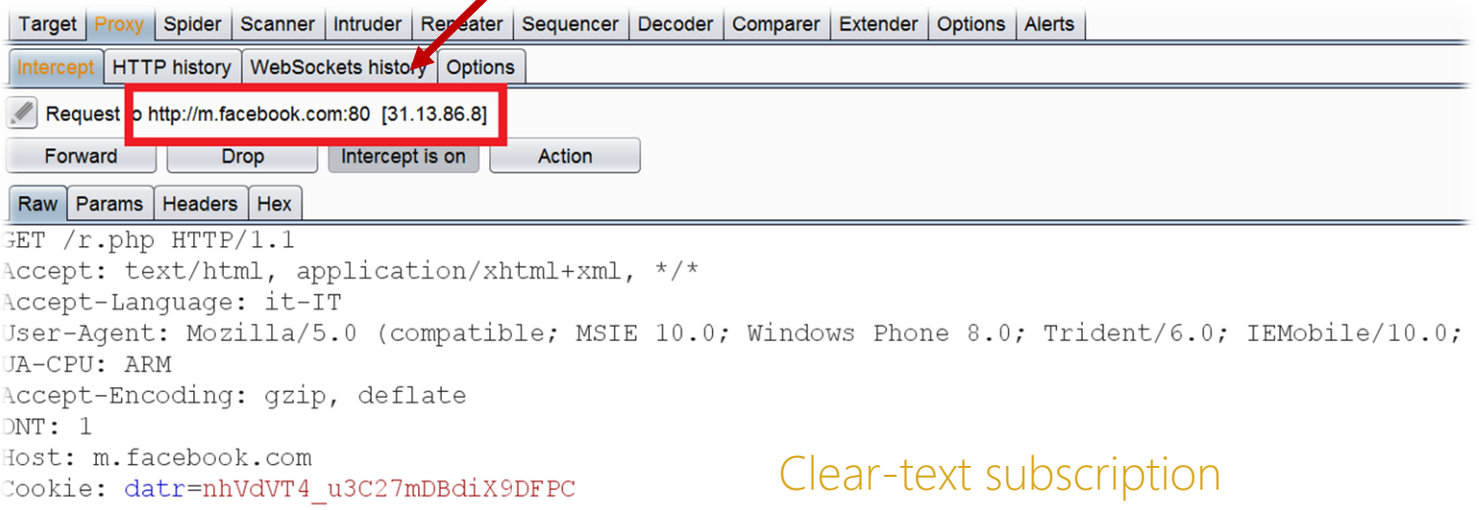
Clear-text subscription

```
private void openHyperlink_Click(object sender, RoutedEventArgs e)
{
   Button button = sender as Button;
   if (button == null)
     return;

   string uriString = button.Tag.ToString();

   if (string.IsNullOrEmpty(uriString) || uriString.Length < 8)
     return;

   if (!uriString.ToLower().StartsWith("http"))
       uriString = "http://" + uriString;

   try
     {
       this._wtask.Uri = new Uri(uriString);
       this._wtask.Show();
     }
```




```
private void aj(object A_0, RoutedEventArgs A_1)
{
   // [...]
   switch (num2 == num3)
   {
     case true:
        int num4 = 0;
        num4 = 0;
        if (num4 == 0);

        num4 = 1;
        if (num4 == 0);

        FbNavService.Current.ShowTask((CustomLauncherBase) new CustomWebBrowserTask()
        {
          Url = "http://m.facebook.com/r.php"
        });
```

Somewhere in the code..



Clear-text subscription

securenetwork
Your Protection, Our Mission

```
public static async void GoToForgotPasswordWebPage()
{
    bool tContinue = NavigationHelper.ShowExitMessageDialog();
    string urlAdjusted =
      TrackingManager.Instance.WrapUriForPaidAppTracking(
         EbaySettings.Instance.CurrentSite.ForgotPasswordSite, "forgotpassword-core"
      );
```

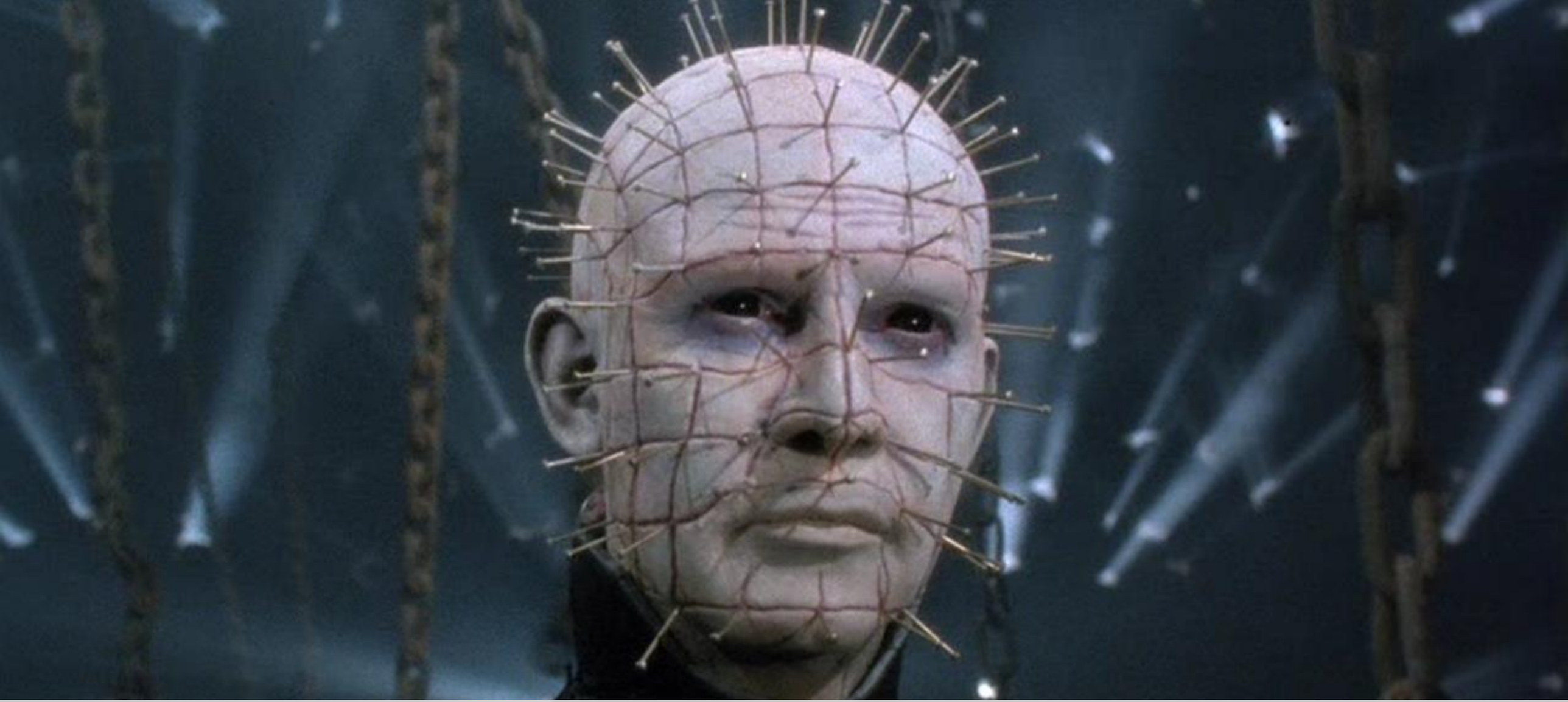

```
public string WrapUriForPaidAppTracking(string inURL, string inMfe)
{
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.Append(string.Format("http://rover.ebay.com/rover/{0}/{1}/{2}?mfe={3}&mpre={4}&mpt={5}", "1",
      this.GetRoverId(EbaySettings.Instance.IsoCodeFromSiteId(EbaySettings.Instance.CurrentSiteId)), "4", inMfe,
        inURL, this.MptCacheBusterValue()));
```

What's your pretext today?

securenetwork
Your Protection, Our Mission

Secure coding tips

- SSL/TLS everywhere!
  - Just adopt proper the **https://** scheme when using **Uri()** object
- WP 8.0 automagically discards invalid certificates
  - No programmatic way to disable the behavior
- WP 8.1 introduced the **IgnorableServerCertificateErrors** class
  - Selective ignore of certificate errors – not all exceptions can be discarded
- Are we completely safe from MiTM attacks? Nope!
  - An attacker can still hack into a Certificate Authority (CA) and forge valid certificates
  - An attacker can induce the victim to install a malicious certificate
- So What?



CERTIFICATE PINNING

Implementing certificate pinning

- Windows Phone 8.0 apps require 3rd parties libraries (e.g., EldoS SecureBlackbox)
- Windows Phone 8.1 provides the **StreamSocket.Information** that returns the **StreamSocketInformation** object [1]
  - **StreamSocketInformation.ServerCertificate** allows getting the remote server digital certificate

```
public async void verifyCertificate(string url) {

   HostName host = new HostName(url);
   StreamSocket socket = new StreamSocket();

   await socket.ConnectAsync(host, "443");
   await socket.UpgradeToSslAsync(SocketProtectionLevel.Ssl, host);

   var cert = socket.Information.ServerCertificate;

   checkCertEntries(cert));
}
```

[1] http://www.slideshare.net/iazza/certificate-pinning-in-mobile-applicationsprosconsv10

DATA STORAGE SECURITY

Device disk encryption

BitLocker disk encryption (AES 128) is supported since WP 8

BitLocker is disabled *by default*

It can be enabled via Exchange ActiveSync policy *RequiredDeviceEncryption*

Device physical memory attacks allow file system content extraction

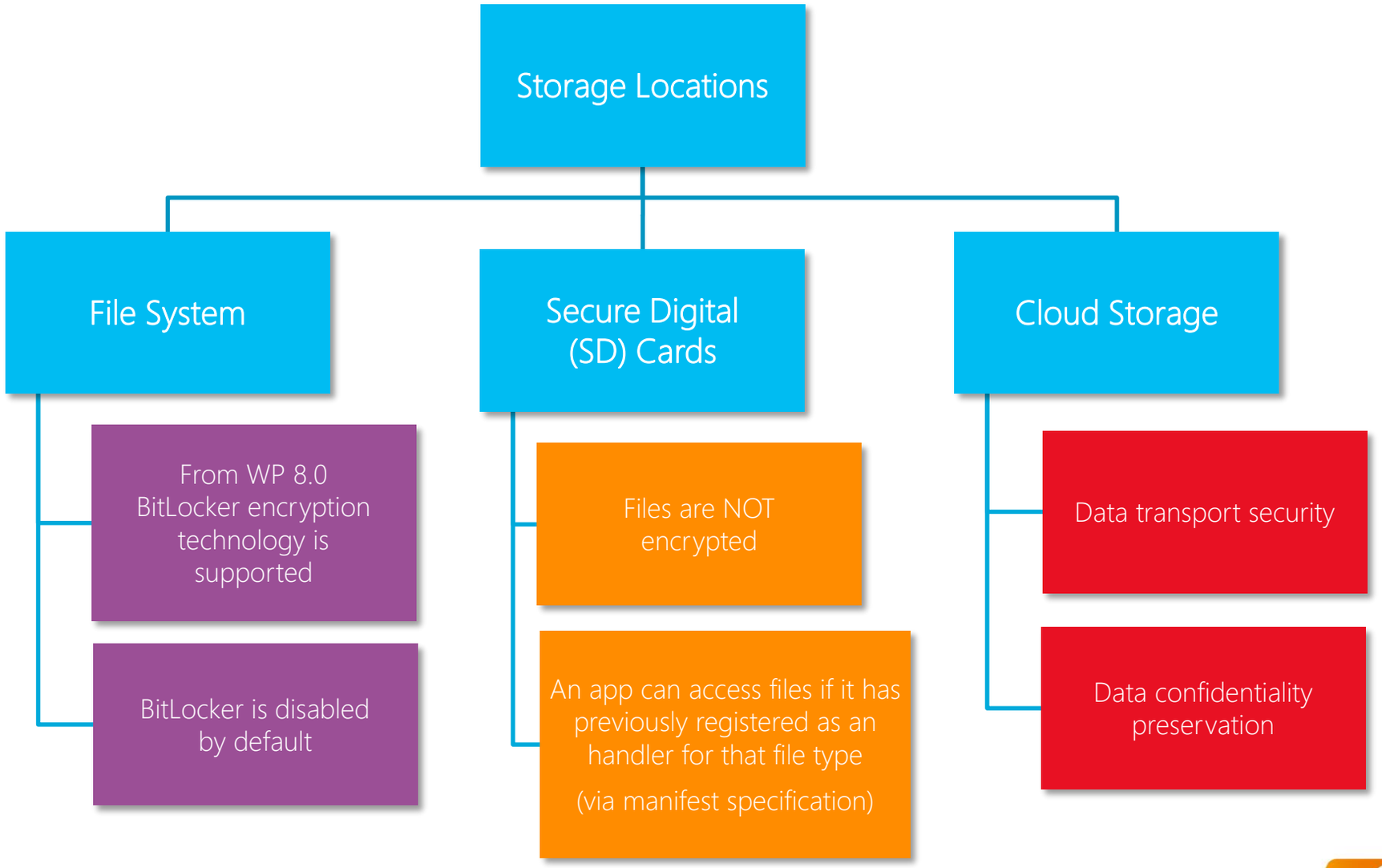
sensitive data

should never be stored on device, even if encrypted*

* I know, this may damage the user experience

securenetwork
Your Protection, Our Mission

Storage locations



Storage locations and physical paths

Locations	Windows Runtime Apps
Local data store	ApplicationData.Current.LocalFolder - URI - ms-appdata:///local/ C:\Data\Users\DefApps\APPDATA\Local\Packages\%packageName%\LocalState
Roaming data store	ApplicationData.Current.RoamingFolder - URI - ms-appdata:///roaming/ C:\Data\Users\DefApps\APPDATA\Local\Packages\%packageName%\RoamingState
Temporary data store	ApplicationData.Current.TemporaryFolder - URI - ms-appdata:///temporary/ C:\Data\Users\DefApps\APPDATA\Local\Packages\%packageName%\TempState
Cache data store	ApplicationData.Current.LocalCacheFolder C:\Data\Users\DefApps\APPDATA\Local\Packages\%packageName%\LocalCache

Storage locations and physical paths

Locations	Windows Runtime Apps
Media Library	KnownFolders.MusicLibrary, KnownFolders.CameraRoll, KnownFolders.PicturesLibrary, KnownFolders.VideosLibrary
Package installation	Windows.ApplicationModel.Package.Current.InstalledLocation URI: ms-appx:// or ms-appx-web:// C:\Data\SharedData\PhoneTools\AppxLayouts\{GUID}\
SD Card	KnownFolders.RemovableDevices
Local Settings and Roaming Settings	Windows.Storage.ApplicationData.Current.LocalSettings Windows.Storage.ApplicationData.Current.RoamingSettings
Cache data store	ApplicationData.Current.LocalCacheFolder C:\Data\Users\DefApps\APPDATA\Local\Packages\%packageName%\LocalCache

Local and Roaming Setting save data in
C:\Data\Users\DefApps\APPDATA\Local\Packages\%packageName%\Settings\settings.dat
which is a Windows NT registry file (REGF) - and NOT encrypted

Storage locations and physical paths

Locations	Silverlight Apps
Application local folder	C:\Data\Users\DefApps\APPDATA\{GUID}\Local
Application Settings	IsolatedStorageSettings.ApplicationSettings C:\Data\Users\DefApps\APPDATA\{GUID}\Local__ApplicationSetting
Package installation	Windows.ApplicationModel.Package.Current.InstalledLocation C:\Data\Programs\{GUID}\Install
Cached data	C:\Data\Users\DefApps\APPDATA\{GUID}\INetCache
Cookies	C:\Data\Users\DefApps\APPDATA\{GUID}\INetCookies
SD Card	(read only)

Credentials stored in clear-text

```
private async void DoLogin()
{
     bool? isChecked = this.checkBoxRicordami.IsChecked;
     if ((!isChecked.GetValueOrDefault() ? 0 : (isChecked.HasValue ? 1 : 0)) != 0)
      this.saveCredentials();

   // [...]

private void saveCredentials()
{
  if (!(this.textBlockUsername.Text != "") || !(this.textBlockPassword.Password != ""))
    return;

  this.storageSettingsRememberMe.Remove("Username");
  this.storageSettingsRememberMe.Remove("Password");
  this.storageSettingsRememberMe.Remove("isChecked");

  this.storageSettingsRememberMe.Add("Username", this.textBlockUsername.Text);
  this.storageSettingsRememberMe.Add("Password", this.textBlockPassword.Password);
  this.storageSettingsRememberMe.Add("isChecked", true);
  this.storageSettingsRememberMe.Save();
}
```

credentials saved in application setting file

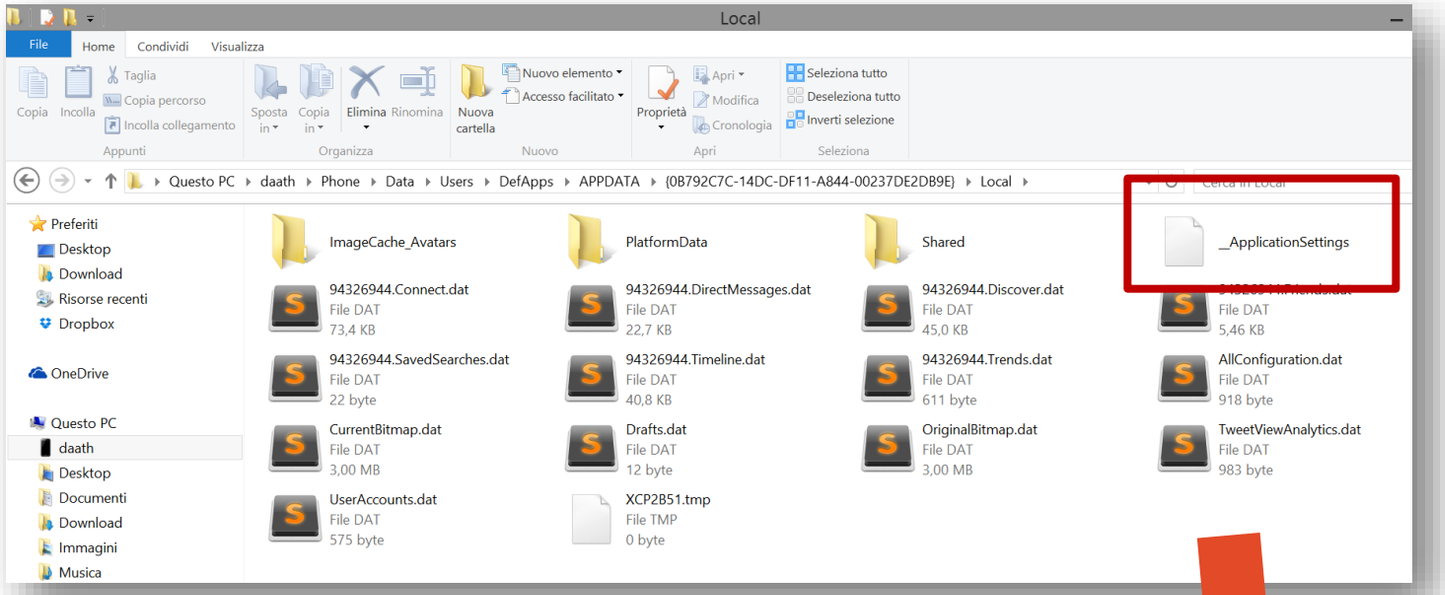
Weak “encryption” mechanism

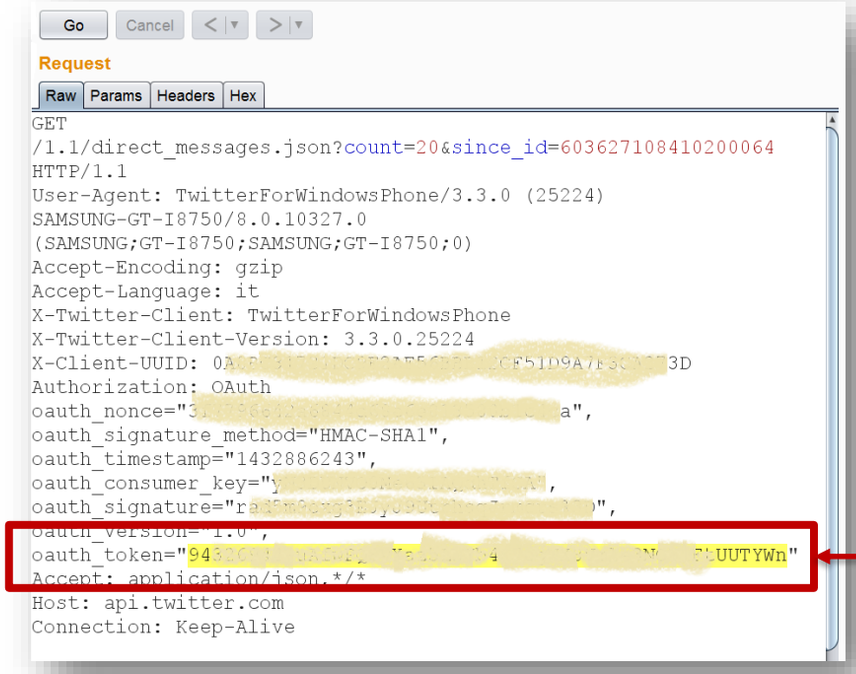
```
private void GetUserCompleted(object sender, EventArgs e)
{
   if (e == null)
   {
     // ...
   }
   else
   {
     NetUserCompletedEventArgs completedEventArgs = (NetUserCompletedEventArgs) e;
     byte[] numArray1 = Crypto.encryptString(completedEventArgs.user.username);
     byte[] numArray2 = Crypto.encryptString(completedEventArgs.user.password);
     this.isolatedStorageSettings.StoreValueForKey("Username", (object) numArray1);
     this.isolatedStorageSettings.StoreValueForKey("Password", (object) numArray2);
     CurrentAppConfig.Instance.User = completedEventArgs.user;
     this.storeCurrentUserStoresPreferences(completedEventArgs.user);
   }
 }

  public class Crypto
  {
    public static byte[] encryptString(string input)
    {
      return Encoding.UTF8.GetBytes(input);
    }
```

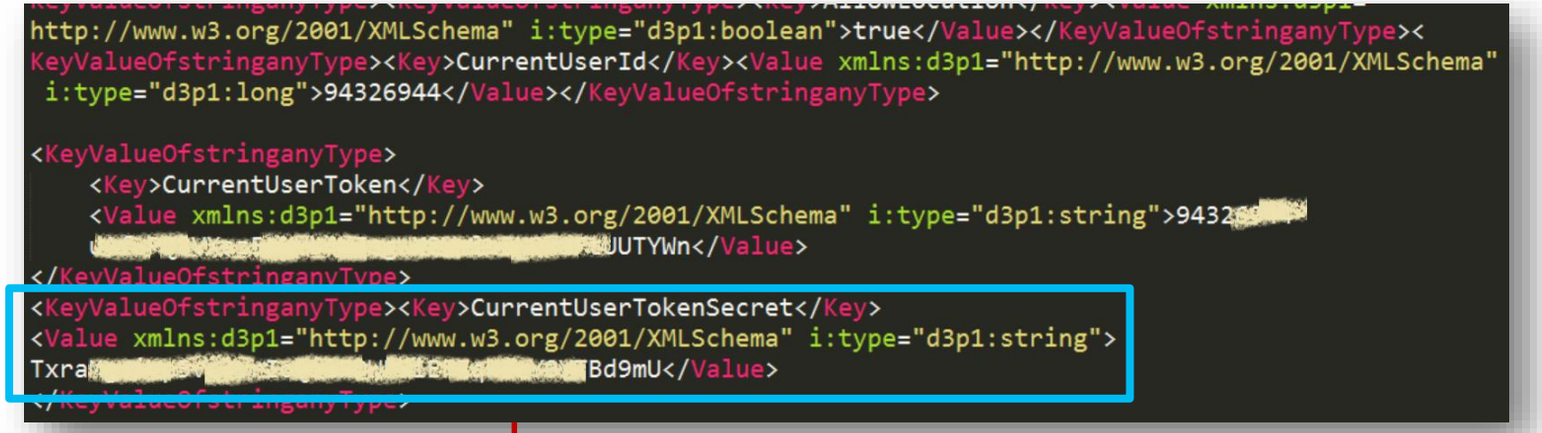
“encrypted” credentials are stored into the sandbox

encoding

is just a data representation, not encryption (at all)







securenetwork
Your Protection, Our Mission

Hunting for insecure data storage

Locations	Classes, Methods and Properties	
Local folders	StorageFile	OpenReadAsync() - OpenAsync() GetFileFromApplicationUriAsync() - GetFileFromPathAsync()
	StorageFolder	GetFilesAsync() - GetFileAsync() - CreateFileAsync()
	IsolatedStorageFile.CreateFile() | IsolatedStorageFile.OpenFile()	
Application or Roaming Settings	IsolatedStorageSettings.ApplicationSettings – property ApplicationData.LocalSettings – property ApplicationData.RoamingSettings - property	
SD Card (WP 8.1 only)	KnownFolders.RemovableDevices returns a StorageFolder object that can be sequentially used to read/write data from the SD card	
Local database	Identify objects that inherit from System.Data.Linq.DataContext. Verify the existence of reserved data stored in the local .sdf file	

Secure coding tips

- Mind my mantras ☺
- The **Data Protection API** (DPAPI) should be used to encrypt data
- Account credentials should be protected using the **PasswordVault** class
- Never hardcode encryption keys
- Never place encryption keys in unsafe device areas
- Do not use custom encryption algorithms

DATA LEAKAGE

Data leakage

- (unintended) data leakage is addressed by M4 MTT for 2014
- Involuntary data exposure caused by OS or frameworks *side-effects*
  - System caching
  - Application backgrounding
  - System logging
  - Telemetry frameworks which expose private data
- A privileged access to target device file system - or connected network - is required to properly exploit these issues

Video

Local exploiting of data leakage

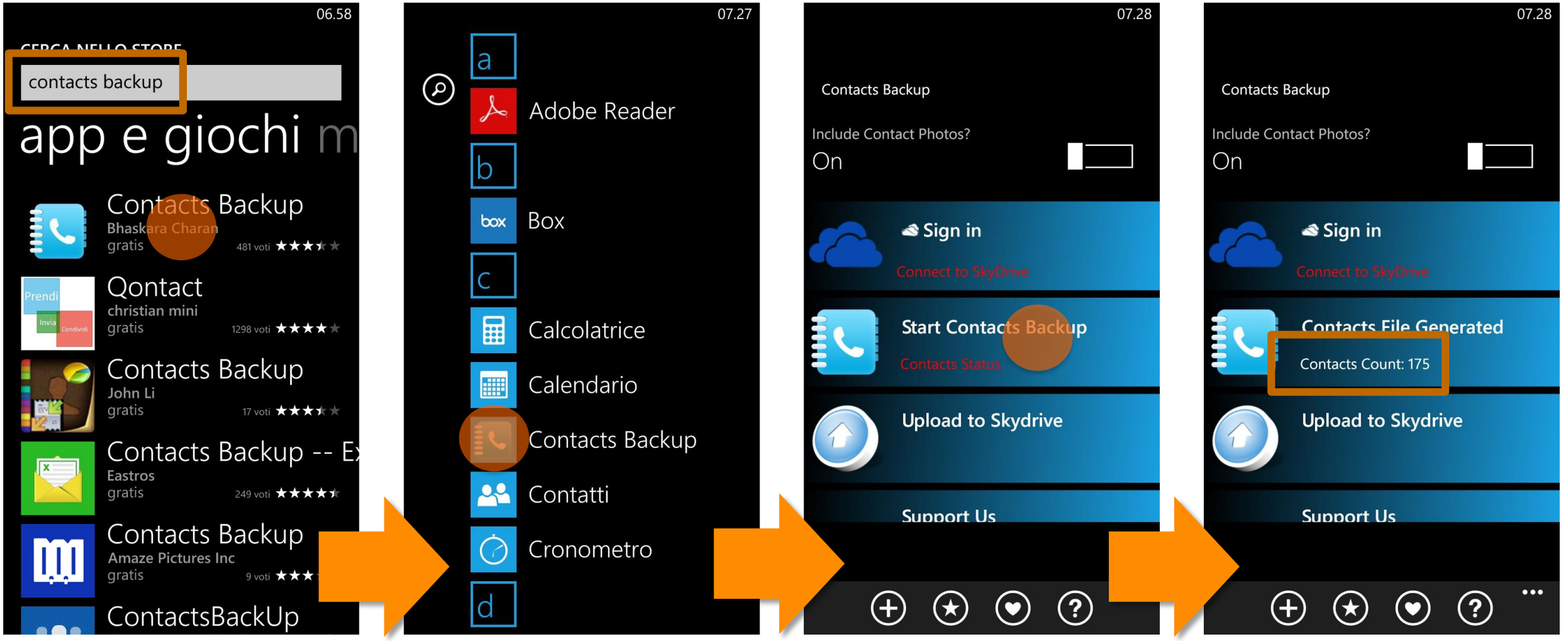
The PIN-protected box who leaked its token

securenetwork
Your Protection, Our Mission

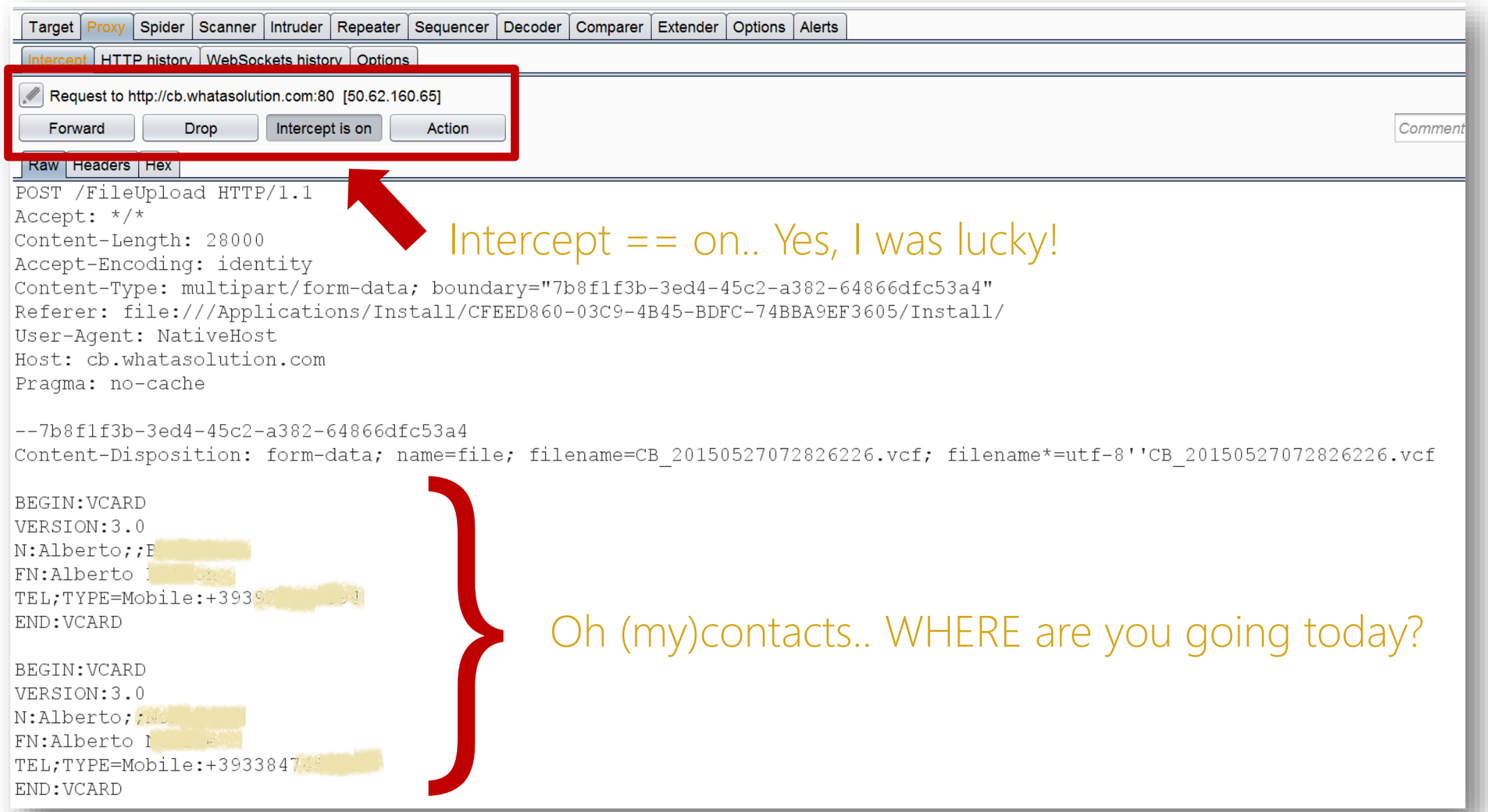
it was a dark and stormy night..

.. and I was analyzing my favorite contacts backup app..

securenetwork
Your Protection, Our Mission



securenetwork
Your Protection, Our Mission



Intercept == on.. Yes, I was lucky!

Oh (my)contacts.. WHERE are you going today?

securenetwork
Your Protection, Our Mission

```
        private void preparecontacts()
        {
            int cellcnt = 0;
            int emailcnt = 0;

            // [...]

            if (str == "")
              this.UploadFile();
            else if (new DateTime(Convert.ToInt32(str.Substring(0, 4)),
                Convert.ToInt32(str.Substring(4, 2)), Convert.ToInt32(str.Substring(6, 2))).AddDays(1.0) < DateTime.Now)
              this.UploadFile();
```

```
private async void UploadFile()
{
   StreamReader reader1 = new StreamReader((Stream) new IsolatedStorageFileStream(this.sFile1, FileMode.Open, this.myFile));
   byte[] byteArray1 = Encoding.UTF8.GetBytes(reader1.ReadToEnd());
   reader1.Close();
   MemoryStream fileStream1 = new MemoryStream(byteArray1);
   string fileUploadUrl = "http://cb.whatasolution.com/FileUpload";
   HttpClient client = new HttpClient();
   fileStream1.Position = 0L;
   MultipartFormDataContent content = new MultipartFormDataContent();
   content.Add((HttpContent) new StreamContent((Stream) fileStream1), "file", "CB_" + DateTime.Now.ToString("yyyyMMddHHmmssfff") + ".vcf");

   try
   {
      await client.PostAsync(fileUploadUrl, (HttpContent) content).ContinueWith((Action<Task<HttpResponseMessage>>) (postTask =>
      {
         try
         {
           postTask.Result.EnsureSuccessStatusCode();
         }
         // [...]
```

Well, a “disaster recovery” backup rules..

securenetwork
Your Protection, Our Mission

Hunting for potential data leakage

Conditions	Classes, Methods or Properties
Application Backgrounding and Closing	Handler for the Application.Suspending event, typically the OnSuspending() method in App.xaml.cs
	Handler for the Application.Deactivated event, typically the Application_Deactivated() method in App.xaml.cs
	Handler for the Application.Closing event, typically the Application_Closing() method in App.xaml.cs
	Handler for the Application.UnhandledException event, typically the Application_UnhandledException() method in App.xaml.cs
Use of Telemetry Frameworks	HockeyApp, BugSense, etc.

Secure coding tips

Actions	Classes, Methods or Properties	
Remove cached data on app closing, suspension or deactivation	server-side	Cache-Control: no-store
	client-side	WebBrowserExtensions.ClearInternetCacheAsync() WebBrowser.ClearInternetCacheAsync() WebView - no programmatic way
Remove stored cookies	WebBrowser.ClearCookiesAsync() WebBrowserExtensions.ClearCookie() WebView – use HttpCookieManager.GetCookies() + HttpCookieManager.DeleteCookie()	

AUTHORIZATION AND AUTHENTICATION ISSUES

Authorization and authentication issues

- Security decisions without *server-side engagement*
  - M5 - Poor Authorization and Authentication (MTT 2014)
  - May also involve M7 – Security Decisions via Untrusted Inputs
- Common issues
  - Offline authentication
  - Issues related to password complexity (e.g., 4 digits PIN)
  - Client-side generation of (predictable) authorization tokens
  - Authorization issues on *premium* functionalities or data access

Hunting for insecure tokens forgery

Identification Data	Namespaces	Classes, Methods or Properties
Device Name	Microsoft.Phone.Info	DeviceStatus.DeviceName
Hardware Identification	Microsoft.Phone.Info	DeviceExtendedProperties.GetValue("DeviceName")
		DeviceExtendedProperties.GetValue("DeviceUniqueId")
Hashing Functions	Windows.Security.Cryptography	SHA1Managed, SHA256Managed, SHA384Managed and SHA512Managed classes (or any other 3° party libraries implementing these functions)
	Windows.Security.Cryptography.Core	HashAlgorithmProvider.OpenAlgorithm()
Geo Location Coordinates	Windows.Devices.Geolocation	Geolocator / Geoposition / Geocoordinate
	System.Device.Location	GeoCoordinateWatcher / GeoPosition / GeoCoordinate

Weak custom code encryption

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo(e);
    using (IsolatedStorageFile storeForApplication = IsolatedStorageFile.GetUserStoreForApplication())
    {
        this.fileExists = storeForApplication.FileExists("wp contacts backup.zip");
        if (!this.fileExists)
        {
            this.infoTextBlock.Text = "No backup file exists! Please create one before trying to download it.";
        }
        else
        {
            try
            {
                this.server = new HttpServer(2, 65536);
                this.server.Start(new IPEndPoint(IPAddress.Parse("0.0.0.0"), 5656));
                this.server.TextReceived += new EventHandler<HttpDataReceivedEventArgs>(this.server_TextReceived);
                this.infoTextBlock.Text = "http://" + this.server.LocalEndpoint.ToString();
            }
            catch
            {
                this.infoTextBlock.Text = "Unable to start WEB Server. Please check your connectivity settings.";
            }
        }
```

contacts backup file is stored in app's sandbox

Weak custom code encryption

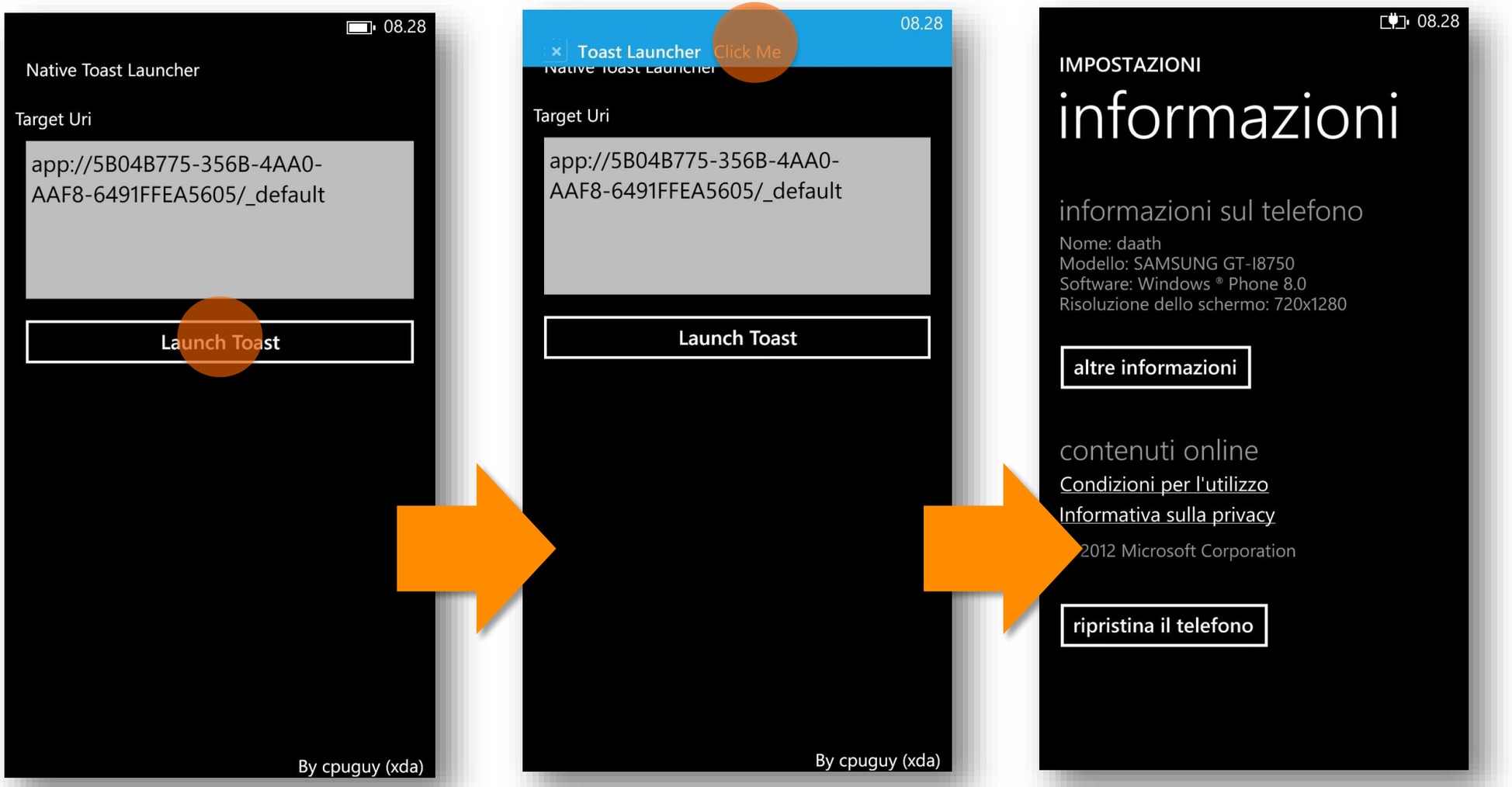
```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo(e);
    using (IsolatedStorageFile storeForApplication = IsolatedStorageFile.GetUserStoreForApplication())
    {
        this.fileExists = storeForApplication.FileExists("wp contacts backup.zip");
        if (!this.fileExists)
        {
            this.infoTextBlock.Text = "No backup file exists! Please create one before trying to download it.";
        }
        else
        {
            try
            {
                this.server = new HttpServer(2, 65536);
                this.server.Start(new IPEndPoint(IPAddress.Parse("0.0.0.0"), 5656));
                this.server.TextReceived += new EventHandler<HttpDataReceivedEventArgs>(this.server_TextReceived);
                this.infoTextBlock.Text = "http://" + this.server.LocalEndpoint.ToString();
            }
            catch
            {
                this.infoTextBlock.Text = "Unable to start WEB Server. Please check your connectivity settings.";
            }
        }
```

local web server lacks any authentication mechanism

Introducing IPC with Windows Phone

- Windows Phone provides limited support to Inter Process Communication (IPC)
  - WP 7.x does not support IPC
  - WP 8.x provides file and URI association
- A third undocumented IPC exists
  - Shell_PostMessageToast (ShellChromeAPI.dll) allows performing Cross-Application Navigation Forgery attacks
  - A malicious app can send a ***toast message*** that, once tapped, allows to open an arbitrary XAML page of an arbitrary app – **XAML page code behind can be fed with malicious input**

app://{GUID}/_default#/AssemblyName;component/Page.xaml?par=val1&par2=val2



Video

Attacking a weak authentication mechanism
Bypassing DropBox security passcode

securenetwork
Your Protection, Our Mission

Behind the bypass

app://47e5340d-945f-494e-b113-b16121aeb8f8/_default#/Dropbox.WindowsPhone80;component/Pages/Lock/LockPage.xaml?type=1

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
   // [...]

   this.ViewModel.Init(Enum.Parse(typeof (LockPageType), this.NavigationContext.QueryString["type"]));
}
```

```
public void Init(LockPageType type)
   {
     this.NbrTry = 0;
     this.Type = type;

     if (this.Type == LockPageType.CHANGEPIN)
       this._createstep = CreationStep.ENTEROLDPASSCODE;

     this.ManageType();
   }
```

this.Type = LockPageType.CREATEPIN

```
namespace Dropbox.Core.ViewModels.Lock
{
    public enum LockPageType
    {
        UNLOCK,     // 0
        CREATEPIN,  // 1
        CHANGEPIN,  // 2
        DISABLEPIN, // 3
    }
}
```

securenetwork
Your Protection, Our Mission

Behind the bypass

...component/Pages/Lock/LockPage.xaml?type=1

this.Type = LockPageType.CREATEPIN = 1

```
public void ManageType()
{
    switch (this.Type)
    {
        case LockPageType.CREATEPIN:

            switch (this._createstep)
            {
              case CreationStep.ENTERPASSCODE:
                this.LegendText = AppResources.ProtectionEnterPin;
                break;
              case CreationStep.VERIFYPASSCODE:
                this.LegendText = AppResources.ProtectionVerifyPin;
                break;
            }
```

So we can overwrite the previous passcode and..

securenetwork
Your Protection, Our Mission

Hunting for insecure IPC

Actions	Platform	Namespaces	Classes, Methods and Properties
URI associations	WP 8.0	System.Windows.Navigation	overridden UriMapperBase.MapUri() method
	WP 8.1	Windows.Ui.Xaml.Application	OnActivated() - ActivationKind.Protocol property
File associations	WP 8.0	System.Windows.Navigation	overridden UriMapperBase.MapUri() method
	WP 8.1	Windows.Ui.Xaml.Application	OnFileActivated() method
(Toast Message)	WP 8.0	System.Windows.Navigation	OnNavigatedTo() (NavigationContext.QueryString)

Secure coding tips

- Avoid client-side generation of tokens
- Avoid using **DeviceExtendedProperties.GetValue("DeviceUniqueId")**
  - The returned identification is unique only *per device*
- **HostInformation.PublisherHostId** property is unique *per device* and *per publisher*
  - A malicious app should be published by the author of the targeted one to steal the ID
- Positive validate all your input and authorize every actions
  - Carefully audit each OnNavigatedTo() methods!

CLIENT SIDE INJECTIONS

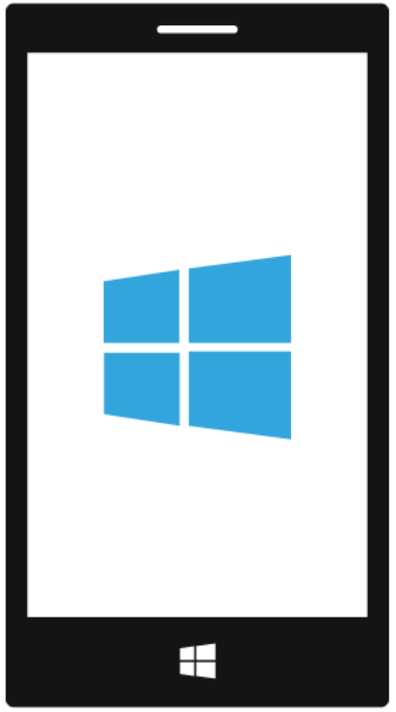
Client-side injection

- Feeding an interpreter with untrusted data
  - Similar to the server-side ones, but the interpreter resides at the “app-side”
- Different interpreters exist – and so for the related injection
  - Offline authentication
  - Local database querying systems
  - XML parsers
  - HTML and JavaScript engines
  - File handling routines
- Attacks impact depends on the data handled by the interpreter

all input is evil

trust no one and map the sources for malicious data

Hunting for untrusted data sources



- Input from network
- Bluetooth and NFC
- Inter Processor Communication (IPC) mechanisms
- Files accessed from SD card – which is a shared storage area
- User typed input – via UI, speech to text, camera (QR code), USB data, ..

Hunting for injections

Interpreters	Namespaces	Classes, Methods and Properties	
HTML/JavaScript	Microsoft.Phone.Controls	WebBrowser	NavigateToString() InvokeScript() IsScriptEnabled = true (property)
	Windows.UI.Xaml.Controls	WebView	NavigateToString() InvokeScript() InvokeScriptAsync() NavigateToLocalStreamUri() NavigateWithHttpRequestMessage()
XML	System.Xml.Linq	XDocument.Load()	
	System.Xml	XmlReaderSettings.DtdProcessing = DtdProcessing.Parse	
XAML	System.Windows.Markup	XamlReader.Load()	

Hunting for injections

Interpreters	Namespaces	Classes, Methods and Properties
SQL	SQLitePCL	SQLiteConnection.Prepare()
	SQLite-Net-WP8	Query() / Query<T>() / QueryAsync<T>() Execute() / ExecuteAsync() ExecuteScalar<T>() / ExecuteScalarAsync<>() DeferredQuery() / DeferredQuery<T>() FindWithQuery<T>() CreateCommand()
	CSharp-SQLite	IDbCommand.CommandText (property)
	SQLiteWinRT	Database.ExecuteStatementAsync() Database.PrepareStatementAsync()

Hunting for injections

Interpreters	Namespaces	Classes, Methods and Properties
File handling	StorageFolder	CreateFileAsync() RenameAsync() GetFolderFromPathAsync() GetFolderAsync()
	StorageFile	CopyAsync() GetFileFromApplicationUriAsync() GetFileFromPathAsync() RenameAsync()
	IsolatedStorageFile	OpenFile() CopyFile() CreateDirectory() – CreateFile() DeleteDirectory() - DeleteFile()

Just click to XSS

```
private void ButtonView_Click(object sender, RoutedEventArgs e)
{
      this.ButtonView.IsEnabled = false;
      this.iptexttemp = this.TextIP.Text.Trim() + "xxxxxxxx";
      this.WebBrowser1.NavigateToString("<body bgcolor=black>" +
        "<form action='https://www.REMOTE-SITE.com/path/resource.asp' method=post>" +
          "<input name='iname' value='" + this.TextAdmin.Text.Trim() + "' type='hidden'>" +
            "<input name='pword' value='" + this.PasswordBox1.Password.Trim() +
              "' type=hidden><input name='ip' value='" + this.TextIP.Text.Trim() +
                "' type=hidden><input name='port' value='" + this.TextPort.Text.Trim() +
                "' type=hidden><input name='versi' value='ori' type='hidden'></form>" +
                "<script>document.forms[0].submit();</script></body>");
```

app renders user-controlled data
without any validation

Video

Cross-Site Scripting (XSS) attack via IPC
Stealing Vodafone Business app credential via XSS

securenetwork
Your Protection, Our Mission

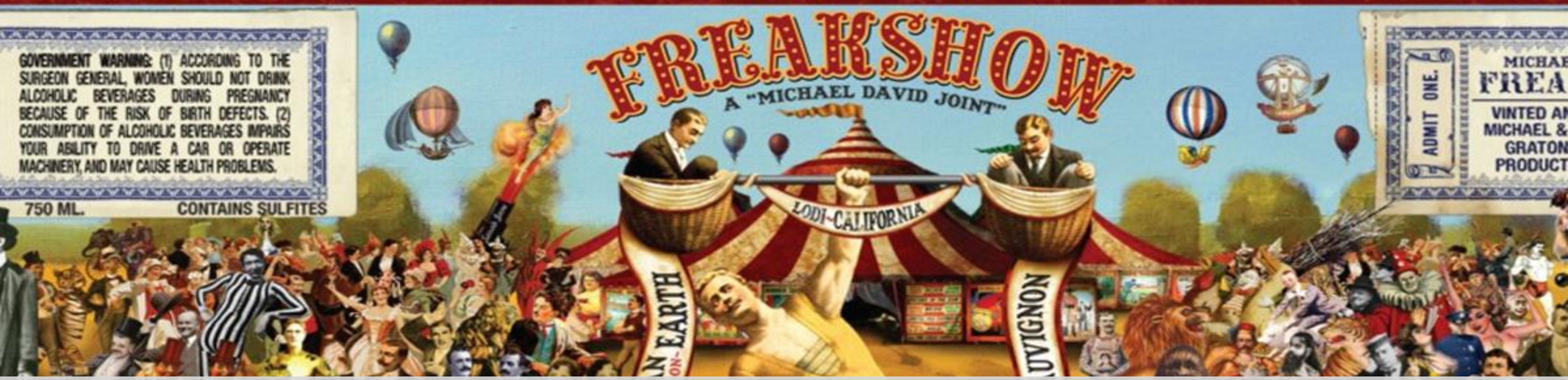
Vulnerable NavigateToString() method

app://945b96a7-aadc-4dd0-806a-c2d1e0e6ca9a/_default#/VodafoneMyBusiness;component/DetailFaq.xaml?Answer=INJECTION

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo(e);
    if (this.NavigationContext.QueryString.ContainsKey("Question"))
    {
      this.QuestionArrived = this.NavigationContext.QueryString["Question"];
      this.textBlockQuestion.Text = this.QuestionArrived;
    }

    if (this.NavigationContext.QueryString.ContainsKey("Answer"))
    {
      this.AnswerArrived = this.NavigationContext.QueryString["Answer"];

      this.webView.NavigateToString("<html><head>" + "<script type=\"text/javascript\"> function getSize(){ " +
          "var h = document.getElementById('content').offsetHeight; var s = \"rendered_height=\" + h; window.external.notify(s);} " +
            "</script> <style type=\"text/css\">body {font-family: \"segoe\"; font-size:19;}</style></head><body><div id=\"content\">" +
            Uri.UnescapeDataString(this.AnswerArrived) + "</div></body></html>");
    }
```

Secure coding tips

- Implement proper input *positive* validation
- Prevent XSS – **WebView** and **WebBrowser** controls
  - Validate parameters passed to InvokeScript(), Navigate(), NavigateToString(), etc.
- Avoid SQL Injections
  - Use LINQ to SQL :-P
  - Adopt parameterized query
- Validating file names/paths handled by methods defined for
  - StorageFolder
  - StorageFile
  - IsolatedStorageFile



FINAL CONSIDERATIONS

Final considerations

Well, **I don't like writing conclusion**, but I have to. A 30-pages-long whitepaper – yeah, it was pretty hard - will be released in 15 minutes - thanks Dhillon. It simply represents the first public catalog of insecure usage of APIs provided by Windows Phone SDK, and covers both Silverlight and Windows Runtime technologies. Substantial part of my work – the Windows Runtime one - will be valid for Universal Apps, too – so your favorite Windows 10 app can be safely developed as well. That's it. Thanks to my wife **Silvia**, **Stefano** and my awesome Tiger Team := {**Giovanni, Alberto, Eros, Francesco, Primo, Filippo, Matteo**} for supporting me during the research! Btw, thanks for your attention and I hope you enjoyed the talk. See ya. Ciao Mamma!

THANKS

@_daath ~ luca@securenetwork.it ~ nibblesec.org