Tampering 101
Automated Testing of Binary Web Protocols

Chilik Tamir
Information Security Architect, AppSec-Labs
Twitter: @_coreDump
Chilik <at> AppSec-Labs <dot> com

OWASP Israel
15 September 2011
1 slide - Standard web applications

99% - Automating Binary protocols analysis
Standard web application

- Text-based Protocol
- Vast testing methodology

Tools
- Proxies / Sniffers
- Crawlers
- Scanners
- Manual Testing
Binary Web apps

non-text protocols

Flash AMF
Java Serialization / RMI
Diameter / Radius
WCF
CORBA
etc...
Binary Comm. Demo (I)

http://192.168.89.131:8080/students/
Binary Comm. Demo (II)

http://amf.riaspace.com/
POST /messagebroker/amf HTTP/1.1
Accept: */*
Accept-Language: he-IL
Referer: http://amf.riaspaces.com/flex/AmfUsersList.swf/[[DYNAMIC]]/4
x-flash-version: 10,3,183,5
Content-Type: application/x-amf
Content-Length: 294
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 [compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 3.0.04506; .NET4.0C; .NET4.0E]
Host: amf.riaspaces.com
Proxy-Connection: Keep-Alive
Pragma: no-cache
Cookie: __utma=263675164.700408203.1315836929.1315836929.1315836929.1; __utmb=263675164.1.10.1315836929; __utmc=263675164; __utmz=263675164.1315836929.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none)

null/4

flex.messaging.messages.RemotingMessagesource@operation@headers
body/clientId@messageId@timeToLive@destination@timestamp@UsersService@findAllUsers

DSId@null@DSEndpoint

I35123751-910B-6EE9-82BE-00002F1930B8IA3D3D3E2-8640-C9EC-1E52-5E01384FBD80
Binary Comm. Setbacks

- Manual Tampering and Haxing
- Can't automate the process
- Can't use standard tools

Standard web application

- Text-based Protocol
- Vast testing methodology

Tools
- Proxies / Sniffers
- Crawlers
- Scanners
- Manual Testing
Previous Work in Binary Communication Testing

- afx - Reconstructing serialized Java objects from sniffer logs (Forbidden Knowledge vol 14, 2001)

- Shai Chen - PT to Java Client Server Apps (OWASP Israel, 2008)

- Eric Monty et. al - Ruby for pentesters: JRuby (BlackHat, 2009)

- Marcin Wielgoszewski - Pentesting Adobe Flex Applications (OWASP NYNJMetro, 2010)
What's Missing

- On the fly analysis
- Co-op with available tools
- Easy to adopt
- Zero-configuration
- Scriptable / Extendable
- Self Updateable
Enter BELCH

Burp External Channel

One Tool to rule them all
Belch - The missing link:

What's Missing:
- On the fly analysis
- Co-op with available tools
- Easy to adopt
- Zero-configuration
- Scriptable / Extendable
- Self Updateble

Additional:
- Open source
- Free
- Simulate Binary requests
- Simulate Binary responses
- Load testing?
- Insert BL into tampering
- Extendable +++
Download demo project

From [here](#) you can download a sample demo project with ping function implemented.
Belch engine: JVMITM

Binary Object ➔ Java Object ➔ XML (pseudo-protocol)

1. Channel binary communication to JVM
2. Use JVM to reconstruct object in Context
3. Use JVM Serialization support to represent Object and Context as XML
4. Wrap with a pseudo-protocol
5. Tamper with data (manual and scanners)
6. Reverse the flow

XML (pseudo-protocol) ➔ Java Object ➔ Binary Object
Questions?
Tampering 101
Automated Testing of Binary Web Protocols

Chilik Tamir
Information Security Architect, AppSec-Labs
Twitter: @_coreDump
Chilik <at> AppSec-Labs <dot> com

OWASP Israel
15 September 2011