Testing and Fuzzing FLEX: More fun with RIA

Christophe De La Fuente
Outline

1. What is Flex?
2. Connecting to back-end
3. Remote Object & AMF
4. What about Security?
5. Testing Flex
What is Flex?
Flex?
... or maybe?
Flex framework

Free & open source
Rich Internet Application
Browser (Adobe Flash Player)
Standalone (Adobe AIR®)
MXML (XML-based language)
ActionScript® (OO language)
<?xml version="1.0" encoding="utf-8"?>
  layout="horizontal"
  creationComplete="srv.getProducts()">

  <mx:RemoteObject id="srv"
    destination="product"/>

  <mx:Panel title="Catalog"
    width="100%" height="100%">
    <mx:DataGrid id="list"
      dataProvider="{srv.getProducts.lastResult}"
      width="100%" height="100%"/>
  </mx:Panel>

  <ProductForm product="{Product(list.selectedItem)}"/>

</mx:Application>
package events {

import flash.events.Event;

public class ShowPreview extends Event {

    public var employeeInfo:Object;
    public var message:String;

    public function ShowPreview(type:String, employeeInfo:Object, message:String) {
        super(type);
        this.employeeInfo = employeeInfo;
        this.message = message;
    }

    override public function clone() : Event {
        return new ShowPreview(type, employeeInfo, message);
    }
}
}
Compilation

Flash Builder (Eclipse plug-in)

Command line
• mxmlc (MXML) → SWF
• compc (ActionScript component compiler) → SWC

ABC (ActionScript Bytecode)

Executed in a Virtual Machine (AVM2)

Easy decompilation
Flex 4

Requires Flash Player 10 support

UI improvements
• new skinning & component architecture (Spark)
• 2D & 3D improvements
• support for FXG (vector-based graphics)
• effects and transitions
Connecting to back-end
Communication

URL Loader

ActionScript Binary Socket

XML Socket

Remote Object
  • invoke server objects
  • serialization

HTTP Service
  • any HTTP method

Web Service
Servers

BlazeDS
- remoting and web messaging
- connect to back-end

LiveCycle Data Services ES2
- infrastructure for enterprise Flex and AIR applications
BlazeDS

Remoting
- RPC calls
- call ColdFusion or Java classes
- mapping of Value Object types

Messaging
- publish/subscribe
- asynchronous communication
- producers and consumers
- clients ↔ servers

Adobe Flex + BlazeDS
BlazeDS - Components

Destination
• server-side class
• message topic
• proxy

Channel
• communication protocol (amf, amf-secure, http,...)
• multiple channels per service

Service Adapters
• integrate with the back-end
• many provided
• customizable
Summarizing…

http://flex.org/what-is-flex
Remote Object & AMF
AMF

AMF0/3
Open source
Compact binary format
Object serialization
One byte marker
Object by value or by Reference
Implementations (ruby, python, java,...)
Remoting with BlazeDS

DEMO
What about Security?
What about security?

Large attack surface

- Channels
- Adapters
- Back-end components

All public method accessible by default
Same-Domain Policy

Enforced by Flash Player from version 7

Access resources from the domain it was loaded from

Granting access

• Security.allowDomain() method
• URL/socket policy file (crossdomain.xml)

```xml
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
  <cross-domain-policy>
    <allow-access-from domain="*.trustwave.com" to-ports="507,516"/>
  </cross-domain-policy>
```
Sanboxing

Assets put in the same security sandbox

Different rules for:
  • loading content (images, HTML, SWF,...)
  • accessing data (accessible only to code)

Remote Sandbox
  • cannot load any local files or resources

Local Sandbox
  • local-with-filesystem
  • local-with-networking
  • local-trusted
  • AIR application sandbox
Permission Controls

Administrator (User Institution) settings

The mms.cfg file
The Global Flash Player Trust directory

User settings

Settings UI / Manager
User Flash Player Trust directory

Website settings

Policy file (crossdomain.xml)

Author settings

Security.allowDomain("www.example.com")
Other Security Features

Disk space constrained (100k)

Outbound port filter
  • commonly reserved ports is blocked (21, 22,...)

BlazeDS Security
  • authentication (custom/built-in)
  • authorization (role-based) on destinations and methods
Testing Flex
Cross Domains Attacks

Attacker

SWF(A)

Domain A

Domain B

SWF(A) makes call to Domain B’s resources

crossdomain.xml allow access from Domain A

SWF(Attacker) makes illegal call to Domain B’s resources through SWF(B)
Cross-scripting

“Code defined in one file can examine and modify variables, objects, properties, methods, and so on in the other, and vice versa”

Domain restriction
  • permission granted with Security.allowDomain()

Version restriction
  • Not supported between AVM1 SWF files & AVM2 SWF files

HTML-to-SWF
  • Scripting can occur with callbacks established through the ExternalInterface.addCallback() method
Cross-scripting

DEMO
Tools

**Decompilers**
- swfdump (SWFTools)
- Sothink

**Debuggers**
- fdb
- Flash Player debug build

**Web Proxy**
- BurpSuite
- Charles Proxy

**Eclipse (Flash Builder plug-in)**
Fuzzer / Bruteforce

Standard vulnerabilities

- SQL injection
- XSS (when using ExternalInterface.addCallback() )
- Password brute forcing
- etc...

Tool to call remote objects via AMF

- Deblaze (Jon Rose)
- Integration with Burp Suite and interactivity needed...

... I wrote mine
Flexible Fuzzer

Ruby

RocketAMF (AMF serializer/deserializer)

Burp Suite integration thanks to Buby (Eric Monti)

Will be added to the OWASP Web Testing Environment (WTE)
Flexible Fuzzer

DEMO
Thanks!
Questions?