

AMF Testing Made Easy!

DeepSec 2012

Luca Carettoni

Agenda

- ✱ **AMF specification, BlazeDS, current techniques and tools**
- ✱ **Blazer**
 - ✱ **architecture, core techniques, heuristics**
- ✱ **Testing with Blazer**
 - ✱ **Objects generation and fuzzing *DEMO***
- ✱ **CVE-2012-3249, Fortify Privileged Information Disclosure**
- ✱ **Finding vulnerabilities with Blazer**
 - ✱ **Unauthenticated methods *DEMO***
 - ✱ **SQL Injection *DEMO***
- ✱ **What's new in Blazer v0.3**
- ✱ **Conclusion**

Thanks!

- * **Matasano Security** - <http://matasano.com/>
 - * Part of this research was performed on behalf of Matasano Security
- * **Dafydd Stuttard** - <http://www.portswigger.net/>
 - * Burp, such an amazing tool

I am a doer. And you?

- ✱ Luca Caretoni - luca@addepar.com
- ✱ Reinventing the Infrastructure that Powers Global Wealth Management - <http://addepar.com>



Introduction and context

- ✱ **Adobe Flex**

- ✱ Framework for building Rich-Internet-Applications
- ✱ Based on Adobe Flash

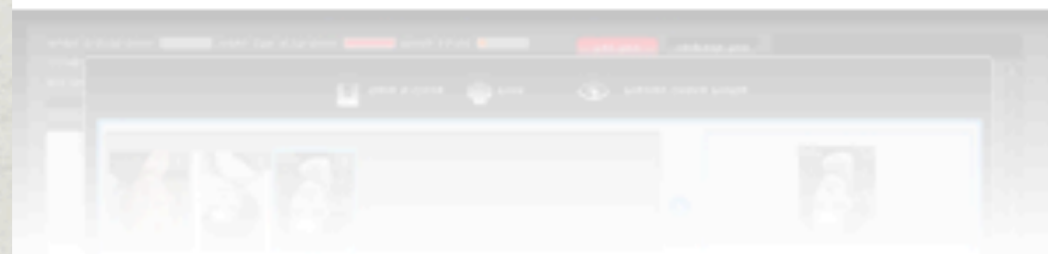
- ✱ **ActionScript**

- ✱ ActionScript is an object-oriented programming language

- ✱ **Action Message Format (AMF)**

- ✱ Introduced with Flash Player 6
- ✱ Compact binary format to serialize ActionScript objects
- ✱ Fast data transfer, comparing to text-based protocols
- ✱ An efficient mechanism to:
 - ✱ Save and retrieve application resources
 - ✱ Exchange strongly typed data between client-server

AMF for end-users



AMF for old-school hackers

```
0x0230: 0d0a 0003 0000 0001 0004 6e75 6c6c 0003 .....null..
0x0240: 2f34 3200 0002 220a 0000 0001 110a 8113 /42...".....
0x0250: 4f66 6c65 782e 6d65 7373 6167 696e 672e Oflex.messaging.
0x0260: 6d65 7373 6167 6573 2e52 656d 6f74 696e messages.Remotin
0x0270: 674d 6573 7361 6765 0d73 6f75 7263 6513 gMessage.source.
0x0280: 6f70 6572 6174 696f 6e09 626f 6479 1163 operation.body.c
0x0290: 6c69 656e 7449 6417 6465 7374 696e 6174 lientId.destinat
0x02a0: 696f 6e0f 6865 6164 6572 7313 6d65 7373 ion.headers.mess
0x02b0: 6167 6549 6415 7469 6d65 546f 4c69 7665 ageId.timeToLive
0x02c0: 1374 696d 6573 7461 6d70 0106 1b67 6574 .timestamp...get
0x02d0: 4174 7472 6962 7574 6573 0905 0106 8231 Attributes.....1
0x02e0: 666c 6578 2e72 756e 7469 6d65 2e42 6c61 flex.runtime.Bla
0x02f0: 7a65 4453 3a48 5454 5050 726f 7879 5365 zeDS:HTTPProxySe
0x0300: 7276 6963 653d 7072 6f78 792d 7365 7276 rvice=proxy-serv
0x0310: 6963 652c 4d65 7373 6167 6542 726f 6b65 ice,MessageBroke
0x0320: 723d 4d65 7373 6167 6542 726f 6b65 7231 r=MessageBroker1
0x0330: 2c69 643d 4465 6661 756c 7448 5454 502c ,id=DefaultHTTP,
0x0340: 7479 7065 3d4d 6573 7361 6765 4272 6f6b type=MessageBrok
0x0350: 6572 2e48 5454 5050 726f 7879 5365 7276 er.HTTPProxyServ
0x0360: 6963 652e 4854 5450 5072 6f78 7944 6573 ice.HTTPProxyDes
0x0370: 7469 6e61 7469 6f6e 0909 0106 1f49 6e76 tination.....Inv
0x0380: 6f6b 6548 5454 5043 6f75 6e74 061f 496e okeHTTPCount..In
0x0390: 766f 6b65 534f 4150 436f 756e 7406 2749 vokeSOAPCount.'I
0x03a0: 6e76 6f6b 6548 5454 5046 7265 7175 656e nvokeHTTPFrequen
0x03b0: 6379 0627 496e 766f 6b65 534f 4150 4672 cy.'InvokeSOAPFr
0x03c0: 6571 7565 6e63 7906 4943 3443 3936 4541 equency.IC4C96EA
0x03d0: 392d 3944 3039 2d33 4243 332d 3238 4532 9-9D09-3BC3-28E2
0x03e0: 2d35 3233 3539 3734 3344 3735 3706 2352 -52359743D757.#R
0x03f0: 756e 7469 6d65 4d61 6e61 6765 6d65 6e74 untimeManagement
0x0400: 0a0b 0109 4453 4964 0649 4334 4339 3634 ....DSId.IC4C964
0x0410: 4535 2d39 4431 382d 3637 4435 2d31 4133 E5-9D18-67D5-1A3
0x0420: 442d 4132 3144 3642 4534 3933 3535 1544 D-A21D68E49355.D
0x0430: 5345 6e64 706f 696e 7406 0761 6d66 0106 SEndpoint..amf..
0x0440: 4941 3739 3032 3034 452d 3842 3538 2d30 IA790204E-8B58-0
0x0450: 3244 452d 3432 3838 2d36 4531 3931 3534 2DE-4288-6E19154
0x0460: 4634 4636 3404 0004 00 F4F64....
```

AMF for web hackers

Burp Intruder Repeater Window About

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Options Alerts

Intercept Options History

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Modified	Status	Length	MIME type	Ext
1	http://127.0.0.1:8400	GET	/ds-console/			304	123		
2	http://127.0.0.1:8400	GET	/ds-console/history/history.js			304	124	script	js
3	http://127.0.0.1:8400	GET	/ds-console/swfobject.js			304	124	script	js
5	http://127.0.0.1:8400	GET	/ds-console/console.swf			304	125	flash	swf
6	http://www.adobe.com	GET	/images/shared/download_buttons/g...			301	672	HTML	gif
8	http://127.0.0.1:8400	POST	/ds-console/messagebroker/amf			200	393	AMF	
9	http://127.0.0.1:8400	POST	/ds-console/messagebroker/amf			200	40809	AMF	
10	http://127.0.0.1:8400	POST	/ds-console/messagebroker/amf			200	919		
11	http://127.0.0.1:8400	POST	/ds-console/messagebroker/amf			200	71565		
12	http://127.0.0.1:8400	POST	/ds-console/messagebroker/amf			200	1730		
13	http://127.0.0.1:8400	POST	/ds-console/messagebroker/amf			200	457		
14	http://127.0.0.1:8400	POST	/ds-console/messagebroker/amf			200	1766		
15	http://127.0.0.1:8400	POST	/ds-console/messagebroker/amf			200	664		
16	http://127.0.0.1:8400	POST	/ds-console/messagebroker/amf			200	470	AMF	
18	http://127.0.0.1:8400	GET	/ds-console/			304	123		

Request Response

Raw Params Headers Hex AMF

	Type	Value
body		
a target	string	null
a response	string	/2
a response method	string	null
[] data	array	
[0]	RemotingMessage	
Source	null	
[] Body	array	
a Operation	string	getFlexMBeanObjectNames
RemoteUsername	null	
RemotePassword	null	
1 Timestamp	number	0
→ Headers	map	
a DSId	string	C48FF0DF-3F18-CF9B-C992-1EDE5935B23F
a DSEndpoint	string	amf
1 TimeToLive	number	0
a Destination	string	RuntimeManagement
ClientId	null	
a Messageld	string	23A4CAC9-A1A5-A30E-A612-6E147317E42D

AMFv0 versus AMFv3

- ✱ **Flash Player 6**

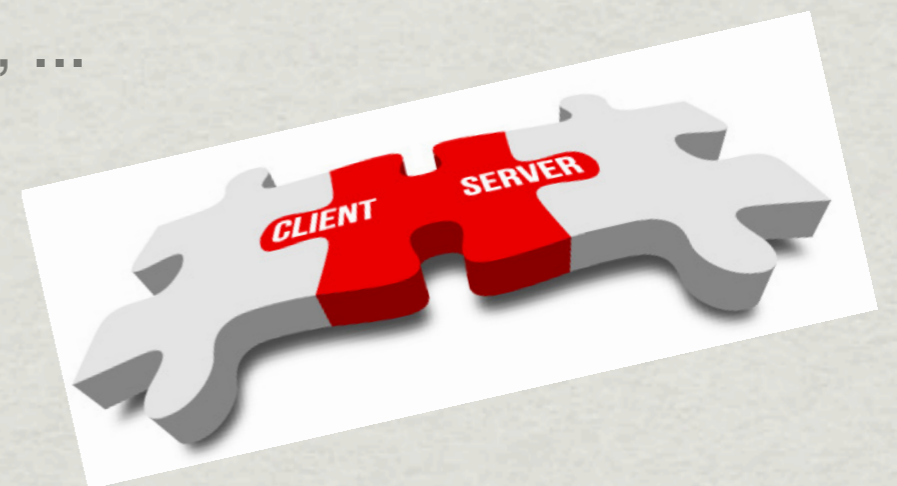
- ✱ Object instances can be sent by reference
- ✱ Support for ActionScript 1.0

- ✱ **Flash Player 9**

- ✱ Object instances, traits and strings can be sent by reference
- ✱ Support for new ActionScript 3.0 data types
- ✱ Support for *flash.utils.IExternalizable*
- ✱ Variable length encoding scheme for integers

Adobe BlazeDS

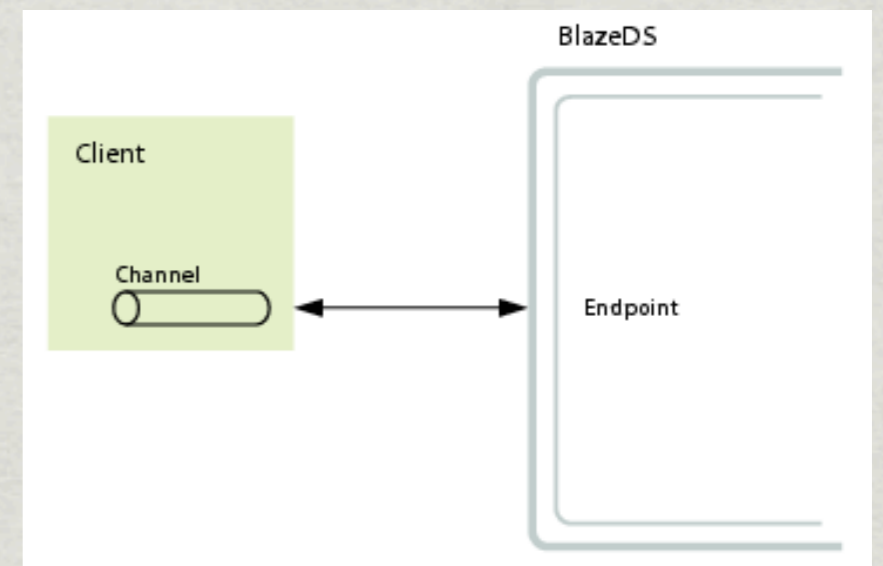
- * **Server-side Java Remoting/Messaging technology**
- * Using Flex Remoting, any Flex client or AIR application can communicate with remote services and inter-exchange data
- * In practice, clients invoke Java methods from classes deployed within a traditional J2EE application server (e.g. Apache Tomcat)
- * A widely deployed implementation
- * Multiple alternatives exist:
 - * **Java:** Adobe LiveCycle Data Service, Granite, ...
 - * **Others:** RubyAMF, FluorineFX, amfPHP, ...



Action Message Format (AMF)

- * **AMF request/response types:**
 - * CommandMessage
 - * **RemotingMessage**
 - *

- * Client-Server communication through channels:
 - * **Endpoint** - `http://<host>/messagebroker/amf`
 - * **Destination Service** - `echoService`
 - * **Operation** - `String echo(String input)`



State of art (research, tools)

- * Testing Flash Applications, OWASP AppSec 2007 - Stefano di Paola
- * Flex, AMF3 And Blazeds - An Assessment, Blackhat USA 2008 - Jacob Karlson and Kevin Stadmeyer
- * Deblaze, Defcon 17 - Jon Rose
- * Pentesting Adobe Flex Applications, OWASP NY 2010 - Marcin Wielgoszewski
- * Starting from v1.2.124, Burp Suite allows to visualize and tamper AMF traffic
- * Other debugging tools
 - * Charles Proxy, WebScarab, Pinta AIR app, ...

Testing remote methods, today

- ✱ **Traffic inspection and tampering**

- ✱ Using network packet analyzers
- ✱ Using HTTP proxies

- ✱ **Enumeration (black-box testing)**

- ✱ Retrieving endpoints, destinations and operations from the traffic
- ✱ Decompiling the Flex application
- ✱ Brute-forcing endpoint, destination and operation names

Life is pain, highness.

Anyone who tells you differently is
selling something

W. Goldman

Is this the best we can do?

- * Ideal for black-box testing, limited knowledge required
- * Time consuming
- * Requires to invoke all application functionalities
- * What about custom objects?
- * What about “hidden” services?
- * How to ensure coverage?

Enterprise-grade applications

- ✱ Large attack surface
- ✱ Custom externalizable classes
- ✱ I've tested applications with **more than 500** remote invokable methods and **more than 600** custom Java objects

```
Request Response
Raw Params Headers Hex
POST /samples/messagebroker/amf HTTP/1.1
Host: 127.0.0.1:8400
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:13.0) Gecko/20100101 Firefox/3.6
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Proxy-Connection: keep-alive
Cookie: JSESSIONID=ABD7C9C82A4A1CAD9615C63C59580110
Referer: http://127.0.0.1:8400/samples/inventory/inventory.swf/[[DYNAMIC]]/6
Content-type: application/x-amf
Content-Length: 466

null /4æ

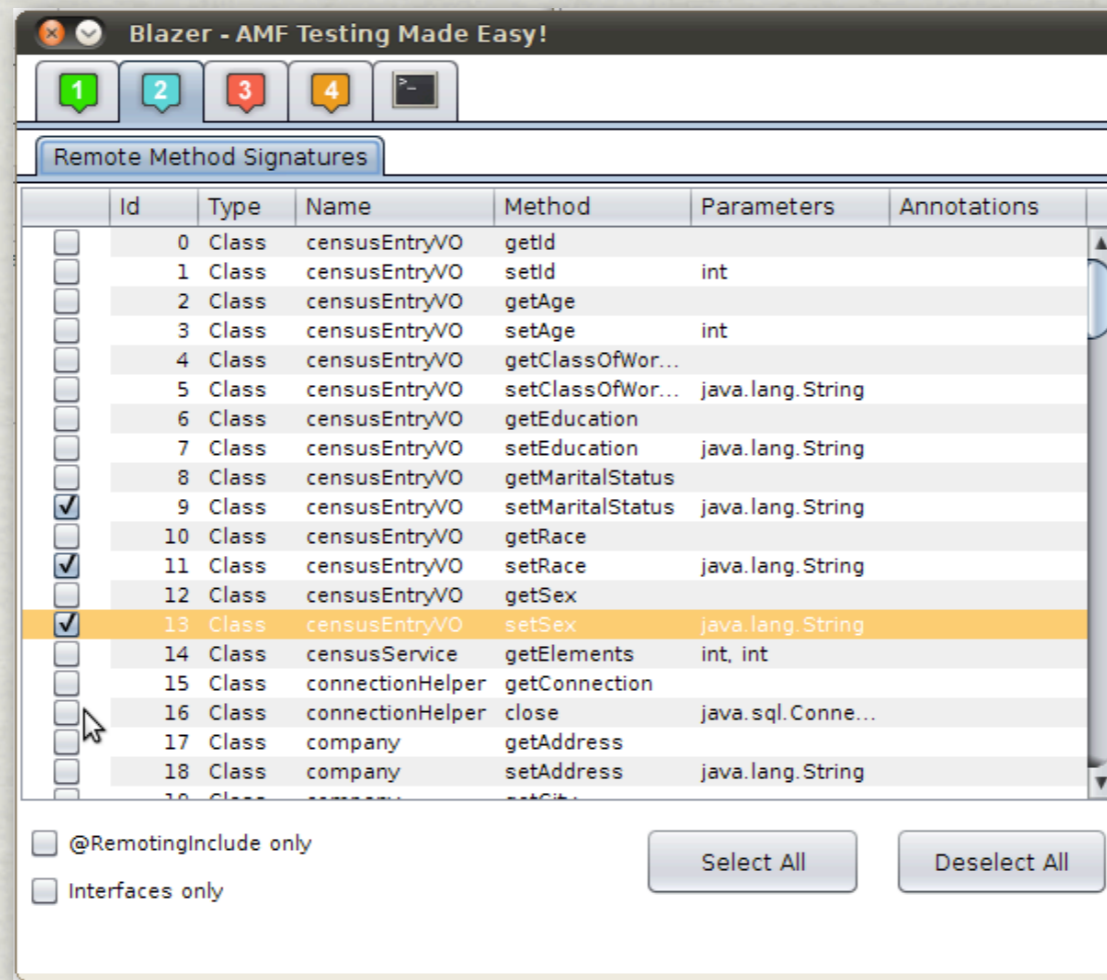
Å Oflex.messaging.messages.RemotingMessage source operation body clientId header
update
s9flex.samples.product.Product price productId category description qtyInStock image
Italy! merlot.jpg Wine ID4D74179-370A-21E5-AD9C-24FB6A2889B8
DSEndpoint my-amf DSid ID4D7373D-9619-76FB-1C1F-87D8A45EA531 I0647C66C-9265-4
```



Life is not #ffffff and #000000

Blazer

- ✦ Custom AMF message generator with fuzzing capabilities
- ✦ Method signatures and Java reflection are used to generate dynamically valid objects



Blazer v0.3 - DeepSec edition

- * **GUI-based Burp Suite plugin**

- * Well-integrated so you won't need to leave your favorite tool

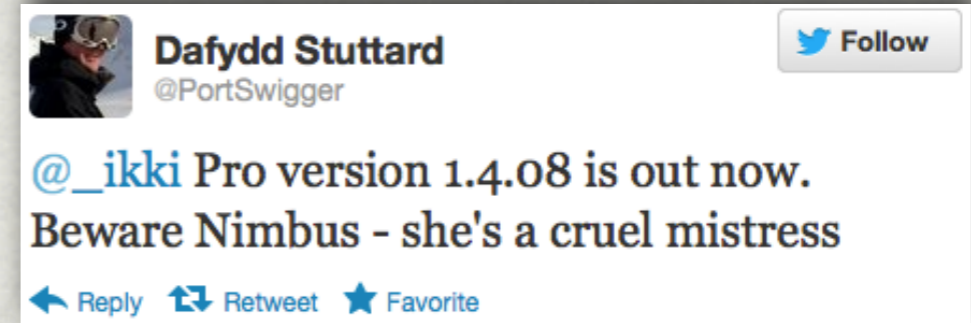
- * Burp Free and Pro

- * With Nimbus look'n'feel too

- * GNU GPL software

- * <http://code.google.com/p/blazer/>

- * Start Burp with `java -classpath Blazer_v0.3.jar:burp.jar burp.StartBurp` and launch Blazer from the context menu



Blazer - Architecture

- ✱ **A packet generator**

- ✱ based on Adobe AMF OpenSource libraries

- ✱ **An object generator**

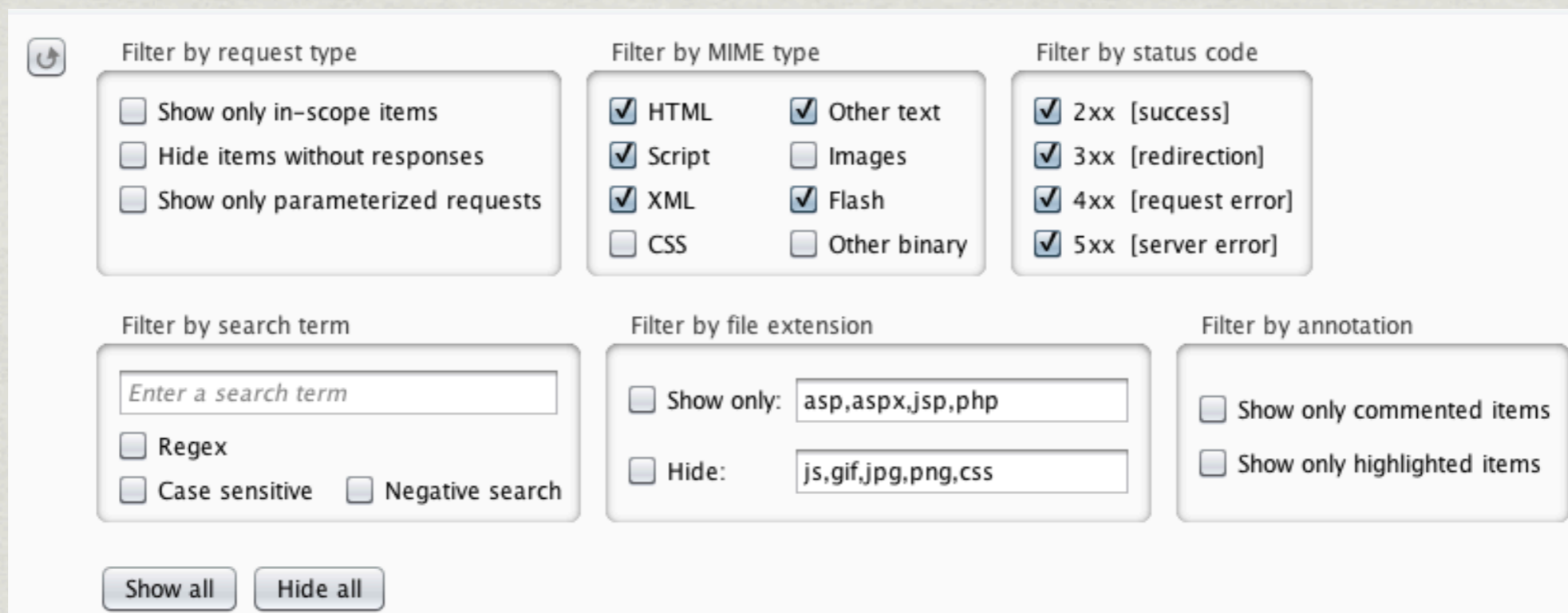
- ✱ to build valid application objects using “best-fit” heuristics

- ✱ **A lightweight fuzzing infrastructure**

- ✱ to generate attack vectors, insert payloads within objects, manage multiple threads and monitor the progress

Blazer as a “custom” AMF client

- * By default, Blazer uses Burp Proxy to record requests and responses
 - * Proxy setting option available
- * Using Burp, you can benefit from all built-in tools available (search, sorting, ...)



The image shows a screenshot of the Burp Suite filter settings interface. It features six filter panels arranged in two rows. The top row includes 'Filter by request type', 'Filter by MIME type', and 'Filter by status code'. The bottom row includes 'Filter by search term', 'Filter by file extension', and 'Filter by annotation'. At the bottom of the interface are 'Show all' and 'Hide all' buttons.

Filter by request type

- Show only in-scope items
- Hide items without responses
- Show only parameterized requests

Filter by MIME type

- HTML
- Script
- XML
- CSS
- Other text
- Images
- Flash
- Other binary

Filter by status code

- 2xx [success]
- 3xx [redirection]
- 4xx [request error]
- 5xx [server error]

Filter by search term

Enter a search term

- Regex
- Case sensitive
- Negative search

Filter by file extension

- Show only: asp,aspx,jsp,php
- Hide: js,gif,jpg,png,css

Filter by annotation

- Show only commented items
- Show only highlighted items

Show all Hide all

It's show time!

- * General usage
- * Objects generation
- * Finding bugs with Blazer: (a) discover exposed methods

CVE-2012-3249

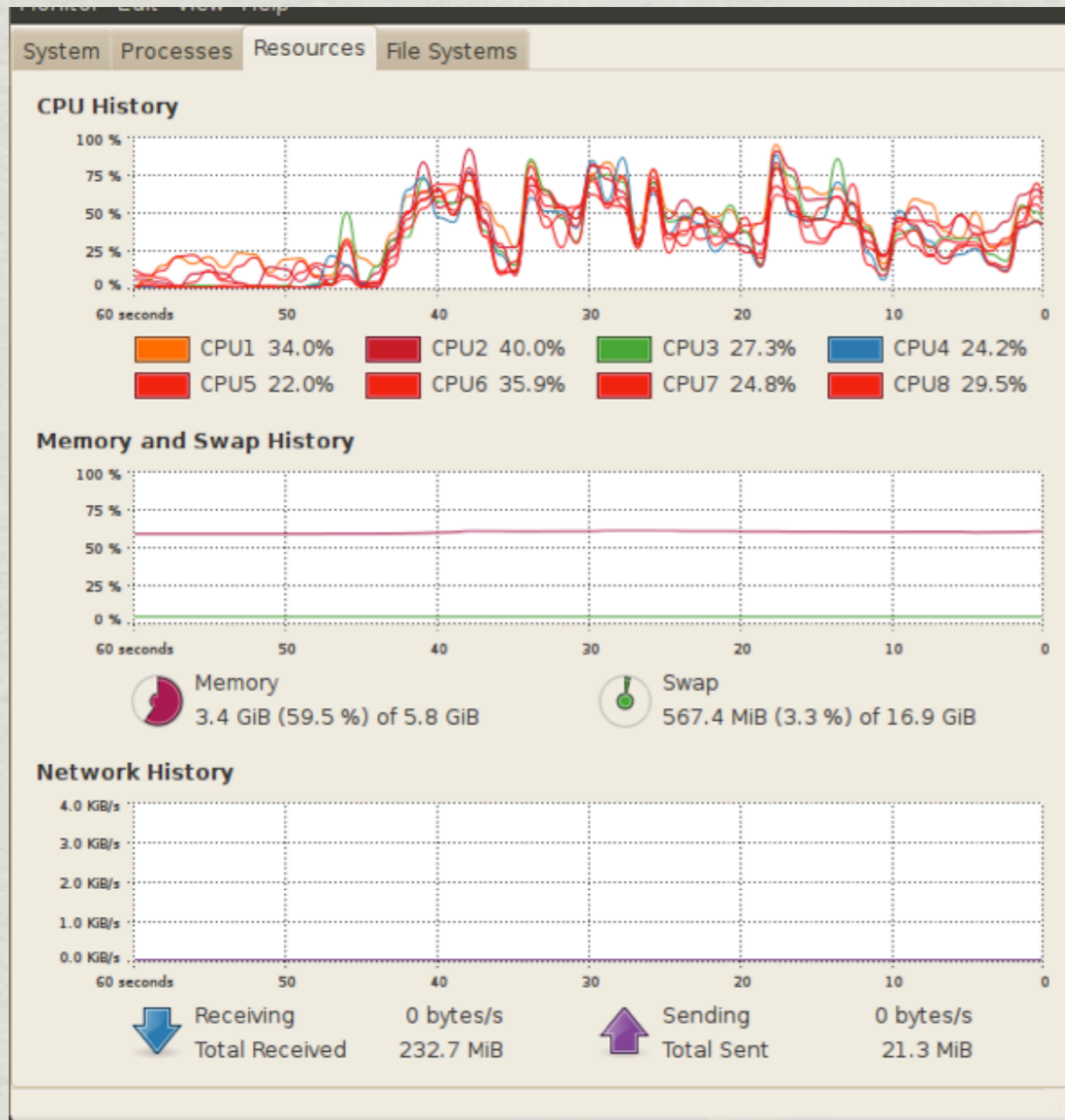
- ✱ **HP Fortify Software Security Center**
 - ✱ Remote Disclosure of Privileged Information
 - ✱ Discovered in June 2012, Patched in August 2012

From the advisory that I sent to HP:

“An AMF endpoint used by the HP Fortify SSC web front-end allows to retrieve sensitive system details, including *user.dir*, *java.vm.name*, *os.name*, *java.vm.vendor*, *version*, *os.version*, *user.home*, *java.runtime.name*, *user.language*, *user.name*, *os.arch*, *java.runtime.version*, *user.country*, *java.version*, ...”

```
public ListResult getFederations(@PName("spec") SearchSpec spec)
```

Testing HP Fortify SSC



Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder

Intercept Options History

Filter: Hiding CSS, image and general binary content

#	Host	Meth...	URL	Para...
1	http://127.0.0.1:8180	POST	/ssc/messagebroker/amf	
2	http://127.0.0.1:8180	POST	/ssc/messagebroker/amf	
3	http://127.0.0.1:8180	POST	/ssc/messagebroker/amf	
4	http://127.0.0.1:8180	POST	/ssc/messagebroker/amf	
5	http://127.0.0.1:8180	POST	/ssc/messagebroker/amf	
6	http://127.0.0.1:8180	POST	/ssc/messagebroker/amf	
7	http://127.0.0.1:8180	POST	/ssc/messagebroker/amf	
8	http://127.0.0.1:8180	POST	/ssc/messagebroker/amf	
9	http://127.0.0.1:8180	POST	/ssc/messagebroker/amf	

Blazer - AMF Testing Made Easy!

1 2 3 4

Status

Method Signatures: **782**
Attack Vectors: **184**
AMF Requests: **1294992**
AMF Requests Sent: **1399**

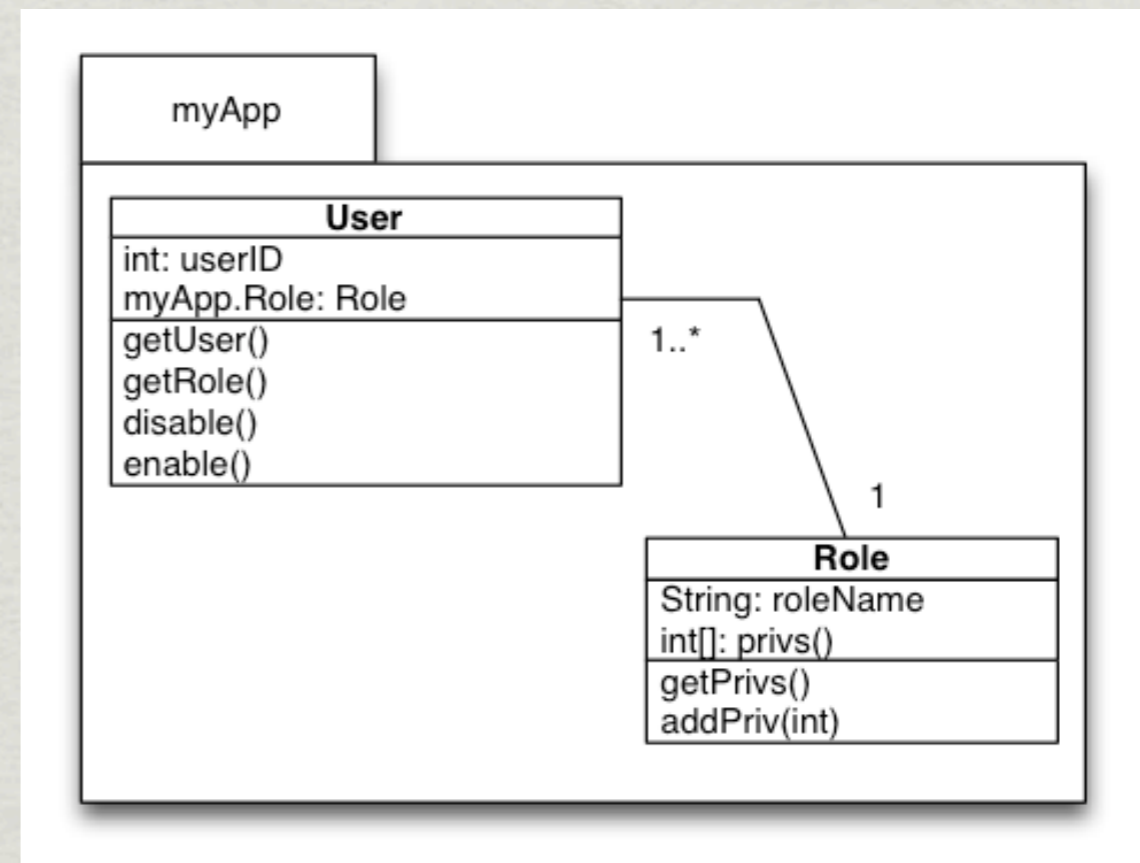
Average Speed (reqs/sec): **29.35314**
Time to Finish (min): **741**
Overall Time (sec): **48**

Current Task: **FUZZING**
Current Status: **RUNNING**

Pause Stop

Blazer - Core techniques

- * **Objects generation**
 - * Java reflection
 - * “Best-fit” heuristics
 - * Randomness and permutations



Blazer - Data pools

- ✱ **Data Pools**

- ✱ Containers for “good” user-supplied input
- ✱ Allow to instantiate objects and invoke methods with semantically valid data
- ✱ Available for all primitive types and String
- ✱ Require to be customized for the target

- ✱ **Attack vectors**

- ✱ Relevant for String objects only
- ✱ Attack vector’s probability allows to unbalance the String data pool with attack vectors

Blazer - Heuristic

∇ Attack vector

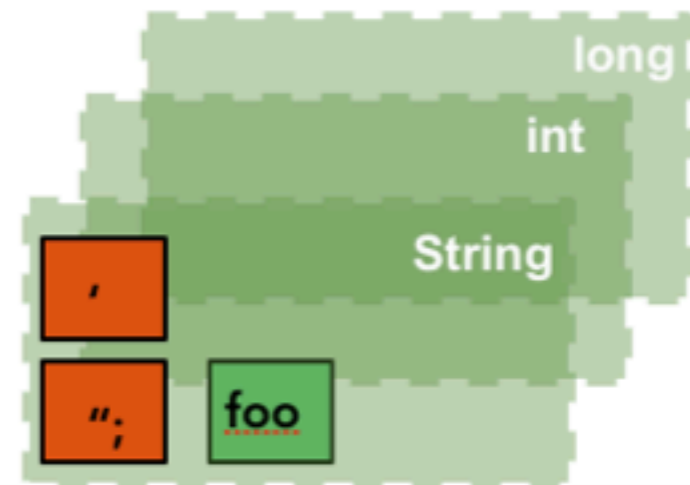
∇ Destination (classes)

∇ Operation (methods)

```
while (numPerm < maxPerm){  
    generateObject(signature);  
    sendObject();  
}
```

Thread

Data Pools



```
Object generate(String signature){  
    if ( int ){  
        getIntFromPool();  
    } else if ( java.lang.String ){  
        getStringFromPool();  
    }  
    ... else {  
        //Build the obj  
        obj = fc.newInstance();  
        //Populate obj using internal methods  
        //Call recursively generate(newSign)  
    }  
}
```

Test case: SQL injection

```
public List getProductsByHash(HashMap paramHashMap)
    throws DAOException
{
46   String str = (String)paramHashMap.get("key");

48   ArrayList localArrayList = new ArrayList();
49   Connection localConnection = null;
    try
    {
52     localConnection = ConnectionHelper.getConnection();
53     PreparedStatement localPreparedStatement = localConnection.prepareStatement("SELECT * FROM product WHERE UPPER(" + str + ")");

55     ResultSet localResultSet = localPreparedStatement.executeQuery();
```

Blazer - “Best-fit” heuristics 1/2

- * For example, let's build a HashMap

```
ObjectGenerator tCObj = new ObjectGenerator(task, null);
```

```
tCObj.generate("java.util.HashMap");
```



Constructor Summary

HashMap()

Constructs an empty `HashMap` with the default initial capacity (16) and the default load factor (0.75).

HashMap(int initialCapacity)

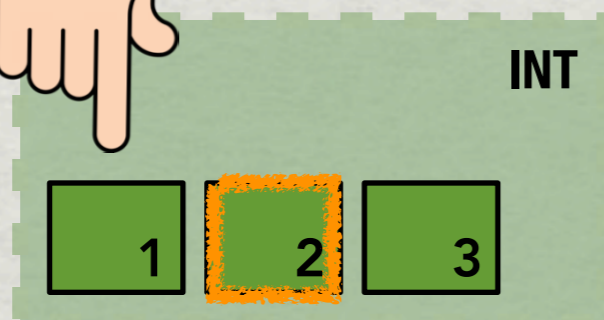
Constructs an empty `HashMap` with the specified initial capacity and the default load factor (0.75).

HashMap(int initialCapacity, float loadFactor)

Constructs an empty `HashMap` with the specified initial capacity and load factor.

HashMap(Map m)

Constructs a new `HashMap` with the same mappings as the specified `Map`.

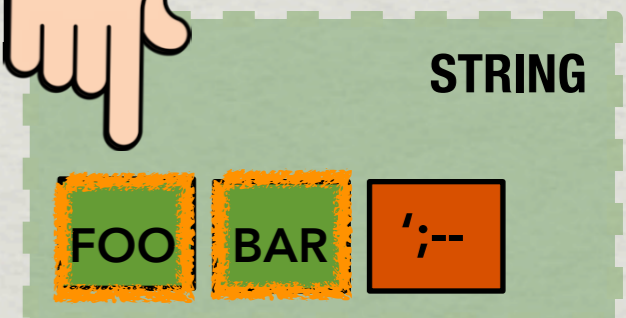


Blazer - "Best-fit" heuristics 2/2

```
{ null, null }  
{ FOO=BAR, null }
```

Method Summary

void	clear() Removes all mappings from this map.
Object	clone() Returns a shallow copy of this <code>HashMap</code> instance: the keys and values themselves are not cloned.
boolean	containsKey(Object key) Returns <code>true</code> if this map contains a mapping for the specified key.
boolean	containsValue(Object value) Returns <code>true</code> if this map maps one or more keys to the specified value.
Set	entrySet() Returns a collection view of the mappings contained in this map.
Object	get(Object key) Returns the value to which the specified key is mapped in this identity hash map, or <code>null</code> if the map contains no mapping for this key.
boolean	isEmpty() Returns <code>true</code> if this map contains no key-value mappings.
Set	keySet() Returns a set view of the keys contained in this map.
Object	put(Object key, Object value) Associates the specified value with the specified key in this map.



It's show time, again!

- * Finding bugs with Blazer: (b) SQL Injection

Coverage and Scalability

- * With unlimited time, you could get theoretically close to 99.9% coverage
- * In practice, **Blazer and target setup are crucial**
 - * Optimize the number of permutations
 - * Balance “good” and “bad” attack vectors
- * Let's do some math:
 - * Application with **~500** exposed operations
 - * **45** attack vectors (Burp's default fuzzing list in Intruder)
 - * **35** permutations (average for big apps, experimentally determined)
 - * $\sim 500 \times 45 \times 35 = \sim \mathbf{787500}$ reqs

So, what's new in Blazer 0.3 ?

- * Import of **classes** and **Java source code**
- * Custom **Java Security Manager** to protect *ObjectGenerator.generate()*
- * Export functionality (**AMF2XML**)

Request in browser
Blazer - AMF Testing
Blazer - AMF2XML Export
Blazer - Enable/Disable SecurityManager

The screenshot shows the Blazer application window titled "Blazer - AMF Testing Made Easy!". The "Export" dialog is open, with the following settings:

- Include:**
 - AMF Request
 - AMF Response
- Output:**
 - Console
 - File

An "Export" button is visible at the bottom of the dialog.

Overlaid on the right is a text editor window titled "ExportExample.txt (~/Desktop) - gedit". The content of the file is as follows:

```
1 -----
2 Blazer v0.3 - AMF2XML Export - November 26, 2012
3 -----
4 Host:127.0.0.1
5 Port:8400
6 Protocol:http
7 Comment:null
8 HTTP Request:
9 <flex.messaging.io.amf.ActionMessage>
10   <version>3</version>
11   <headers/>
12   <bodies>
13     <flex.messaging.io.amf.MessageBody>
14       <targetURI>null</targetURI>
15       <responseURI>/25</responseURI>
16       <data class="object-array">
17         <flex.messaging.messages.RemotingMessage>
18           <clientId class="string">80EF183D-8B08-
19             <destination>product</destination>
20             <messageId>EC66B39C-86F6-4641-7C9B-3BD3
21             <timestamp>0</timestamp>
22             <timeToLive>0</timeToLive>
23           <headers>
```

Conclusions

- ✦ During real-life assessment, the approach has been proven to increase **coverage** and **effectiveness**
- ✦ Blazer was designed to make **AMF testing easy**, and yet allows researchers to control fully the entire security testing process
- ✦ **From 0 to message generation and fuzzing in just few clicks**

- ✦ If you find bugs **using** Blazer, either credits or buy a beer
- ✦ If you find bugs **in** Blazer and provide a patch, I'll buy you a beer

References

- * **AMF 3 Specification, Adobe Systems Inc.**

http://download.macromedia.com/pub/labs/amf/amf3_spec_121207.pdf

- * **Adobe BlazeDS Developer Guide, Adobe Systems Inc.**

http://livedocs.adobe.com/blazeds/1/blazeds_devguide/index.html

- * **BlazeDS Java AMF Client, Adobe Systems Inc.**

<http://sourceforge.net/adobe/blazeds/wiki/Java%20AMF%20Client/>

- * **Testing Flash Applications, Stefano di Paola**

http://www.owasp.org/images/8/8c/OWASPApSec2007Milan_TestingFlashApplications.ppt

- * **Adobe Flex, AMF 3 and BlazeDS: An Assessment, Jacob Karlson and Kevin Stadmeyer**

http://www.blackhat.com/presentations/bh-usa-08/Carlson_Stadmeyer/BlackHat-Flex-Carlson_Stadmeyer_vSubmit1.pdf

- * **Deblaze, Jon Rose**

<http://deblaze-tool.appspot.com/>

- * **Pentesting Adobe Flex Applications, Marcin Wielgoszewski**

http://blog.gdssecurity.com/storage/presentations/OWASP_NYNJMetro_Pentesting_Flex.pdf

- * **Burp Suite v1.2.14 Release Note, PortSwigger Ltd.**

<http://releases.portswigger.net/2009/08/v1214.html>

Pictures

- * <http://www.rialitycheck.com/portfolio.cfm>
- * <http://www.silexlabs.org/amfphp/>
- * http://cloudfront.qualtrics.com/blog/wp-content/uploads/2010/05/thumbs-up-thumbs-down_orange.jpg
- * http://livedocs.adobe.com/blazeds/1/blazeds_devguide/index.html
- * http://1.bp.blogspot.com/_zMthNE3rsTA/TQjjurmc-tI/AAAAAAAAAL8/fmfG0QP6ODo/s1600/Disappointed_by_taleb83.jpg
- * <http://www.clker.com/clipart-pointer-finger.html>