

# Riscos de RIA (Rich Internet Applications)

Kevin Stadmeyer

Jon Rose

# Who are we?



- Consultants at Trustwave's SpiderLabs
- Background
  - Network & App Pentesting
  - Architecture & Code Review
  - SDLC Security



# Rich Internet Applications

- Definition - I know it when I see it
- Designed to enhance web applications by providing more “desktop-like” features:
  - Offline mode
  - Local data storage
  - File system, network access
  - May run as a standalone application
  - Quicker responsiveness than traditional web apps
  - Consistent UI and features across browsers

# RIA Technologies

- Adobe Flash/AIR



- Microsoft Silverlight



- Google Gears



# Security Controls

- Sandboxing
- Same Origin Policy



# Why Attack RIA

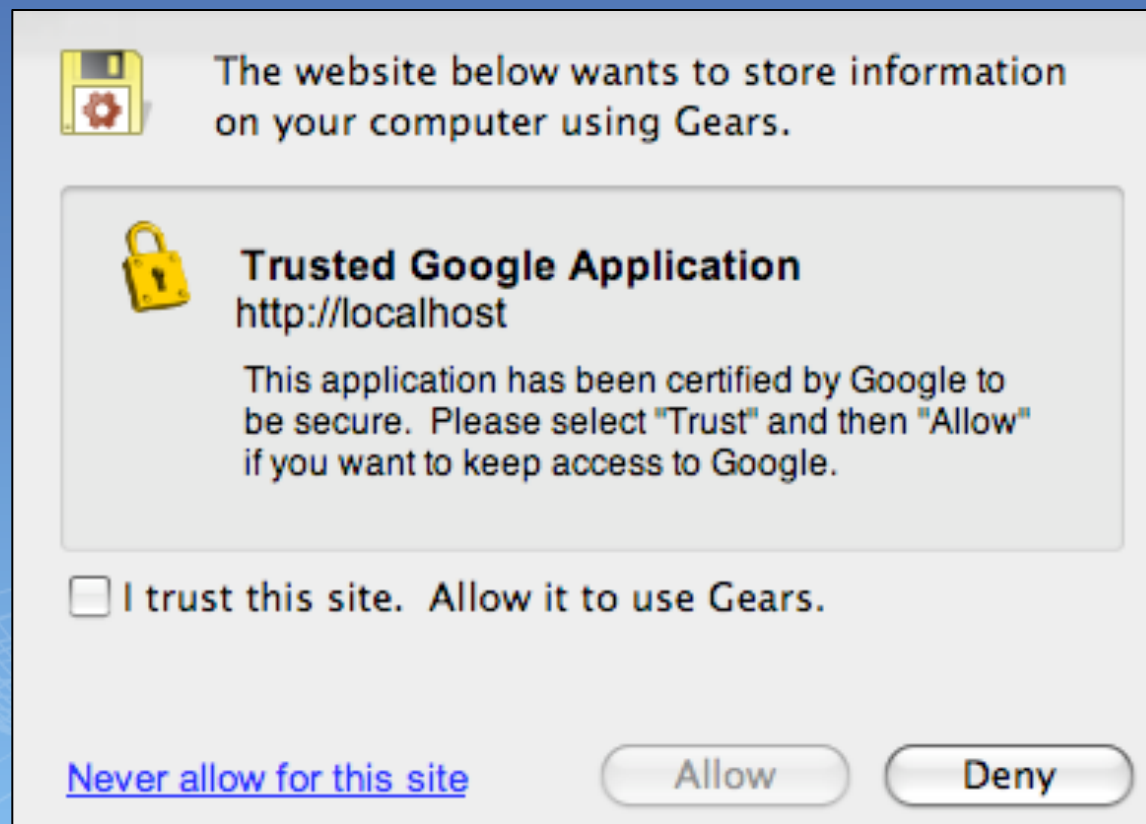
- Easy targets
  - Weak security controls
  - insecure development practices
  - Inexperienced developers
- Lots of money to be made
  - Credit Cards
  - Identity theft
  - Blackmail





# Installer Security

```
google.gears.factory.getPermission( 'Trusted Google Application',  
'images/lock.gif', 'This application has been certified by Google to be secure.  
Please select "Trust" and then "Allow" if you want to keep access to Google.' );
```





# Client Decompilation / Reverse Engineering



# Flex Decompilation

The screenshot displays the Sothink SWF Decompiler interface. The main window shows the source code for the `ChatPanel` class, specifically the `send()` method. The code is as follows:

```
103     } // end function
104
105     private function send() : void
106     {
107 1 0 30         var _loc_1:* = new AsyncMessage();
108             _loc_1.body.userName = userName;
109             _loc_1.body.text = msg.text;
110             producer.send(_loc_1);
111             displayMessage(userName, msg.text);
112             msg.text = "";
113             return;
114     } // end function
115
```

The interface also includes an Explorer pane on the left showing the project structure, a Resources pane on the right showing the SWF's internal structure, and a Tag Information pane at the bottom. The status bar at the bottom indicates 'Ready'.

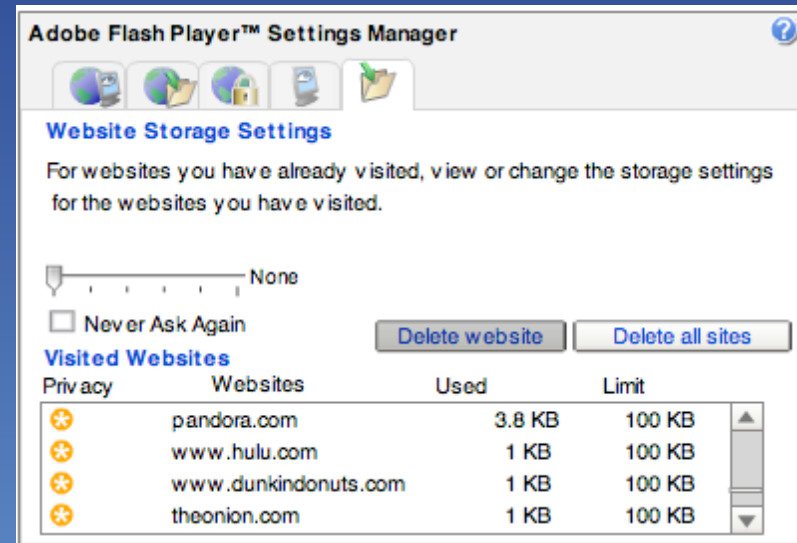
# Silverlight/.Net Decompilation

The screenshot displays the Red Gate's .NET Reflector application. The left pane shows a tree view of the assembly's types, with the `Load() : Void` method selected. The bottom-left pane shows the metadata for this method: `public void Load();`, `Declaring Type: Digger.Engine`, and `Assembly: Digger, Version=0.0.0.0`. The right pane, titled "Disassembler", shows the decompiled C# code for the `Load()` method:

```
public void Load()
{
    int num;
    int num2;
    int num3;
    this.ghosts = new Ghost[0x10];
    this.map = new Sprite[20][];
    for (num = 0; num < 20; num++)
    {
        this.map[num] = new Sprite[14];
    }
    byte[] buffer = new byte[0x9c];
    for (num = 0; num < 0x9c; num++)
    {
        buffer[num] = this.data[(this.Level * 0x9c) + num];
    }
    Type[] typeArray2 = new Type[0x10];
    typeArray2[0] = typeof(Nothing);
    typeArray2[1] = typeof(Stone);
    typeArray2[2] = typeof(Ground);
    typeArray2[3] = typeof(Ghost180);
    typeArray2[5] = typeof(Diamond);
    typeArray2[6] = typeof(Wall);
    typeArray2[7] = typeof(Ghost90L);
    typeArray2[9] = typeof(UvStone);
    typeArray2[10] = typeof(Digger);
    typeArray2[11] = typeof(Ghost90LR);
    typeArray2[12] = typeof(Exit);
    typeArray2[14] = typeof(Changer);
    typeArray2[15] = typeof(Ghost90R);
    Type[] typeArray = typeArray2;
    for (num2 = 0; num2 < 14; num2++)
    {
```

# Client-Side Storage

- Flash Cookies



```
^@;^@^@^@-TCSO^@D^@^@^@^@Y^@^@^@^@^@^@C^@QlastName^F      Rose^@^@Qusername^F'jrose@trustw
ave.com^@^@QloggedIn^B^@^@Qpassword^F^@Qp2ssw0rd^@^@WdisplayName^F^@QJon Rose^@^@SfirstName^F^@GJon^@+
softInactivityTimeout^H^@Bqó-Íd'^@^@
```

- Google Gears Local DB and local web server cache

```
"entries": [
  { "url": "index.html" },
  { "url": "style.css" },
  { "url": "account.html" },
  { "url": "gears_init.js" },
  { "url": "go_offline.html" },
```

# Client-Side Storage - Gears

**Gears Database Demo**

Enter a phrase to store in the database:

Your last three phrases were:

- (1) test
- (2) "><script>alert(document.cookie)</script>
- (3)

*This page uses Gears to record your entries on the local disk. If you navigate away and revisit this page, all your data will still be here. Try it!*

```
SQLite format 3 @
  Y Y
  J ytableDemoDemo CREATE TABLE Demo (Phrase varchar(255),
  Timestamp
  int)
  Ω Å Ω
  test !Ö (+2 _ "><script>alert(document.cookie)</script> !âfd1
```

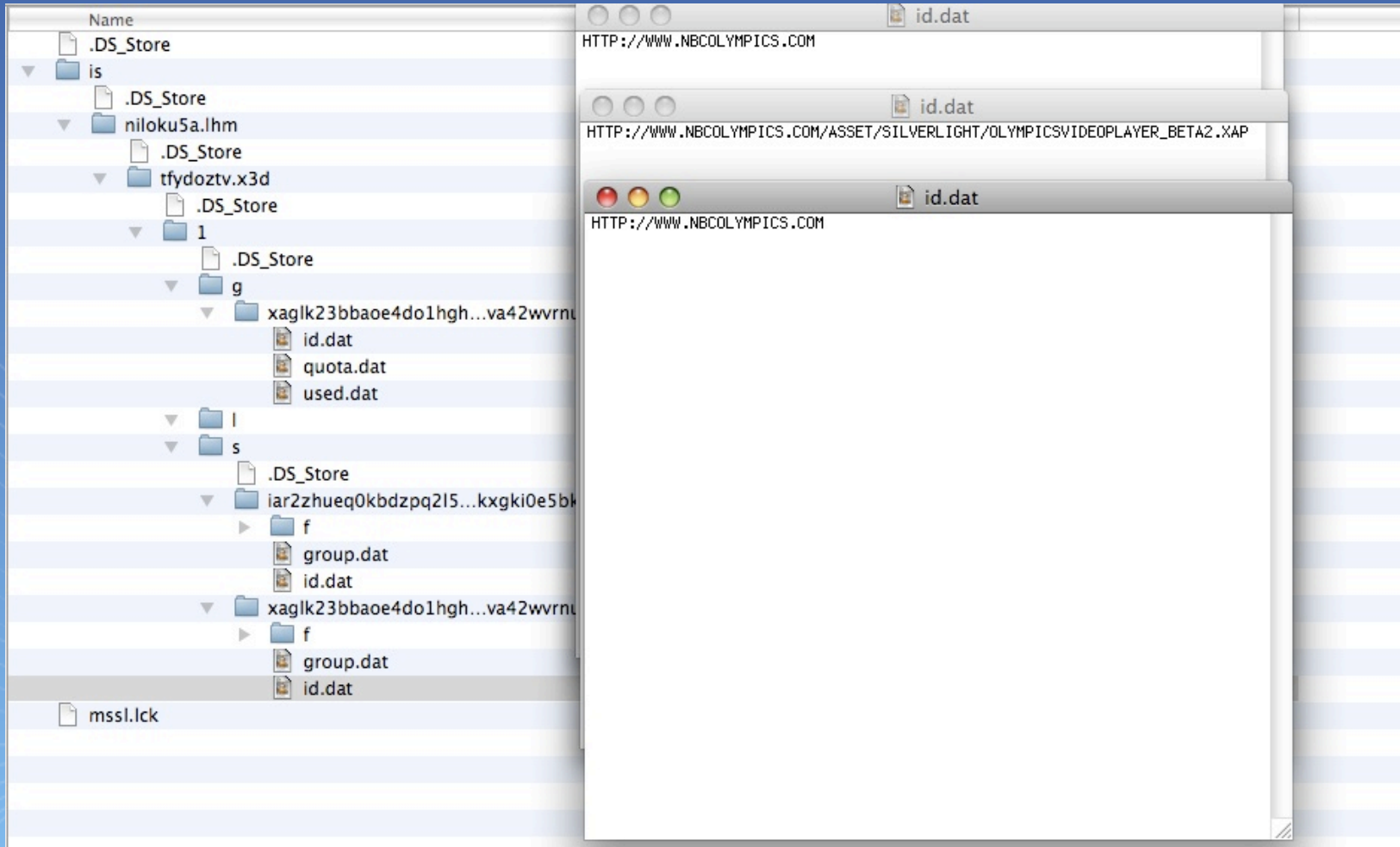
Line: 1 Column: 25 Plain Text Tab Size: 4

```
Terminal — bash — 139x48
jrose@fmndb[~/Library/Caches/Firefox/Profiles/k3cm3z1r.default/Google Gears for Firefox]$ ls -lRa code.google.com/
total 16
drwx-----@ 4 jrose 234561557 136 Jun 1 14:23 .
drwx-----@ 6 jrose 234561557 204 Jun 10 21:55 ..
-rw-----@ 1 jrose 234561557 6148 Jun 1 14:23 .DS_Store
drwx-----@ 4 jrose 234561557 136 Jun 10 21:55 http_80

code.google.com//http_80:
total 8
drwx-----@ 4 jrose 234561557 136 Jun 10 21:55 .
drwx-----@ 4 jrose 234561557 136 Jun 1 14:23 ..
-rw-----@ 1 jrose 234561557 3072 Jun 10 21:55 database-demo#database
drwx-----@ 6 jrose 234561557 204 Jun 10 21:55 helloworld-store[1]#localhostserver

code.google.com//http_80/helloworld-store[1]#localhostserver:
total 48
drwx-----@ 6 jrose 234561557 204 Jun 10 21:55 .
drwx-----@ 4 jrose 234561557 136 Jun 10 21:55 ..
-rw-----@ 1 jrose 234561557 3392 Jun 10 21:55 gears_init[4].js
-rw-----@ 1 jrose 234561557 4885 Jun 10 21:55 hello_world_resourcestore[1].html
-rw-----@ 1 jrose 234561557 4142 Jun 10 21:55 sample[2].js
-rw-----@ 1 jrose 234561557 1776 Jun 10 21:55 sample[3].css
```

# Client-Side Storage - Silverlight



# Client-Side Storage - TOCTOU

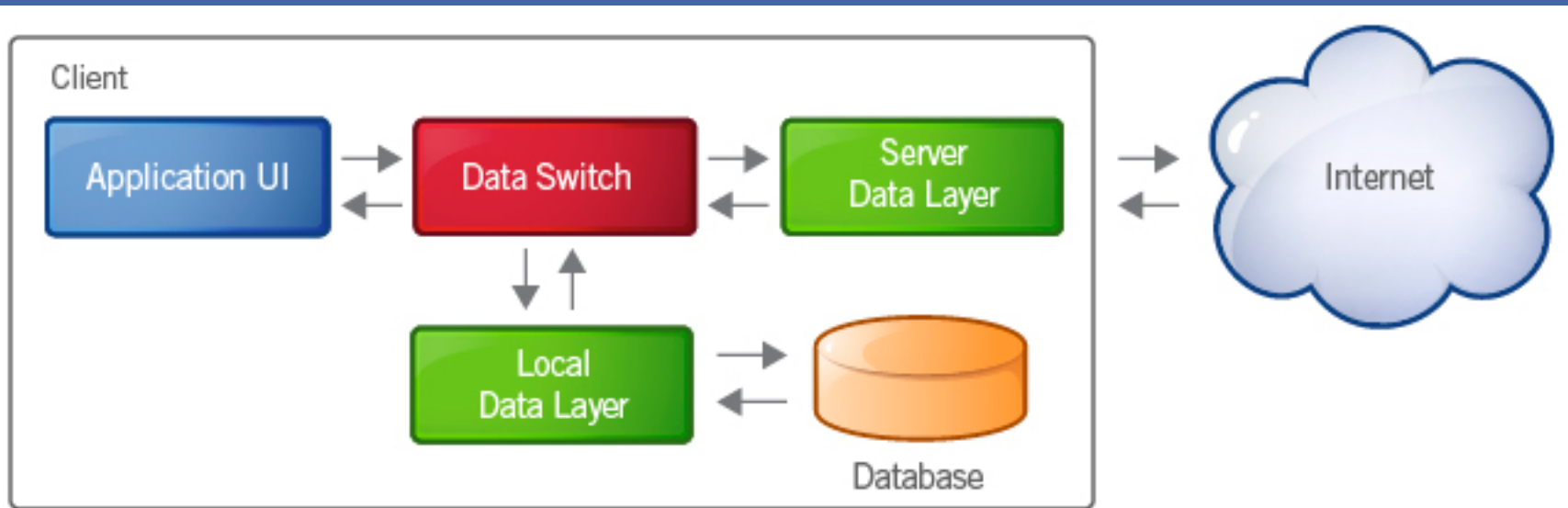


Figure: Local Data Layer

# Debug Functionality

- Often inadvertently left in production code
- “Backdoors” used for testing





# User Supplied Parameters

- Garbage In / Garbage Out

```
<object width="300" height="300"  
  data="data:application/x-silverlight-2,"  
  type="application/x-silverlight-2" >  
  <param name="source" value="SilverlightApplication1.xap"/>  
  document.write("<param name='UA'")  
  document.write(navigator.userAgent + "</>")  
</object>
```

```
private function loadImage(arg0:String)  
{  
  var loc0:* = null;  
  trace("Loading: " + arg0);  
  loc0 = new Loader();  
  addEventListener(Event.COMPLETE, this.handleLoadedImage);  
  loadedList.push(loc0);  
  loc0.alpha = 0;  
  loc0.load(new URLRequest(arg0));  
  return;
```

# Insecure Network Communication





# Cross Domain Access

## Client Access Policy (Silverlight)

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
  <cross-domain-access>
    <policy>
      <allow-from http-request-headers="*">
        <domain uri="http://contoso.com"/>
      </allow-from>
      <grant-to>
        <resource path="/" include-subpaths="true"/>
      </grant-to>
    </policy>
  </cross-domain-access>
</access-policy>
```

## Cross Domain Access Gears Workers

```
var workerPool = google.gears.workerPool;
// enable cross domain communication
workerPool.allowCrossOrigin();
// set up listener
workerPool.onmessage = function(a, b, message) {
  //Code to handle callers message
  return;
}
// listen for requests
processRequest(message);
}
```

## Cross Domain Policy File (Flex)

```
<xml version="1.0">
<!-- http://www.trustwave.com/crossdomain.xml-->
<cross-domain-policy>
<allow-access-from domain="*.trustwave.com"/>
<allow-access-from domain="trustwave.com" to-ports="4,8,15,16,23,42" />
</cross-domain-policy>
```

## Cross Domain Policy File (Silverlight)

```
<xml version="1.0">
<cross-domain-policy>
  <allow-http-request-headers-from domain="*" headers="*" />
</cross-domain-policy>
```

# Security Guidance / Best Practices

- Don't store sensitive info on clients
- Enforce SSL connections
- Implement authentication and authorization for remote server methods
- Keep developers trained in secure application development best practices.
- Perform security testing of all applications
- Be careful with parameters – anything that can be supplied by the user...

# riaEnumerator.py

- W3af plugin for identifying Rich Internet Applications
- Currently supports:
  - Google gears manifests
  - Flex crossdomain.xml
  - Silverlight clientaccesspolicy.xml



# Questions & Comments

- Additional Resources:

- Trustwave Spiderlabs: [trustwave.com/spiderLabs.php](http://trustwave.com/spiderLabs.php)
- W3AF RIA plugin: [riaEnumerator.py](http://riaEnumerator.py)
- Deblaze: [Deblaze-tool.appspot.com](http://Deblaze-tool.appspot.com)

- Contact us:

- [jrose@trustwave.com](mailto:jrose@trustwave.com)
- [kstadmeyer@trustwave.com](mailto:kstadmeyer@trustwave.com)