Fracking Flex

SummerC0n 2010

New York, NY

Because a Flash 0day is so hard to come by…

GOTHAM
DIGITAL • SCIENCE

Fracking Flex

SummerC0n 2010

New York, NY

```
buf[rn] = "%c" % rbyte
```

Who am I?

Marcin Wielgoszewski

- Security Engineer
- Gotham Digital Science

Intro to Flash, Flex and AIR

What is Flex and how does it differ from Flash?

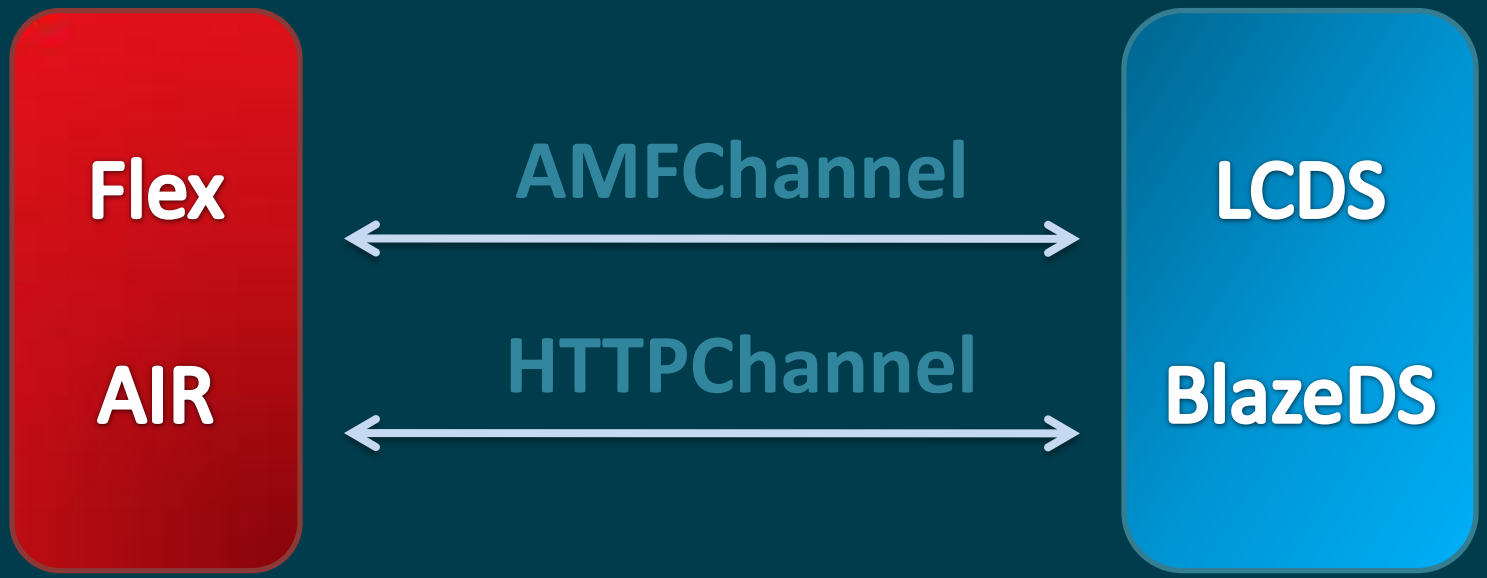
- Flash originally developed for client-side, vector-based animations and video
- Flex provides the framework for building RIA's using the Adobe Flash platform
- AIR allows developers to build desktop applications using Adobe Flash

Adobe LiveCycleDS, BlazeDS, et al.

Utilize existing application logic with Flex

- Provides remoting and messaging capabilities
- Connects backend data services
- Real-time data push to Flash clients

Client / Server Architecture



Channels

Client talks to a server *endpoint* over a Channel

- AMFChannel encapsulates data in AMF
- HTTPChannel encapsulates data in AMFX
- Streaming and Polling channels
- “Secure” channels occur over HTTPS

Endpoints

Channels route requests to a defined endpoint

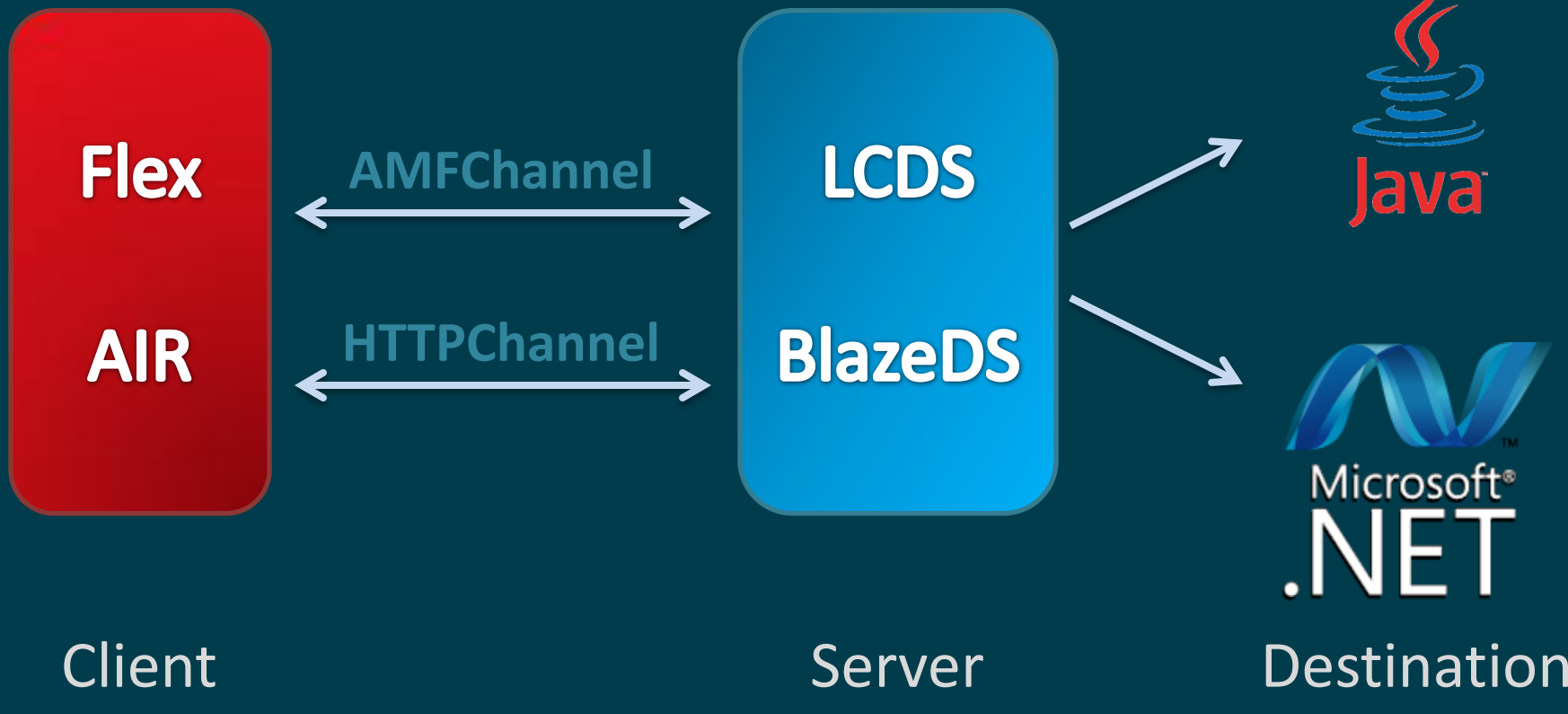
- Servlet-based – AMF/HTTP
- NIO-based – RTMP/AMF/HTTP
- Endpoints ultimately route to a *destination*

Destinations

Here is where a request will ultimately end up

- Could be one of
  - Remoting service
  - Proxy service
  - Message service

Client / Server Architecture



Action Message Format

Adobe format used for data exchange

- Used over AMFChannel/AMFEndpoints
- Requests are serialized into a compact binary format
- Responses are deserialized and processed
- 7-10x faster over XML*

Peek into AMF

AMF Envelopes contain Request Messages

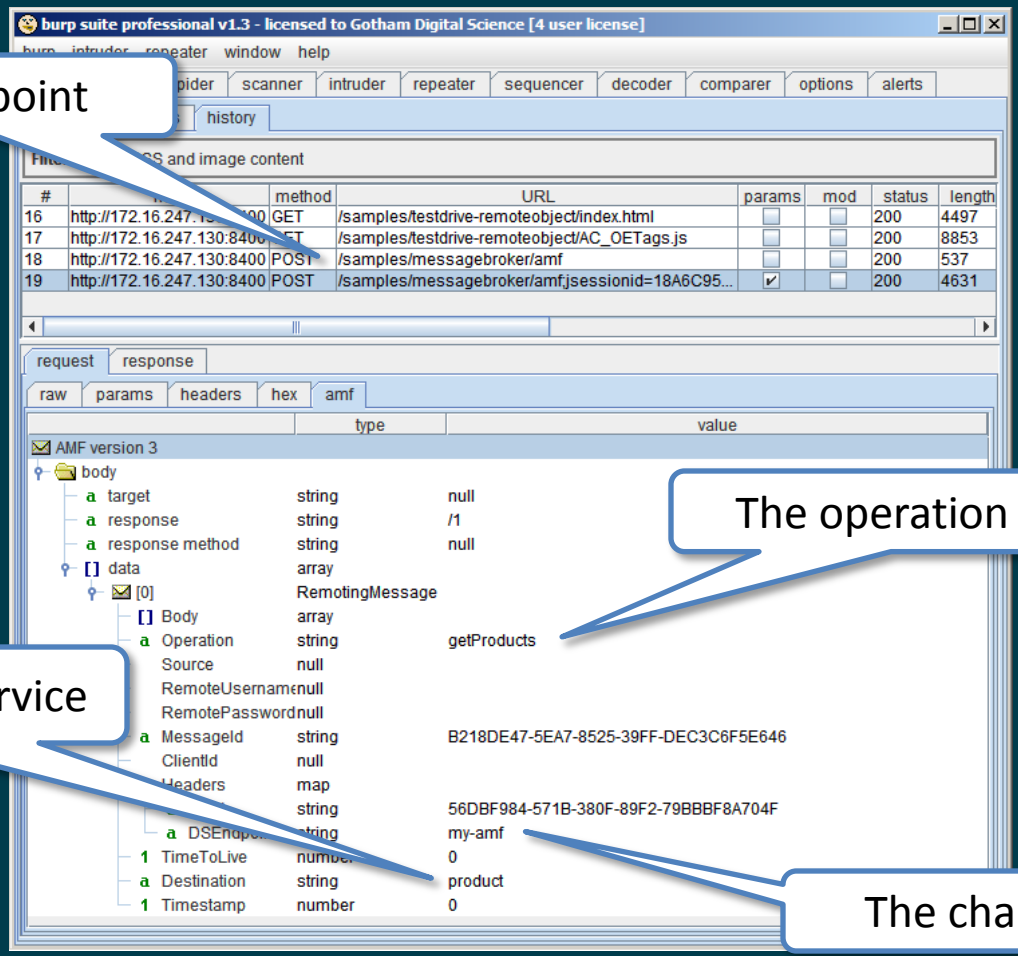
- One HTTP request/response may have several AMF requests/responses
  - RemotingMessage
  - AsyncMessage / CommandMessage
  - AcknowledgeMessage / ErrorMessage
  - HTTPMessage / SOAPMessage

AMF over the wire

```
0x00000000:  00 03 00 00 00 01 00 04 6e 75 6c 6c 00 02 2f 31  |........null../1|
0x00000010:  00 00 00 00 0a 00 00 00 01 11 0a 81 13 4f 66 6c  |.............Ofl|
0x00000020:  65 78 2e 6d 65 73 73 61 67 69 6e 67 2e 6d 65 73  |ex.messaging.mes|
0x00000030:  73 61 67 65 73 2e 52 65 6d 6f 74 69 6e 67 4d 65  |sages.RemotingMe|
0x00000040:  73 73 61 67 65 09 62 6f 64 79 11 63 6c 69 65 6e  |ssage.body.clien|
0x00000050:  74 49 64 17 64 65 73 74 69 6e 61 74 69 6f 6e 0f  |tId.destination.|
0x00000060:  68 65 61 64 65 72 73 13 6d 65 73 73 61 67 65 49  |headers.messageI|
0x00000070:  64 13 6f 70 65 72 61 74 69 6f 6e 0d 73 6f 75 72  |d.operation.sour|
0x00000080:  63 65 15 74 69 6d 65 54 6f 4c 69 76 65 13 74 69  |ce.timeToLive.ti|
0x00000090:  6d 65 73 74 61 6d 70 09 01 01 01 06 0f 70 72 6f  |mestamp......pro|
0x000000A0:  64 75 63 74 0a 0b 01 09 44 53 49 64 06 49 38 32  |duct....DSId.I82|
0x000000B0:  33 30 44 32 35 31 2d 37 42 31 43 2d 34 44 36 46  |30D251-7B1C-4D6F|
0x000000C0:  2d 39 33 43 45 2d 45 30 30 41 33 41 42 37 37 46  |-93CE-E00A3AB77F|
0x000000D0:  34 41 15 44 53 45 6e 64 70 6f 69 6e 74 06 0d 6d  |4A.DSEndpoint..m|
0x000000E0:  79 2d 61 6d 66 01 06 49 45 33 38 39 42 45 45 41  |y-amf..IE389BEEA|
0x000000F0:  2d 46 45 32 45 2d 34 43 37 45 2d 42 31 44 30 2d  |-FE2E-4C7E-B1D0-|
0x00000100:  37 33 31 43 46 44 31 30 46 41 36 32 06 17 67 65  |731CFD10FA62..ge|
0x00000110:  74 50 72 6f 64 75 63 74 73 01 01 01              |tProducts...     |
```

GOTHAM DIGITAL SCIENCE

Identifying message properties



GOTHAM DIGITAL SCIENCE

AMF RemotingMessage

Send RPC's to remote service methods

- Contain the following attributes
  - body
  - destination
  - operation
  - and more…

Flex Remoting Services

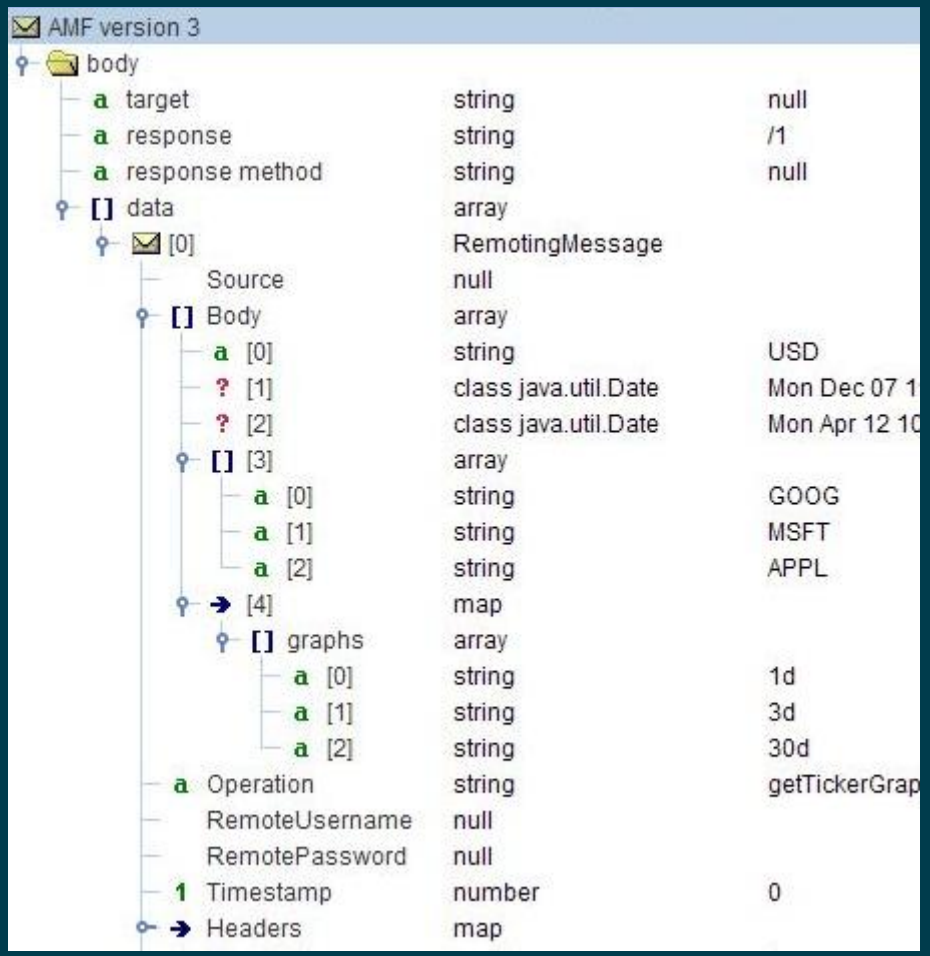
Send complex data structures to services

- Data types and object are preserved from client to server
- Client side Flash ValueObjects interact with backend POJOs

Complex Data Structures

body is an array of objects

- body[0] = string
- body[1] = java.util.Date
- body[2] = java.util.Date
- body[3] = array [
  - string, string, string]
- body[4] = map {
  - [string, string, string] }



Fracking Flex

RECONNAISSANCE

Is it time for flip cup yet?!?

Identify Services and Methods

Inspect the traffic through an HTTP proxy

- Burp Suite, WebScarab, Charles, Wireshark
- Identify the
  - Destination service
  - Operation
  - Endpoint
- How many parameters (and type) are passed?

Decompiling SWFs

The beauty of having client-side code

- AS and MXML is compiled to bytecode
- Developers expose all sorts of good stuff
  - Usernames and passwords
  - URLs and connection strings
  - Hidden functionality
  - and other sensitive data

Decompiling SWFs

Common strings to look for in decompiled code

- RemoteObject | WebService | HTTPService
- .destination | .operation | .useProxy
- get | set | add | remove | create | delete

Local SharedObjects

Persistent “cookies” that reside on filesystem

- Often used to save UI preferences
- Sometimes find cool stuff
  - Session IDs
  - User/Role information
  - Sensitive data

Fracking Flex

ATTACKING REMOTING SERVICES

Enumerating Remoting Services

Do methods/destinations show a pattern?

- Try calling other methods that might be there
  - DeBlaze attempts to enumerate by bruteforce

I got 99 Messages

But my HTTP requests' only one

- Remember, an AMF Envelope can contain more than one Request
- Can we enumerate in just one HTTP request?

Remoting Services

DEMO

A Quick Comparison

Significantly reduce bytes sent and time to test

- Same technique can be applied to fuzzing
- For example…

530 separate HTTP requests

- 150 bytes of headers
- Content-Length: 282
- 1 destination: 1 method
- About 3 minutes

1 HTTP request to do it all:

- 155 bytes of headers
- Content-Length: 148538
- 1 destination: 530 methods
- < 3 seconds

Custom ValueObjects

The server complains about invalid types. WTF?

```
"Cannot convert type java.lang.String with value 'marcin' to
an instance of class flex.samples.crm.employee.Employee"
```

- The client binds ActionScript ValueObjects to server-side POJO's
- Simply passing a string, boolean or an integer isn't enough

Reversing a ValueObject

Well then, what do we do now?

- Decompile client-side code
- Identify the object's namespace
- Identify the object members that are set
- Read the AMF spec and start reversing…

Creating ValueObjects

Use PyAMF or similar API to create a VO

- Define your class and class members
- Alias the class with a namespace
- Pass object as parameter to method

Crafting VO's with Python

```
# Below is some Python-fu for creating an Object Factory
class Factory(object):
    def __init__(self, *args, **kwargs):
        self.__dict__.update(kwargs)

# Register our object factory with a class alias
pyamf.register_class(Factory, "flex.samples.crm.employee.Employee")

# Instantiate a "Employee" using our object factory:
marcin = Factory(**{'firstName': "Marcin",
                    'lastName': "Wielgoszewski",
                    'phone': "555-555-5555",
                    'email': "labs@gdssecurity.com",})
```

Custom ValueObjects

DEMO

No scanner does this, wtf

Fracking Flex

WE HOP THESE THROUGH PROXIES

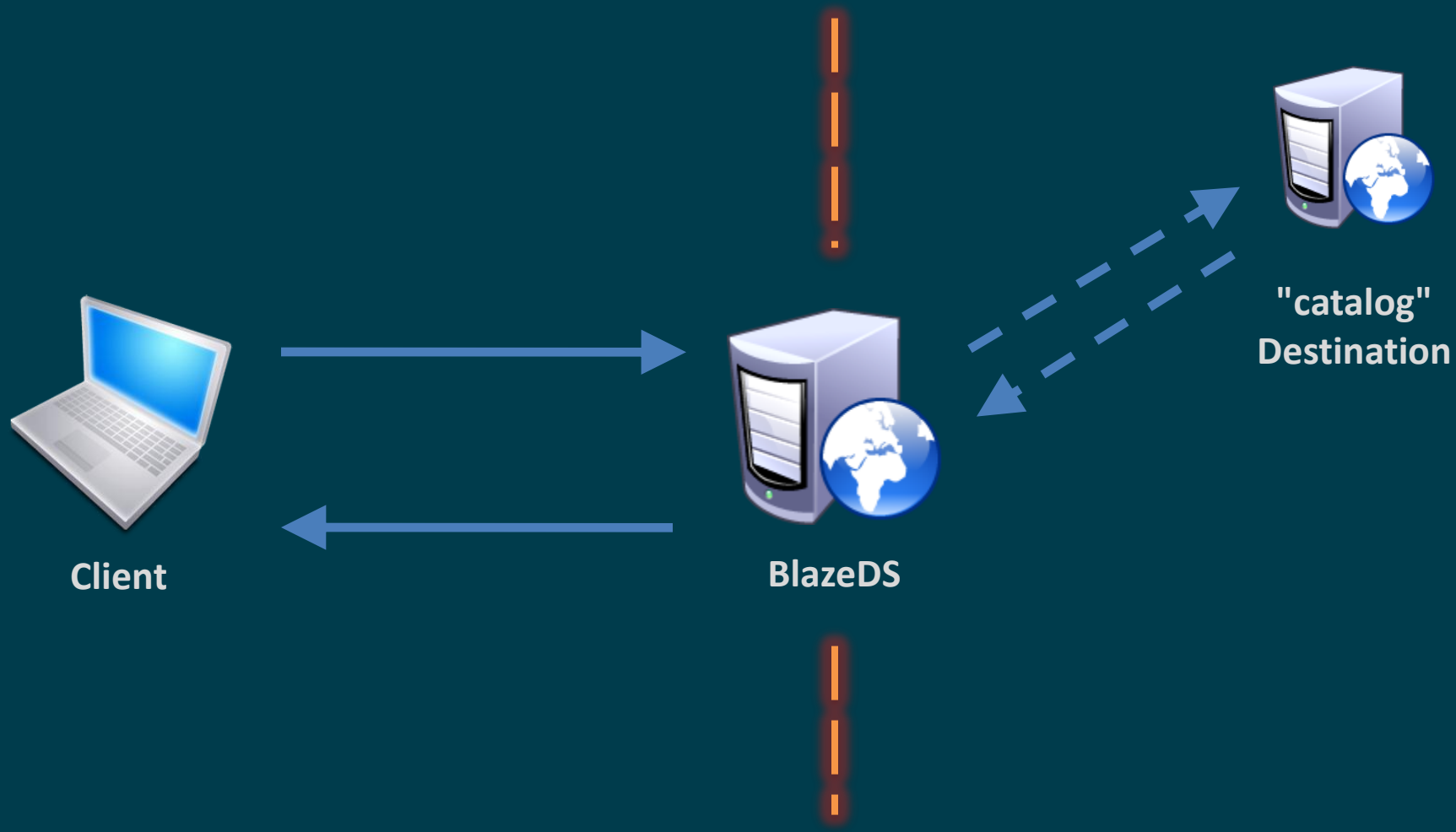
So your packet log is nothing…

BlazeDS Proxy Services

Connect Flex applications to backend services

- Request resources from another domain
- AMF/X wrapped HTTP/SOAP requests

Proxy Service Architecture



AMF HTTPMessage / SOAPMessage

BlazeDS will call a destination on client's behalf

- Get around crossdomain policy restrictions
- Don't want to expose internal service publicly
- HTTP methods supported
  - GET, POST, HEAD, OPTIONS, TRACE, DELETE

Pivoting Intranets through BlazeDS

Proxy Services have inherent risks

- Proxy Services often configured insecurely
- Expose internal/Intranet apps to world
- Culprit? wildcards in *proxy-config.xml*
  - <dynamic-url>*</dynamic-url>
  - <soap>*</soap>

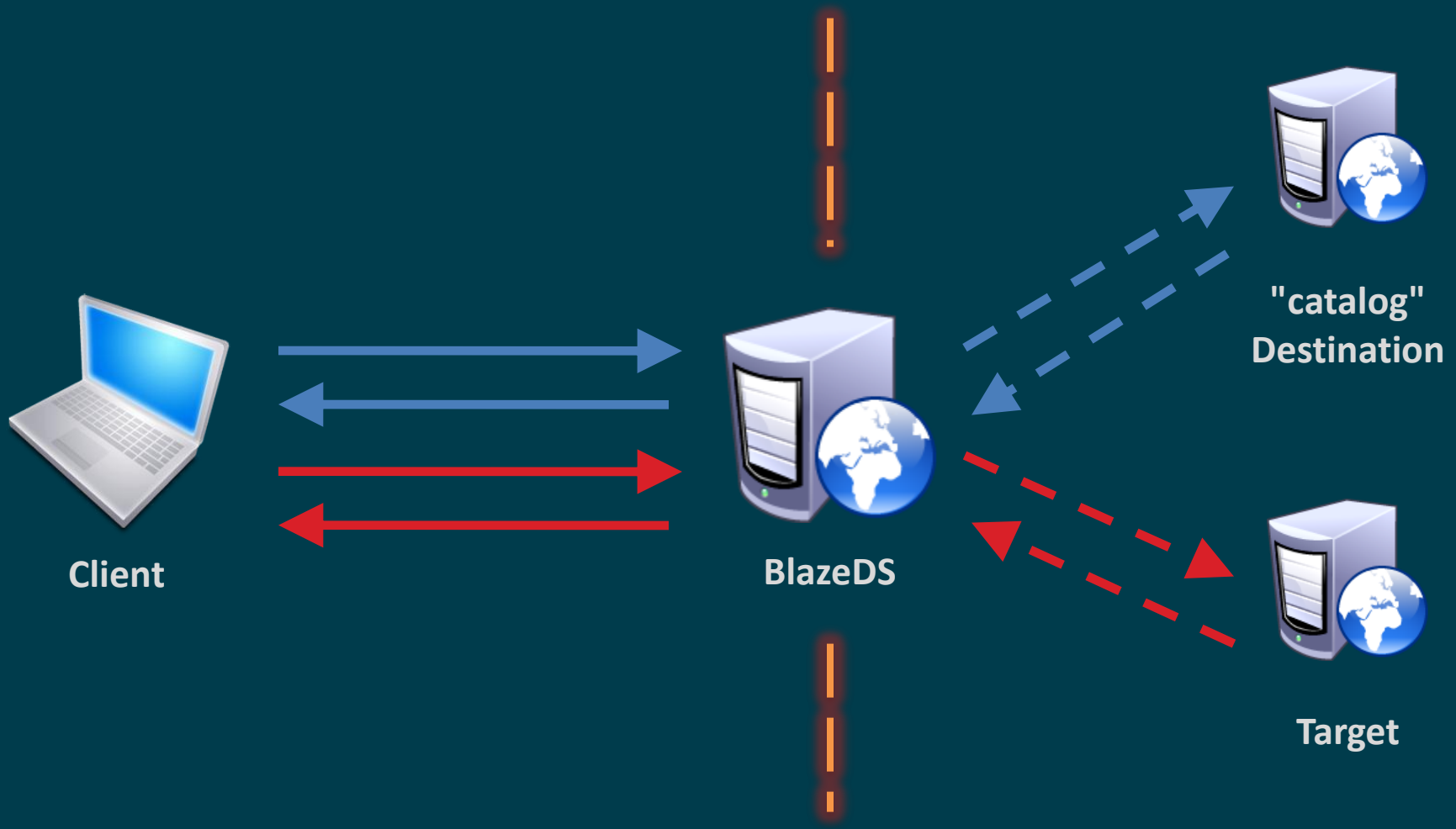
WEB-INF\flex\proxy-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<service id="proxy-service" class="flex.messaging.services.HTTPProxyService">
..snip..

        <destination id="catalog">
          <properties>
            <dynamic-url>*</dynamic-url>
          </properties>
        </destination>

        <destination id="ws-catalog">
          <properties>
            <wsdl>http://livecycledata.org/services/ProductWS?wsdl</wsdl>
            <soap>*</soap>
          </properties>
          <adapter ref="soap-proxy"/>
        </destination>
```

Proxy Service Architecture



Blazentoo

A tool to exploit Proxy Services

- Browse websites reachable from server
  - Hello Intranet applications!
- Can also be a crude port scanner
  - Just specify another port
  - Connection might get refused, reset or stay open…

Blazentoo

DEMO

So f*k your firewall trying to hide your ports

Some Peculiar Behavior…

Destination server response header leakage?

- Proxy request to http://www.google.com/

```
HTTP/1.1 200 OK
..snip..
Server: Apache-Coyote/1.1
Set-Cookie: FLEX_1703289594_47_NID=<blah>; Path=/
Server: gws
X-XSS-Protection: 1; mode=block
```

Flex Assessment Methodology

Let's recap:

- Passively analyze traffic
- Decompile SWF and identify stored secrets
- Enumerate services, methods & endpoints
  - Input validation, fuzzing, etc
  - Check enforcement of AuthN and AuthZ controls
- Exploit insecure configurations

Thanks!

SummerC0n and everyone else who came

- NYSEC crew and all who’ve seen this 3x now
- My fellow GDS colleagues

Marcin Wielgoszewski
Gotham Digital Science
http://www.gdssecurity.com
labs@gdssecurity.com

QUESTIONS?

References

References

BlazeDS Developer Guide - http://livedocs.adobe.com/blazeds/1/blazeds_devguide/

GDS Security Blog - http://www.gdssecurity.com/l/b/

Tools

Burp Suite - http://portswigger.net/suite/

Charles Proxy - http://www.charles.com/

DeBlaze - http://deblaze-tool.appspot.com/

Libraries

PyAMF - http://www.pyamf.org/

RubyAMF - http://rubyamf.org/

AMF::Perl - http://www.simonf.com/flap/

AMFX

Uses an HTTPChannel/HTTPEndpoint

- AMF objects are serialized to **XML**
- Usually provided as a fallback channel
- Different channel == different endpoint
  - URL for AMFX endpoint will differ from AMF

Message serialized to AMFX

```
<amfx ver="3"
   xmlns="http://www.macromedia.com/2005/amfx">
  <body>
    <object type="flex.messaging.messages.HTTPMessage">
      <traits>
        <string>body</string>
        <string>clientId</string>
     ..snip..

    </object>
  </body>
</amfx>
```

AMF CommandMessage

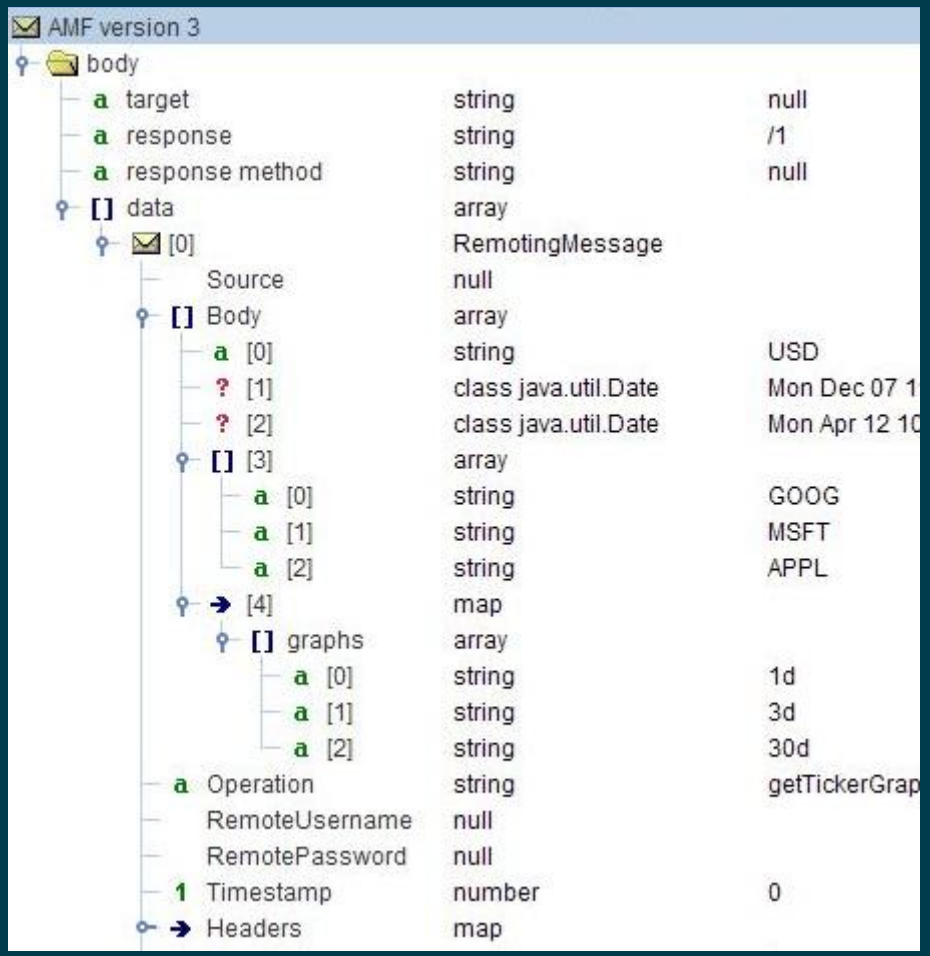
is used to… send commands!

- Mechanism for sending commands related to publish/subscribe, ping, cluster operations
  - Ping
  - Login / Logout
  - Subscribe / Unsubscribe
  - and more..

Complex Data Structures Revisited

body is an array of objects

- body[0] = string
- body[1] = java.util.Date
- body[2] = java.util.Date
- body[3] = array [
  - string, string, string]
- body[4] = map {
  - [string, string, string] }



Complex Data Structures Revisited

Check your API's language type mapping

- Python datetime = date
- Python int/float/long = number
- Python list/tuple = array
- Python dict = map