Riscos de RIA (Rich Internet Applications)

Kevin Stadmeyer

Jon Rose

Who are we?

SpiderLabs℠
A division of AmbironTrustWave

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

Trustwave®
Information Security & Compliance

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

Trustwave®
Information Security & Compliance

Security Controls

- Sandboxing
- Same Origin Policy

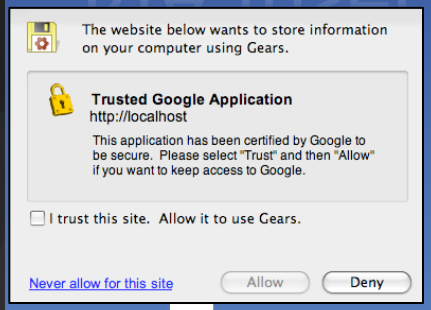
Trustwave®
Information Security & Compliance

Why Attack RIA

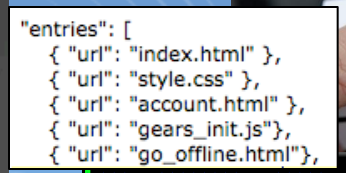
- Easy targets
  - Weak security controls
  - insecure development practices
  - Inexperienced developers
- Lots of money to be made
  - Credit Cards
  - Identity theft
  - Blackmail

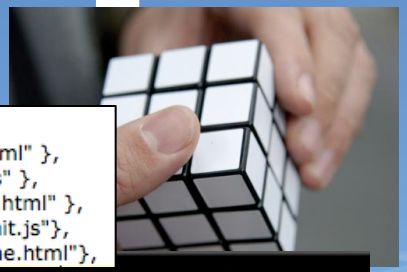
Trustwave®
Information Security & Compliance

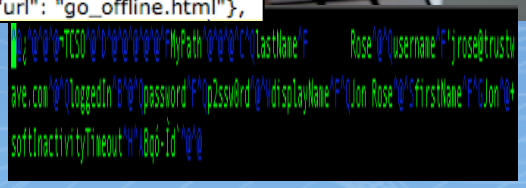
RIA Insecure Lifecycle



Installation → Client security → Networking → Remoting → Uninstall





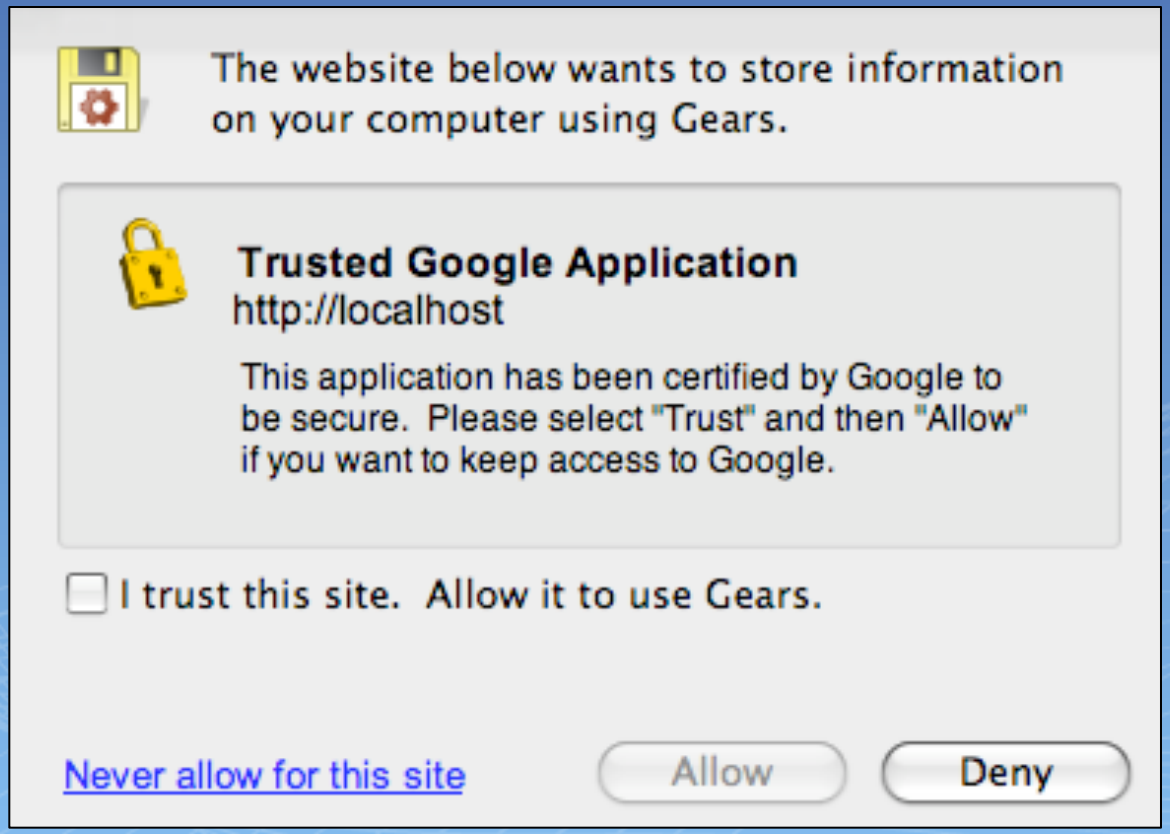


```
private function loadImage(arg0:String)
{
    var loc0:* = null;
    trace("Loading: " + arg0);
    loc0 = new Loader();
    addEventListener(Event.COMPLETE, this.handleLoadedImage);
    loadedList.push(loc0);
    loc0.alpha = 0;
    loc0.load(new URLRequest(arg0));
    return;
```

Trustwave®
Information Security & Compliance

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.' );
```

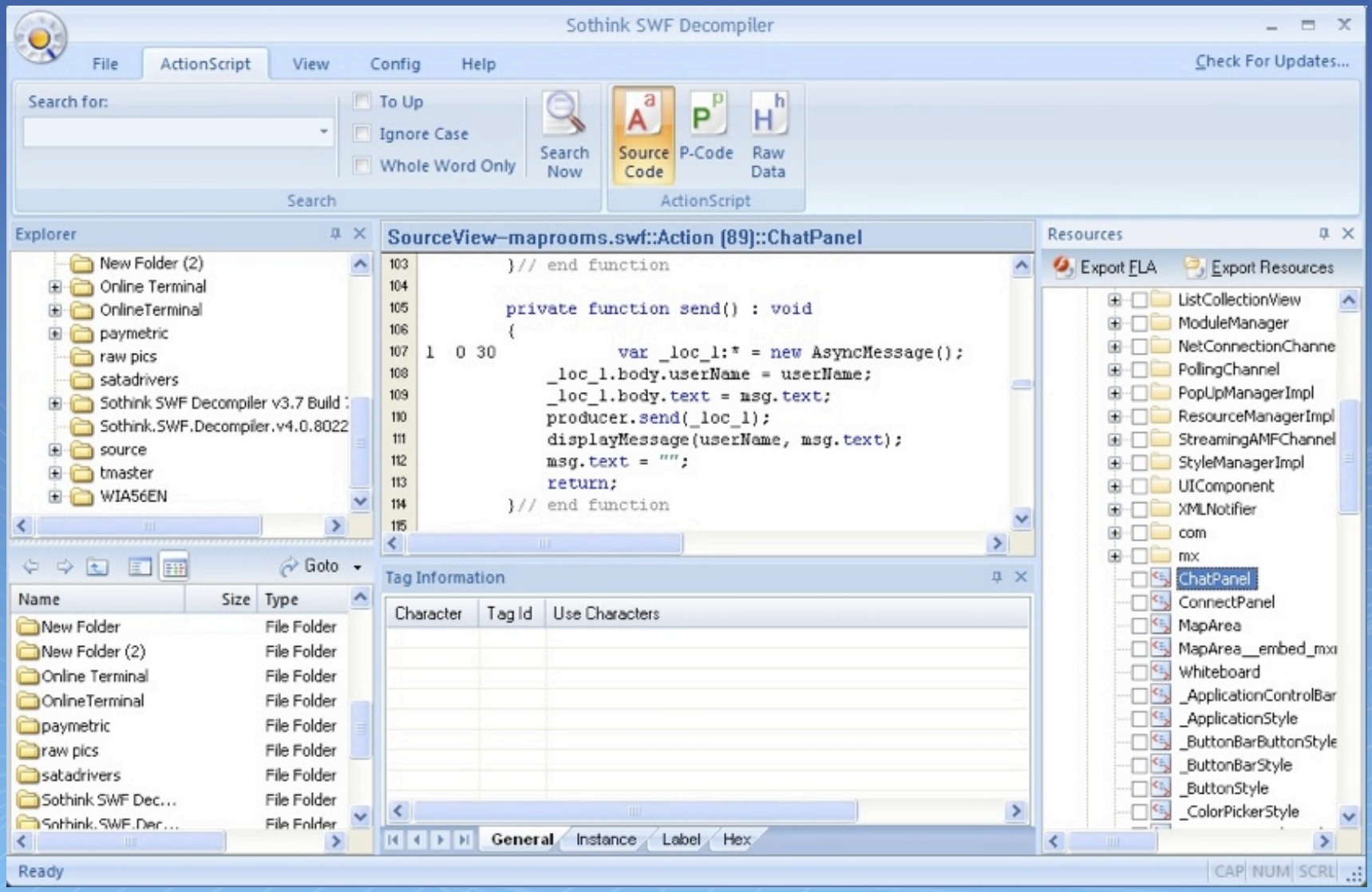


Trustwave®
Information Security & Compliance

Client Decompilaton / Reverse Engineering

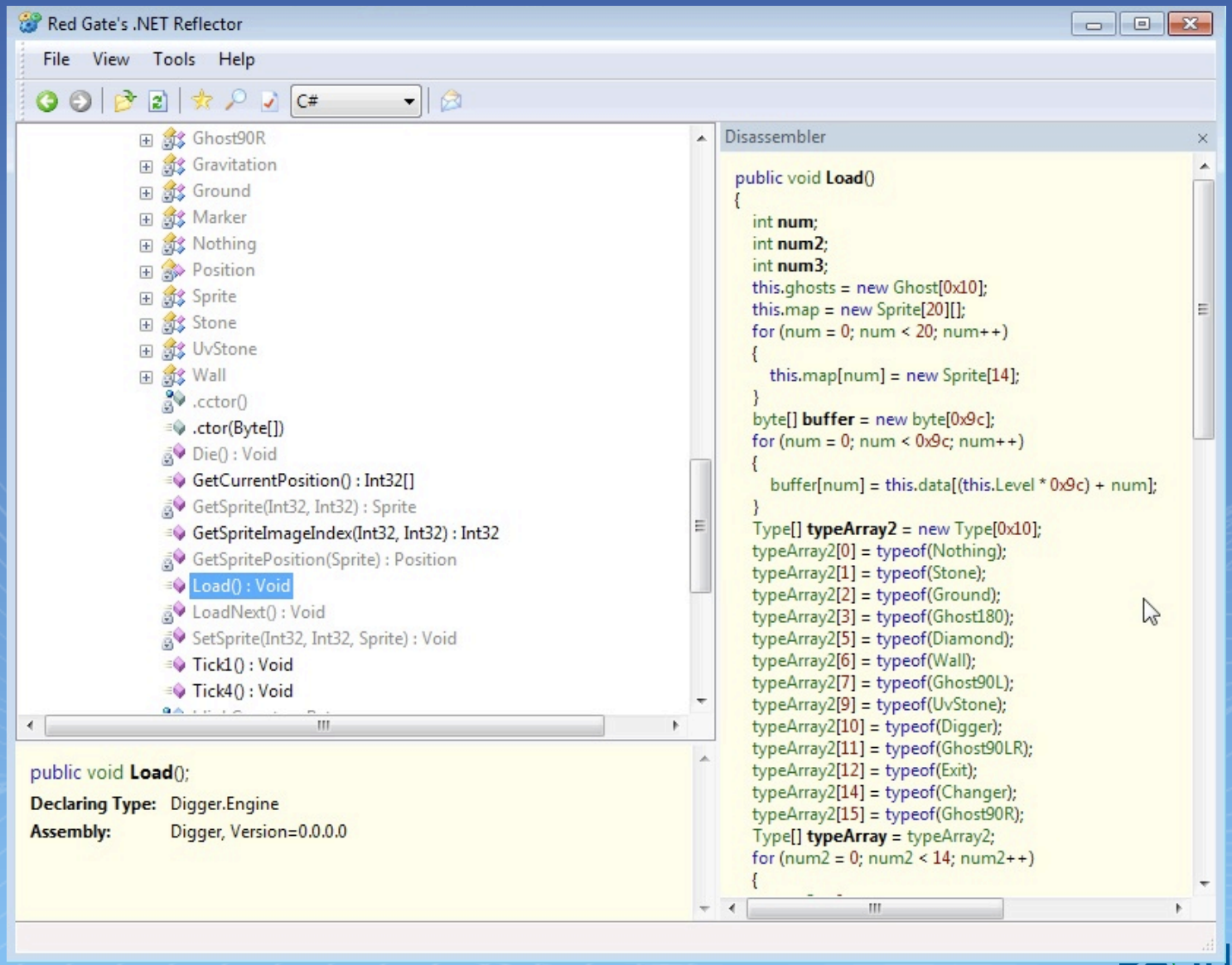
Trustwave®
Information Security & Compliance

Flex Decompilation



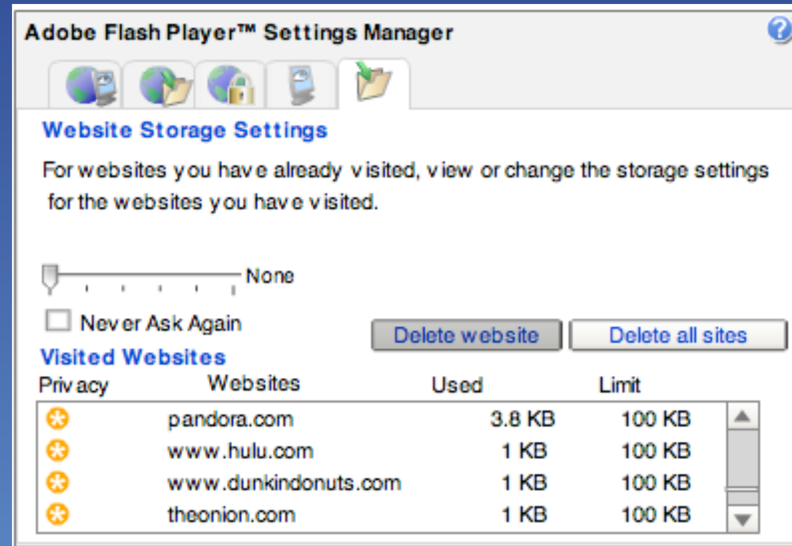
Trustwave®
Information Security & Compliance

Silverlight/.Net Decompilation



Client-Side Storage

- Flash Cookies

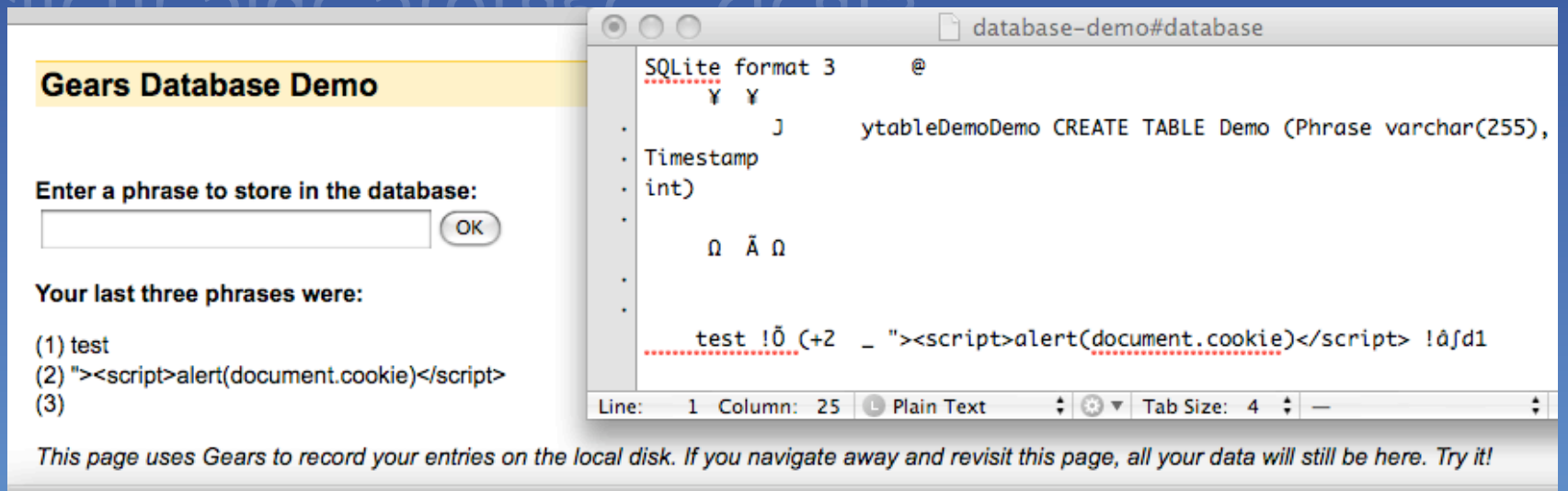


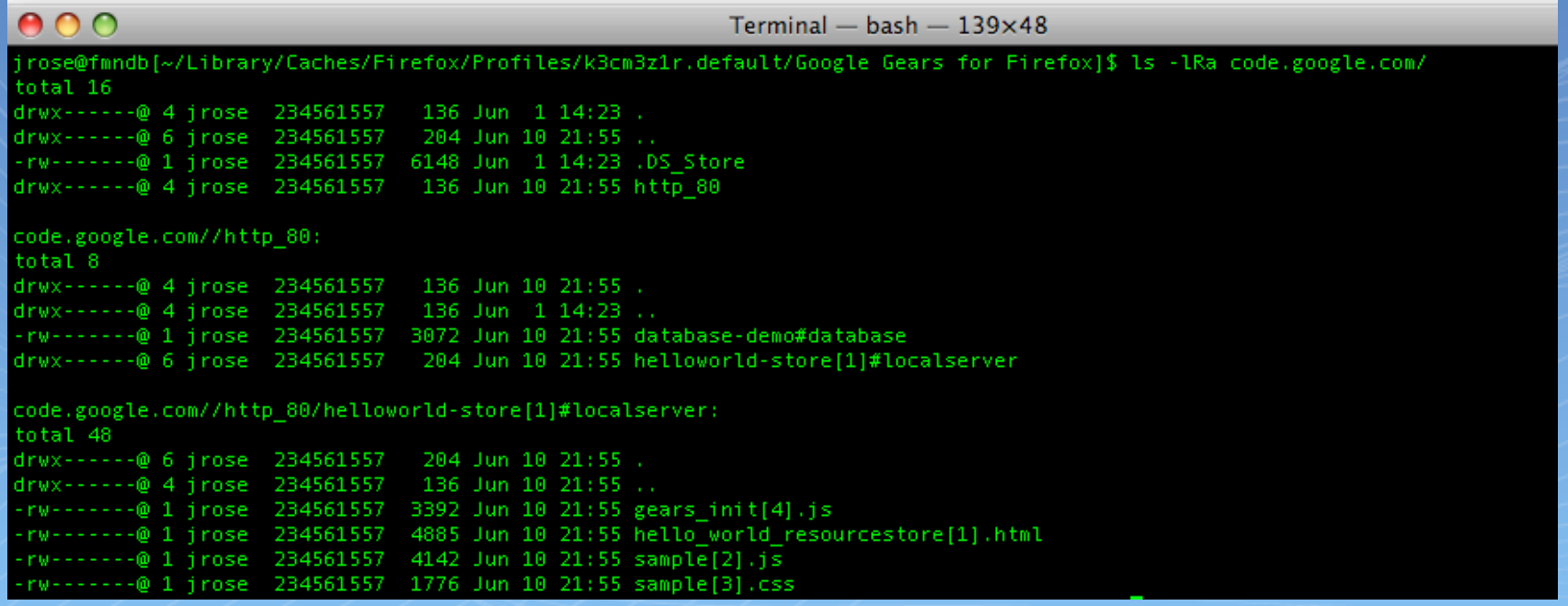
^@¿^@^@^@¬TCSO^@^D^@^@^@^@^(^@^@^@^@^@^@^@^C^QlastName^F Rose^@^Qusername^F'jrose@trustw
ave.com^@^QloggedIn^B^@^Qpassword^F^Qp2ssw0rd^@^WdisplayName^F^QJon Rose^@^SfirstName^F^GJon^@+
softInactivityTimeout^H^ABqó-Íd`^@^@

- Google Gears Local DB and local web server cache

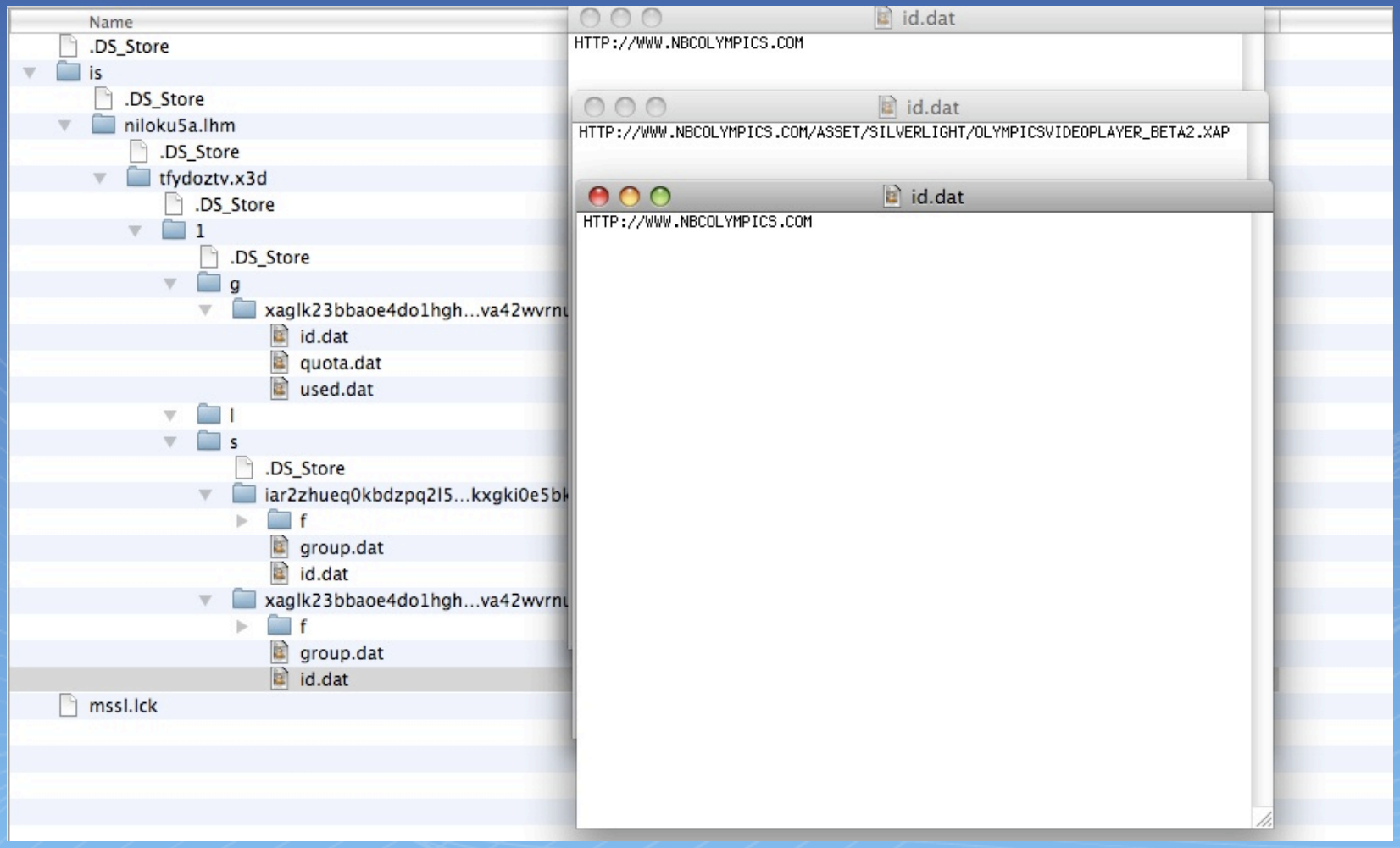

```
"entries": [
   { "url": "index.html" },
   { "url": "style.css" },
   { "url": "account.html" },
   { "url": "gears_init.js"},
   { "url": "go_offline.html"},
```

Client-Side Storage - Gears





Client-Side Storage - Silverlight



Trustwave®
Information Security & Compliance

Client-Side Storage - TOCTOU

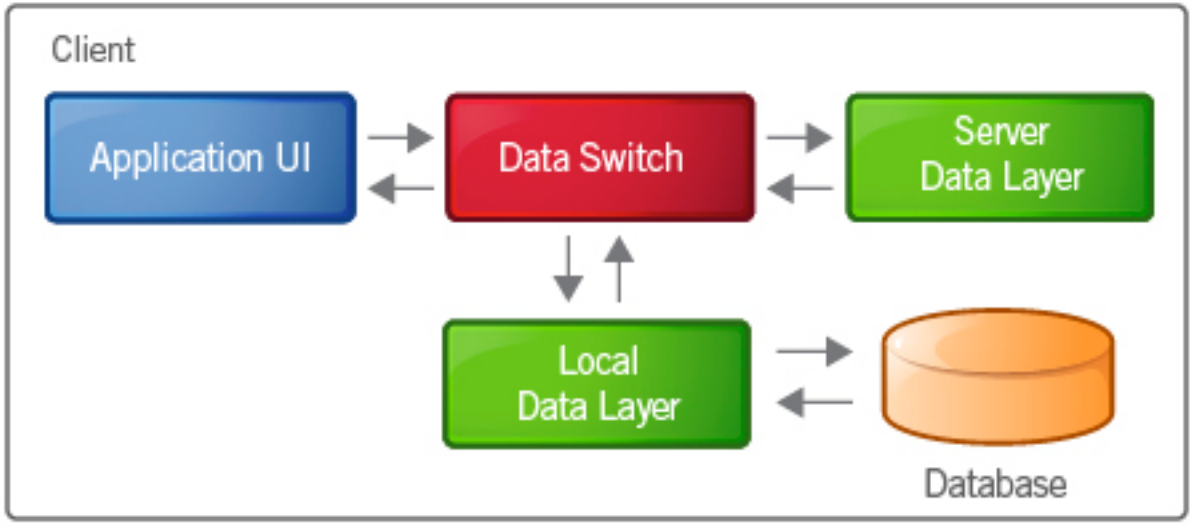


Figure: Local Data Layer

Source: google.com

Debug Functionality

- Often inadvertently left in production code
- "Backdoors" used for testing

Trustwave®
Information Security & Compliance

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

Trustwave®
Information Security & Compliance

Remoting Insecurity

- Developers fail to restrict access to methods:
  - Authentication
  - Authorization
- Deblaze - Servers can be fingerprinted, method & service names can be brute-forced

```
jrose@fmndb[~/bluedog]$ python2.4 deBlaze.py --url="http://                    messagebroker/amf"
--method="resetUserPassword" --params="kstadmeyer@trustwave.com,password1"
      .___    ___________.__
   __| _/____\_____   \  | _____  ________ ____
  / __ |/ __ \|    |  _/  | \__  \ \___   // __ \
 / /_/ \  ___/|    |   \  |__/ __ \_/    /\  ___/
 \____ |\___  >______  /____(____  /_____ \\___  >
      \/    \/       \/          \/      \/    \/
URL: http://                    messagebroker/amf endpoint
Service:
Method: resetUserPassword
Params: kstadmeyer@trustwave.com,password1
['kstadmeyer@trustwave.com', 'password1']
None
<type 'NoneType'>
```

Trustwave®
Information Security & Compliance

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>
```

Trustwave®
Information Security & Compliance

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

Trustwave®
Information Security & Compliance

Questions & Comments

- Additional Resources:
  - Trustwave Spiderlabs: trustwave.com/spiderLabs.php
  - W3AF RIA plugin: riaEnumerator.py
  - Deblaze: Deblaze-tool.appspot.com

- Contact us:
  - jrose@trustwave.com
  - kstadmeyer@trustwave.com

Trustwave®
Information Security & Compliance