



Get off your AMF
and don't REST on JSON

www.NTOBJECTives.com
www.ManVsWebApp.com

Today's Presenter



Dan Kuykendall

Co-CEO & Chief Technology Officer
NT OBJECTives, Inc.

- NT OBJECTives
 - Member of NTO's founding team
 - Leads security research team
 - Leads vision and design of products
- Blogger and Podcaster
 - **Man vs WebApp** (formerly Mightyseek) - <http://www.manvswebapp.com>
 - An Information Security Place Podcast - <http://infosecplace.com>
- Open Source Developer
 - Linux, Red Hat, Qmail Patches, phpGroupware, podPress, among others

Twitter: @dan_kuykendall



@dan_kuykendall



Overview

1. The changing landscape
2. SQL Injection 101
3. How Web Applications are changing
 - Details about: AMF, REST, JSON
 - Applying SQL Injection to these new formats
4. How this applies to Mobile Apps
5. Defenses
6. Conclusions

Overview

- 1. The changing landscape**
2. SQL Injection 101
3. How Web Applications are changing
 - Details about: AMF, REST, JSON
 - Applying SQL Injection to these new formats
4. How this applies to Mobile Apps
5. Defenses
6. Conclusions



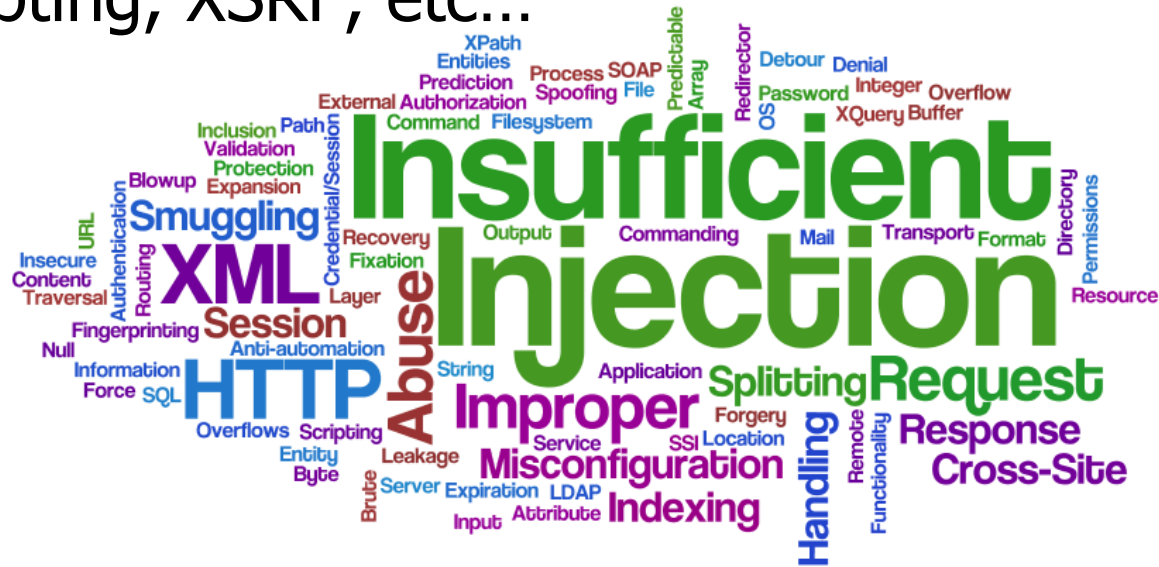
The changing landscape

Overview

www.NTOBJECTives.com
www.ManVsWebApp.com

Web Application Security

- Most common classes of attacks
 - Against the application and database
 - SQL/LDAP/Xpath/OS Injection, etc...
 - Against the application users
 - Cross Site Scripting, XSRF, etc...



@dan_kuykendall



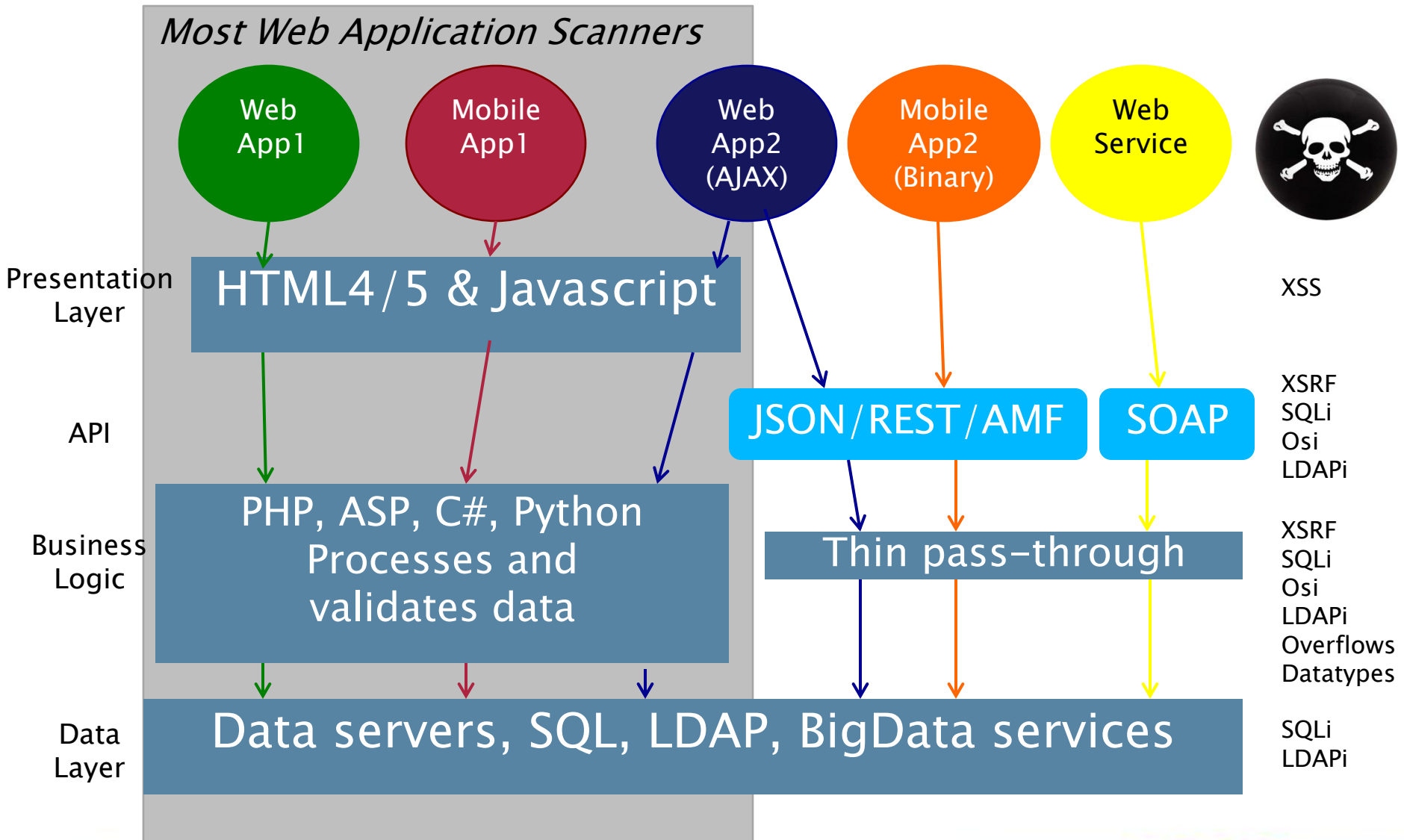
Web Applications are changing

- **No longer just “HTML based” applications**
 - RIA: Flash/Flex, Silverlight, AJAX
 - Mobile clients: Communicate over HTTP to backend services
- No longer just *name=value* format
 - AMF, REST, JSON, SOAP, GWT-RPC, etc...

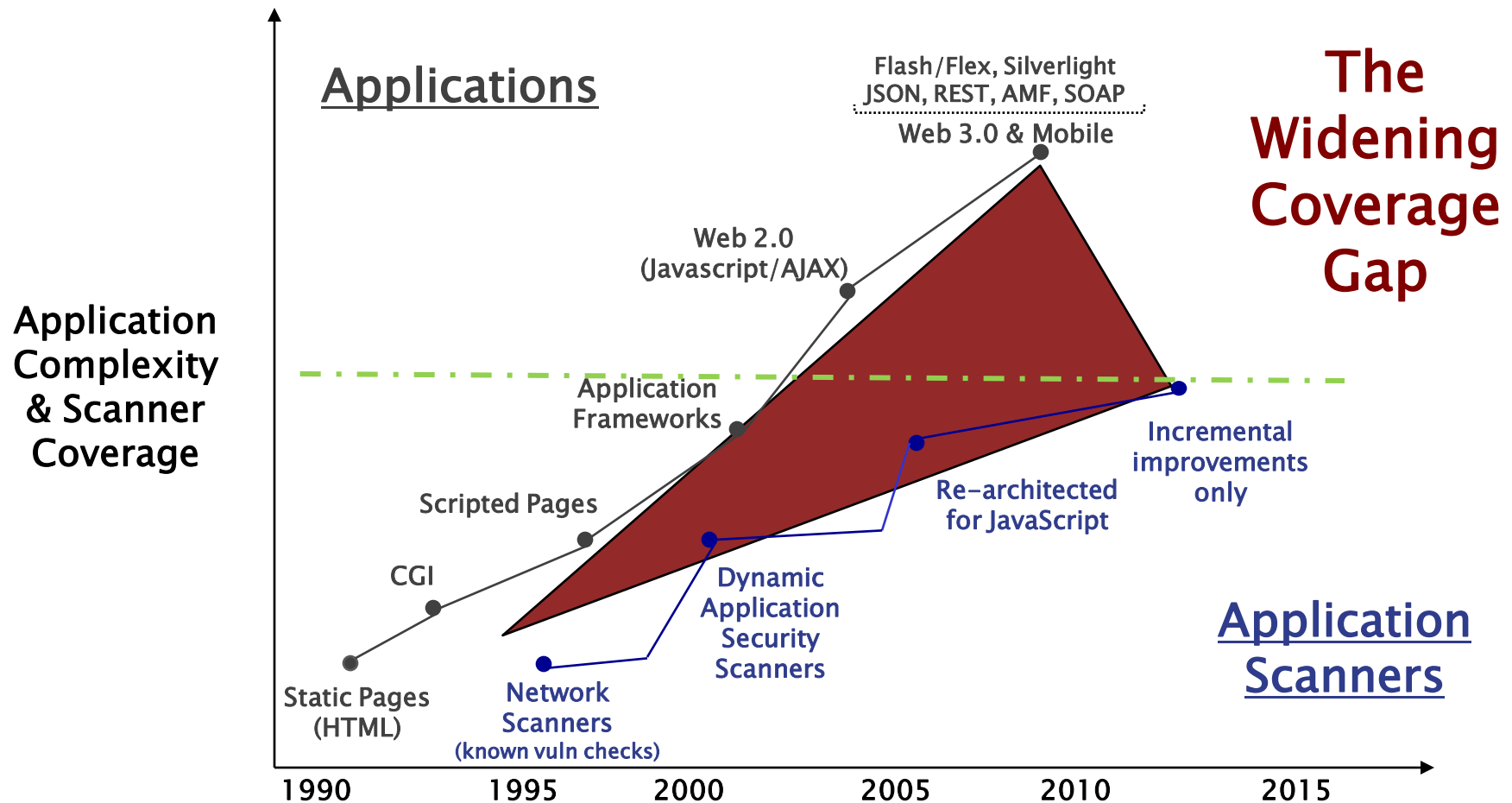
Web Applications are changing

- **No longer just “HTML based” applications**
 - RIA: Flash/Flex, Silverlight, AJAX
 - Mobile clients: Communicate over HTTP to backend services
- No longer just *name=value* format
 - AMF, REST, JSON, SOAP, GWT-RPC, etc...
- Lacking automated tools to help
 - Until NTOSpider 6

Web Application Security Technologies



Automated tools have fallen behind



Overview

1. The changing landscape
- 2. SQL Injection 101**
3. How Web Applications are changing
 - Details about: AMF, REST, JSON
 - Applying SQL Injection to these new formats
4. How this applies to Mobile Apps
5. Defenses
6. Conclusions



SQL Injection 101

Hacking the database



www.NTOBJECTives.com
www.ManVsWebApp.com

Hacking Traditional WebApps - Traffic

- Inputs in simple 'name=value' pairs
- Clicking on a link, the input is on the URL

request	response
<pre>GET /crosstraining/linkout.php?name=Rake HTTP/1.1 Accept: text/html, application/xhtml+xml, */* Referer: http://www.webscantest.com/crosstraining/ Accept-Language: en-US User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; NP06) Accept-Encoding: gzip, deflate Host: www.webscantest.com Proxy-Connection: Keep-Alive Cookie: SESSIONID_VULN_SITE=p5v5h7qp75ccqkj7bp9qep0e92</pre>	

- Submitting a FORM, generates a POST requests

request	response
<pre>POST /crosstraining/linkout.php HTTP/1.1 Accept: text/html, application/xhtml+xml, */* Referer: http://www.webscantest.com/crosstraining/ Accept-Language: en-US User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; NP06) Content-Type: application/x-www-form-urlencoded Accept-Encoding: gzip, deflate Host: www.webscantest.com Proxy-Connection: Keep-Alive Pragma: no-cache Cookie: SESSIONID_VULN_SITE=p5v5h7qp75ccqkj7bp9qep0e92 Content-Length: 76</pre>	

name=Rake



© 2012 NT OBJECTIVES



Web Apps using SQL

- Common to take user input and generate a SQL Statement



http://www.webscantest.com/datastore/search_get_by_lastname.php?name=O'Brian

- Behind the scenes a SQL statement is waiting for user input

```
SELECT * from tPeople WHERE lastname='<b>%name%</b>'
```

%name% will be replaced by the **name** value from the URL

Web Apps using SQL

- Common to take user input and generate a SQL Statement

http://www.webscantest.com/datastore/search_get_by_lastname.php?name=O'Brian

- Behind the scenes a SQL statement is waiting for user input

```
SELECT * from tPeople WHERE lastname='<b>%name%</b>'
```

%name% will be replaced by the **name** value from the URL


- All too often the input is dropped into the statement without escaping

```
SELECT * from tPeople WHERE lastname='<b>O'Brian</b>'
```

invalid SQL statement

Web Apps using SQL

- Common to take user input and generate a SQL Statement



`http://www.webscantest.com/datastore/search_get_by_lastname.php?name=O'Brian`

- Behind the scenes a SQL statement is waiting for user input

```
SELECT * from tPeople WHERE lastname='<b>%name%</b>'
```

%name% will be replaced by the **name** value from the URL

- All too often the input is dropped into the statement without escaping

```
SELECT * from tPeople WHERE lastname='<b>O'Brian</b>'
```

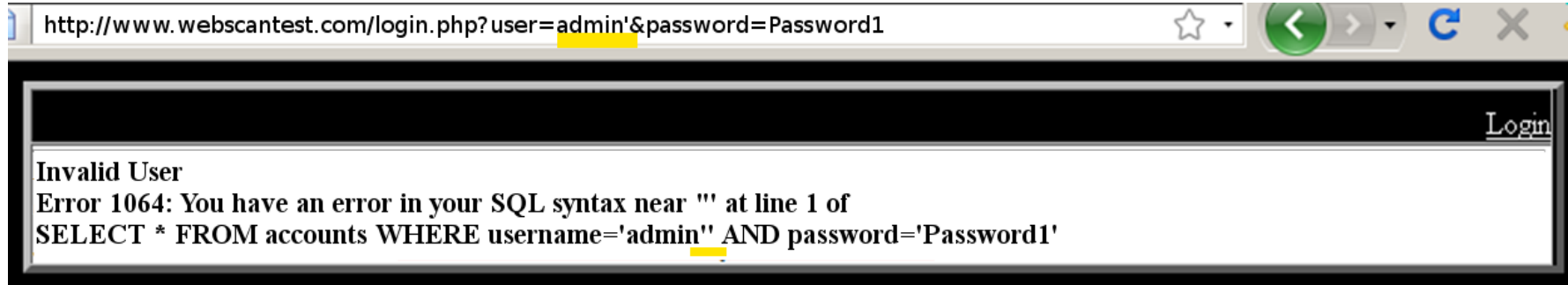
invalid SQL statement

- Developers should be escaping special characters (i.e. quote marks)

```
SELECT * from tPeople WHERE lastname='<b>O\'Brian</b>'
```


Standard SQL Injection Example

- Often errors will be created and displayed



```
SELECT * from tAccounts WHERE username='admin' AND password='Password1'
```

- Attackers can use these errors as clues and then create an exploit
- Example: **admin'#**

```
SELECT * from tAccounts WHERE username='admin'# ' AND password='Password1'
```

After the # the rest of the statement will be considered comments and will be discarded. Now logged in as admin

SQL Injection Demo: Classic web app

Store

192.168.100.212/Catalog1.aspx?id=1

☆ 🔍

BOOKSTORE

Classic

JSON

REST

AMF

SOAP

Flash

Welcome to Book Store!

This is a template designed by free website templates for you for free you can replace all the text by your own text. This is just a place holder so you can see how the site would look like. If you're having problems editing the template please don't hesitate to ask for help on the forum. You will get help



Dont make me think

Common Sense
People won't use your web site if they can't find their way around it. Whether you call it usability, ease-of-use, or just good design, companies staking their fortunes and their futures on their Web sites are starting to recognize that it's a bottom-line issue. In Don't Make Me Think, usability expert Steve Krug distills his years of experience and observation into clear, practical--and often amusing--common sense advice for the people in the trenches (the designers, programmers, writers, editors, and Webmasters), the people who tell them what to do (project managers, business planners, and marketing people), and even the people who sign the checks.

Steve Krug

**DON'T
MAKE
ME
THINK**

0 matches

Overview

1. The changing landscape
2. SQL Injection 101
- 3. How Web Applications are changing**
 - Details about: AMF, REST, JSON
 - Applying SQL Injection to these new formats
4. How this applies to Mobile Apps
5. Defenses
6. Conclusions



Changing Landscape

The wild west is getting wilder



www.NTOBJECTives.com
www.ManVsWebApp.com

Popular new formats

- Flash & Flex Applications

- **AMF**: Adobe/**A**ctionscript **M**essaging **F**ormat

```
00000000null00/3000<
000000
00Oflex.messaging.messages.RemotingMessagesource0operation0destination0time
0000admin0junk
00DSId0IBA98D1B7-5E1C-6007-6D98-2C3EF173C5AF0DSEndpoint0my-amf00IOB63
```

- AJAX: Asynchronous JavaScript and XML

- **REST**: **RE**presentational **S**tate **T**ransfer

```
http://example.com/catalog/item/17
```

- **JSON**: **J**ava**S**cript **O**bject **N**otation

```
{"catalog": { "item": 17 }}
```

- There are several others that will not be covered today

AMF: Actionscript Messaging Format

- The heart of Flash-Remoting
- Used by Flash & Flex apps
 - Online games
 - Marketing campaigns
- Binary data object
- Has 2 major versions (AMF0 and AMF3)
- Decoders are available in many languages



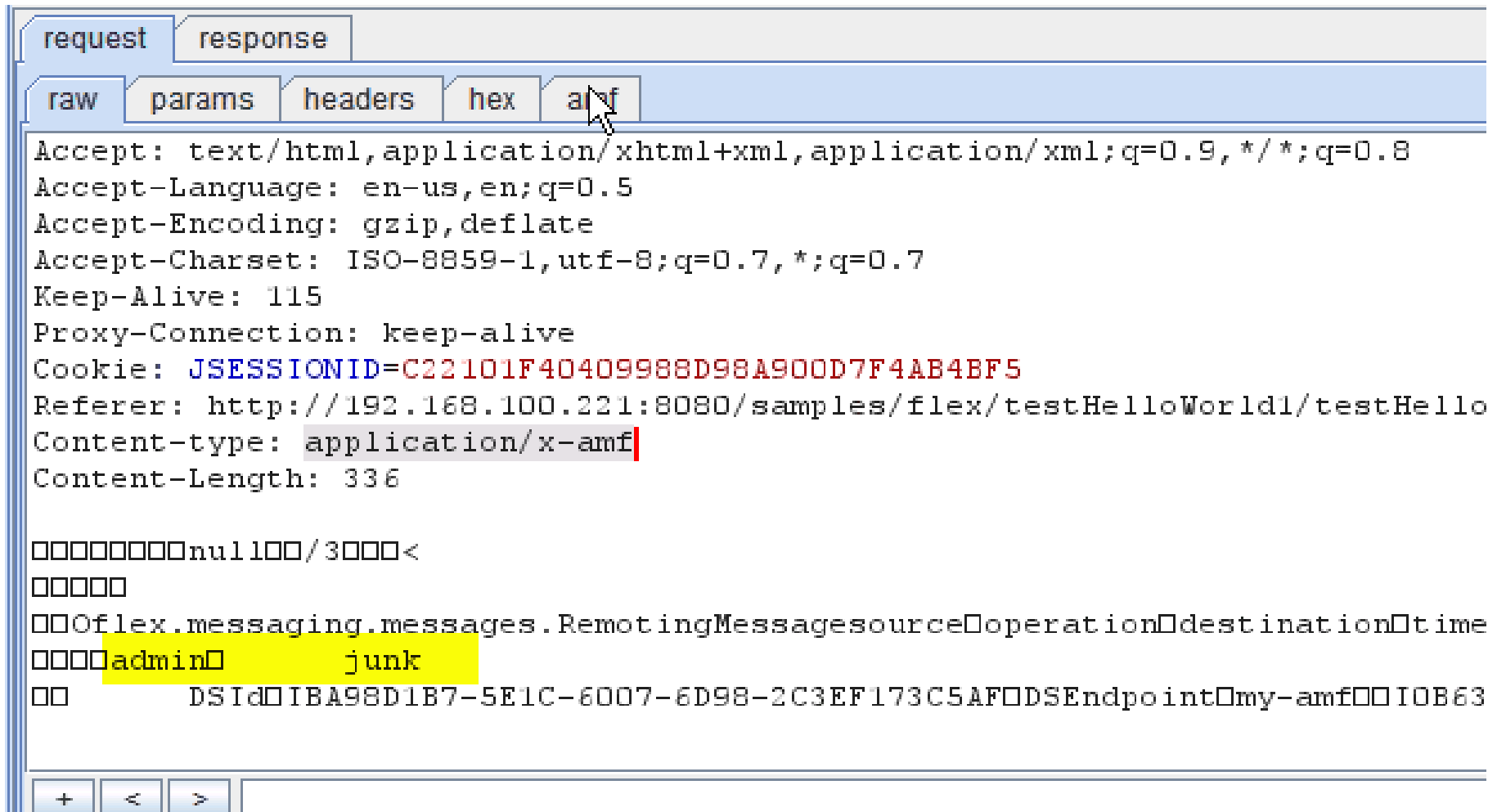
Hacking AMF – Loading applet

http://192.168.100.221:8080/samples/flex/testHelloWorld1/testHelloWorld1.html


```
Source of: http://192.168.100.221:8080/samples/flex/testHelloWorld1/testHelloWorld1.html - Mozilla Firefox
File Edit View Help
+ 'This content requires the Adobe Flash Player. '
+ '<a href=http://www.adobe.com/go/getflash/>Get Flash</a>';
document.write(alternateContent); // insert non-flash content
}
// -->
</script>
<noscript>
  <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    id="testHelloWorld1" width="100%" height="100%"
    codebase="http://fpdownload.macromedia.com/get/flashplayer/current
/swflash.cab">
    <param name="movie" value="testHelloWorld1.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#869ca7" />
    <param name="allowScriptAccess" value="sameDomain" />
    <embed src="testHelloWorld1.swf" quality="high" bgcolor="#869ca7"
      width="100%" height="100%" name="testHelloWorld1" align="middle"
      play="true"
      loop="false"
      quality="high"
      allowScriptAccess="sameDomain"
      type="application/x-shockwave-flash"
      pluginspage="http://www.adobe.com/go/getflashplayer">
    </embed>
  </object>
</noscript>
</body>
</html>
Line 105, Col 54
```

Hacking AMF – Raw traffic

- Looks like binary garbage



The screenshot shows a web browser's developer tools interface. The 'request' tab is selected, and the 'raw' sub-tab is active. The raw request data is displayed as a series of headers and a body. The headers include Accept, Accept-Language, Accept-Encoding, Accept-Charset, Keep-Alive, Proxy-Connection, Cookie, and Referer. The body is an AMF message, which is displayed as a series of bytes. The message is a 'flex.messaging.messages.RemotingMessage' with a 'destination' of 'admin' and a 'time' of 'junk'. The message is also identified by a 'DSId' and a 'DSEndpoint'.

```
request response
raw params headers hex amf
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
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Proxy-Connection: keep-alive
Cookie: JSESSIONID=C22101F40409988D98A900D7F4AB4BF5
Referer: http://192.168.100.221:8080/samples/flex/testHelloWorld1/testHello
Content-type: application/x-amf
Content-Length: 336

          null  /3    <
      
  Oflex.messaging.messages.RemotingMessage  operation  destination  time
    admin  junk
  DSId  IBA98D1B7-5E1C-6007-6D98-2C3EF173C5AF  DSEndpoint  my-amf  IOB63
```


Hacking AMF – Decoded

- AMF can be decoded and its easy to pinpoints the data elements

	type	value
a response	string	/3
a response method	string	null
[] data	array	
✉ [0]	RemotingM...	
Source	null	
[] Body	array	
a [0]	string	admin
a [1]	string	junk
a Operation	string	sayHello
RemoteUsername	null	
RemotePassword	null	
1 Timestamp	number	0
➔ Headers	map	
1 TimeToLive	number	0
a Destination	string	HelloWorld

SQL Injection Demo: Using AMF

Store

← → ↺ 192.168.100.212/catalogamf.aspx

BOOKSTORE

Classic

JSON

REST






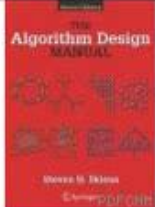

AMF

SOAP

Flash

Welcome to Book Store!

This is a template designed by free website templates for you for free you can replace all the text by your own text. This is just a place holder so you can see how the site would look like. If you're having problems editing the template please don't hesitate to ask for help on the forum. You will get help



Dont make me think

Common Sense

People won't use your web site if they can't find their way around it. Whether you call it usability, ease-of-use, or just good design, companies staking their fortunes and their futures on their Web sites are starting to recognize that it's a bottom-line issue. In Don't Make Me Think, usability expert Steve Krug distills his years of experience and observation into clear, practical--and often amusing--common sense advice for the people in the trenches (the designers, programmers, writers, editors, and Webmasters), the people who tell them what to do (project managers, business planners, and marketing people), and even the people who sign the checks.

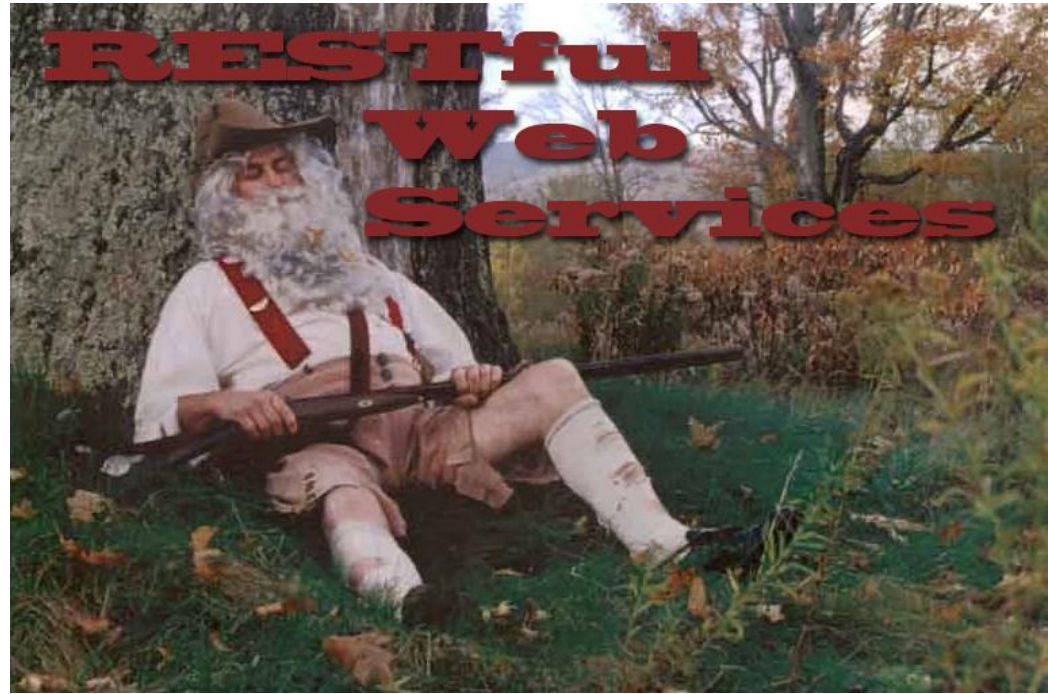
Steve Krug

**DON'T
MAKE
ME
THINK**

REST: REpresentational State Transfer

- An Architectural Style, Not a Standard

- RESTful applications provide often provide easily understood interfaces



- The output could be in any chosen format
 - Common RESTful file formats:
RSS, ATOM, CSV, ICS, VCF

RESTful Example - URI

- Simple URI example
 - Collection URI would provide list of record

```
http://example.com/catalog/list/*
```

- Element URI would provide record data

```
http://example.com/catalog/item/17
```

RESTful Example - XML

- Simple XML example – XML sent in POST data
 - Collection method would provide list of resources

```
POST /resources/
```

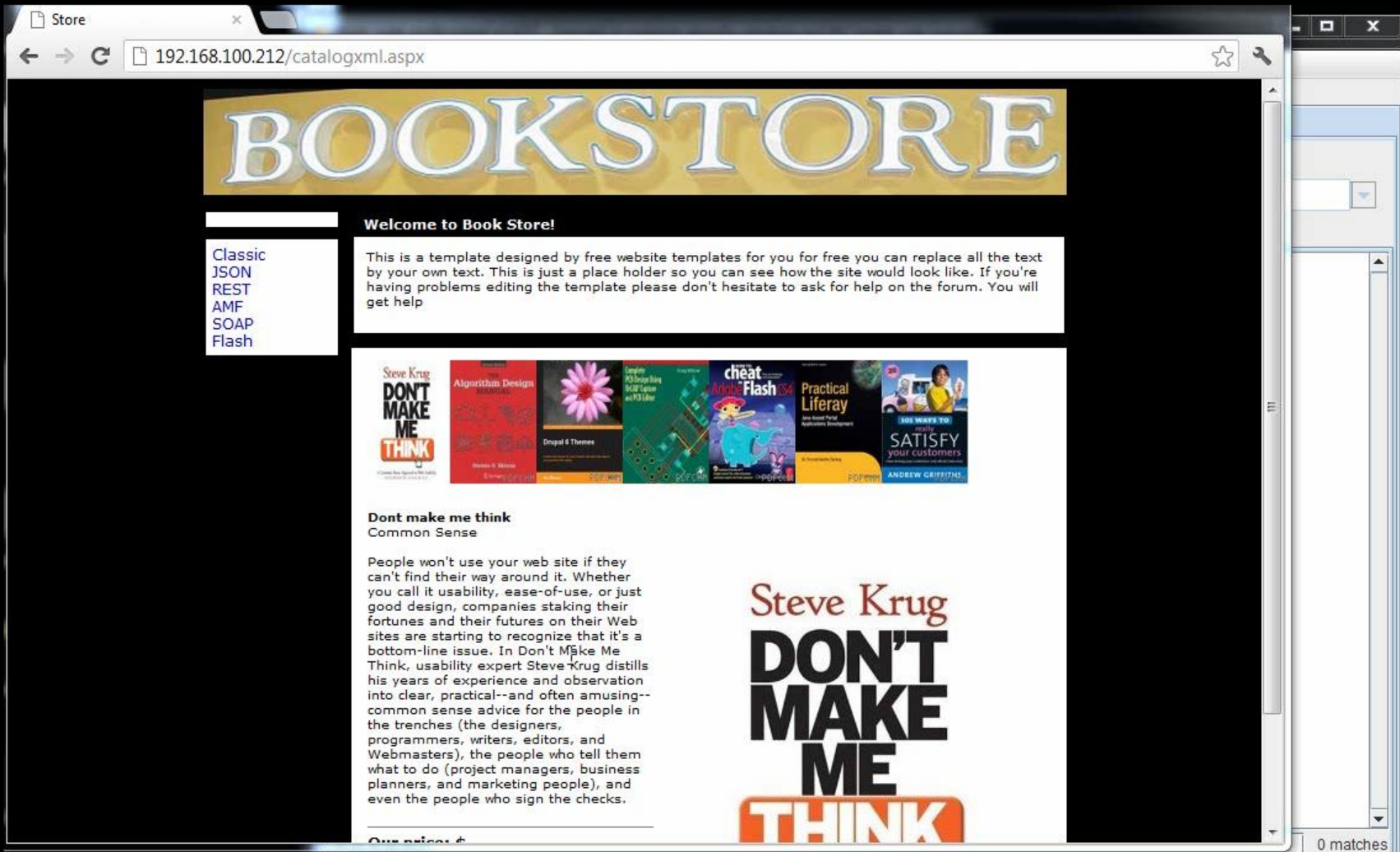
```
<Catalog><List>*</List></Catalog>
```

- Element value would provide record data

```
POST /resources/
```

```
<Catalog><id>17</id></Catalog>
```


SQL Injection Demo: RESTful services



JSON: JavaScript Object Notation

- Created specifically for **JavaScript**
- Text based serialized arrays
- Has become the most popular format in use because its easy to create & parse
- Sent as GET parameter or POST data
- AJAX: **A**synchronous **J**avascript **A**nd **X**ML
 - JSON is native for AJAX frameworks, including jQuery
- *Mobile apps use JSON*



JSON: Example Traffic

- Serialized array – Shown nicely formatted

```
{
  "widget": {
    "debug": "on",
    "window": {
      "title": "Sample Konfabulator Widget",
      "name": "main_window",
      "width": 500,
      "height": 500
    },
    "image": {
      "src": "Images/Sun.png",
      "name": "sun1",
      "hOffset": 250,
      "vOffset": 250,
      "alignment": "center"
    },
    "text": {
      "data": "Click Here",
      "size": 36,
      "style": "bold",
      "name": "text1",
      "hOffset": 250,
      "vOffset": 100,
      "alignment": "center",
      "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
    }
  }
}
```

- Serialized array – As one line

```
{"widget": { "debug": "on", "window": { "title": "Sample Konfabulator Widget", "name": "main_window", "width": 500, "height": 500 }, "image": { "src": "Images/Sun.png", "name": "sun1", "hOffset": 250, "vOffset": 250, "alignment": "center" }, "text": { "data": "Click Here", "size": 36, "style": "bold", "name": "text1", "hOffset": 250, "vOffset": 100, "alignment": "center", "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;" } }}
```


SQL Injection Demo: Using JSON

Store






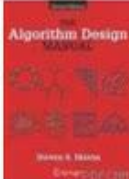

← → ↺ 192.168.100.212/Catalog.aspx

BOOKSTORE

Classic
JSON
REST
AMF
SOAP
Flash

Welcome to Book Store!

This is a template designed by free website templates for you for free you can replace all the text by your own text. This is just a place holder so you can see how the site would look like. If you're having problems editing the template please don't hesitate to ask for help on the forum. You will get help



Teach Yourself TCP/IP in 24 Hours

As the Internet continues to expand its reach, more and more users, administrators, and programmers need to learn about TCP/IP the core standard behind the Internet, and the dominant protocol for networks throughout the world. Sams Teach Yourself TCP/IP in 24 Hours provides a clear and concise introduction to TCP/IP. It is accessible enough for non-technical readers, yet specific enough for technical readers who are looking for a solid foundation in TCP/IP. This edition adds coverage of recent developments that affect TCP/IP. New topics added include: wireless networking, spam control, broadband, and peer-to-peer networking technologies.

Steve Krug

DON'T
MAKE
ME
THINK

Overview

1. The changing landscape
2. SQL Injection 101
3. How Web Applications are changing
 - Details about: AMF, REST, JSON
 - Applying SQL Injection to these new formats
- 4. How this applies to Mobile Apps**
5. Conclusions



Hacking Mobile...

Because no one seems to be
talking about mobile...



www.NTOBJECTives.com
www.ManVsWebApp.com

Hacking Mobile Apps

- Mobile app markets - Lack monitoring
- BYOD – Means users are integrating personal devices and bad apps with company systems
- Assumed secure transport protocol... wrong
- Problems with authentication, often just use MEID/IMEI/ESN
- Most security people still have no idea what to look for
 - Few tools to help
 - No time to do manual reviews

Hacking Mobile Apps

- Stop looking at the device in your hand...
 - the real target is the on the backend
- Get traffic data
 - Run App in Emulator
 - Proxy traffic from phone
 - Transparent proxy on router
- Mobile app formats
 - JSON is native format used by many development toolkits
 - REST is also used
 - Some even use classic web app parameter format

Hacking Mobile Apps – Words With Friends

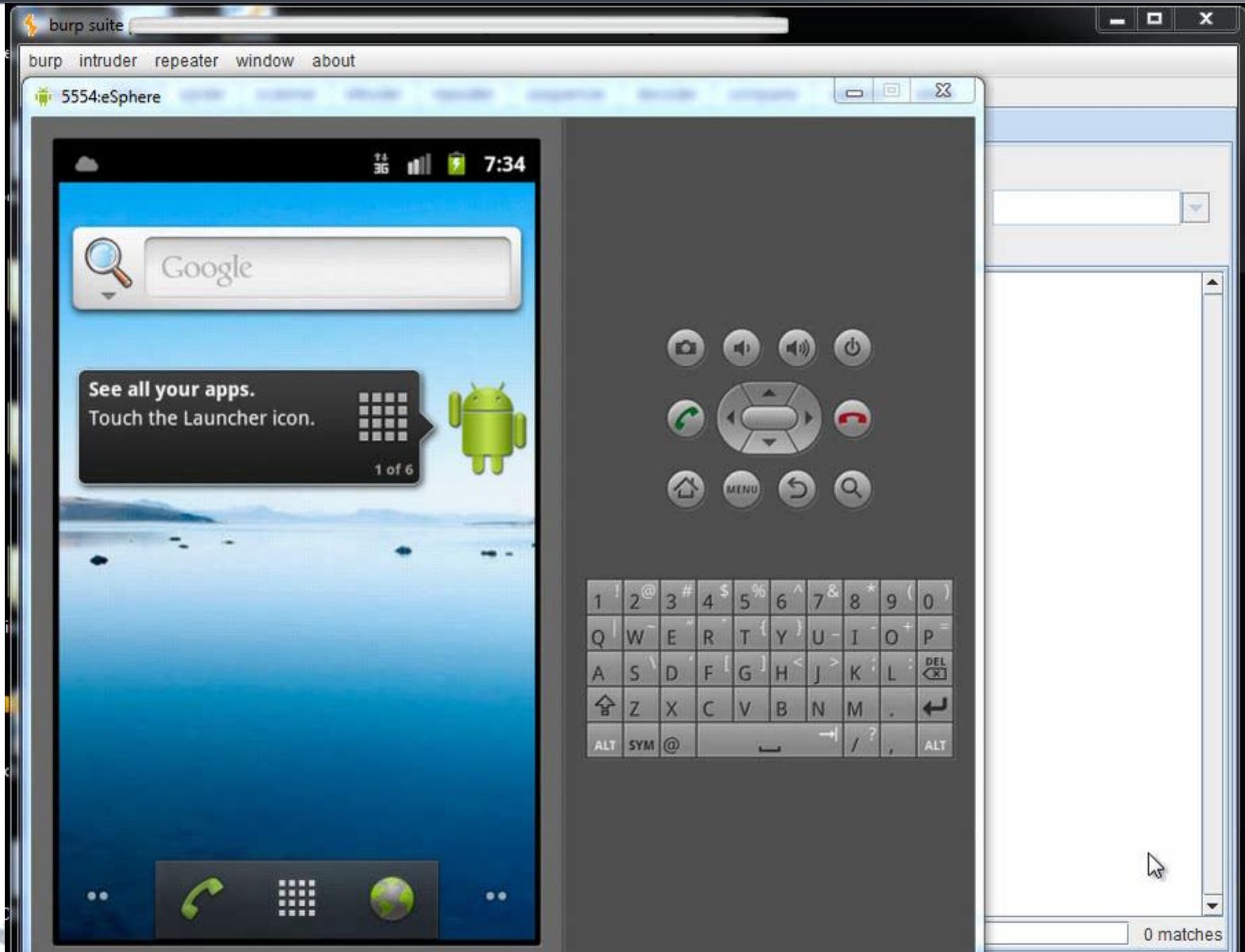
- Request to get advertising

```
POST /Services/PubAd.svc/GetSingleAdPlacement HTTP/1.1
Content-type: text/json
Host: jupiter.appads.com
Proxy-Connection: close
Connection: close
Content-Length: 1315
```

```
{ "data": { "Id": "100de38f", "acceptLanguage": "", "adPool": 0, "adSizes": [ "320x48", "320x24", "300x50", "250x50", "320x480" ], "androidId": "3MM/3lp3yaN5wWHUjebSGuMA1BScjKzgFvAw6htgMwY=", "bundleId": "com.zynga.words", "ccs": "310260;T-Mobile;GSM", "cct": 2, "cctDetailed": "", "clientDateTime": "4,248", "cookie": [ { "maxage": 2592000, "name": "fc821", "value": "ref=1202221048" }, { "maxage": 2592000, "name": "vw", "value": "ref=1202221048&14f=6,df,default,NT&31d=,df,default,3PO" }, { "maxage": 172800, "name": "1c", "value": "14744" }, { "maxage": 86400, "name": "iuq1S7ZzRhvUej10-cTBRERA", "value": "x" }, { "maxage": 172800000, "name": "u", "value": "ePU8WZJF4UidalJlpo55LA" } ], "crParms": "", "debugFlags": 0, "deviceId": "22c009a281dd2030f110b9c63d95ac44f283d8fc", "encDevId": "1329dA+FS1guVoOQSiCvLlhmEn4M5H8Bo+Af38vCwdI=", "ipAddress": "", "noTrack": 0, "placement": "", "pubTargeting": "Millennial=1", "publisher": "q1S7ZzRhvUej10-cTBRERA", "rvCR": "13842", "type": "iq", "userAgentInfo": { "Build": "1.11.0.6962", "Carrier": "T-Mobile", "Density": "High", "Device": "HTC Vision", "DeviceFamily": "htc_ww", "MCC": "310", "MNC": "260", "Platform": "Android", "PlatformVersion": "2.3.7", "ScreenResolution": "480x800", "v": "1", "webUserAgent": "Mozilla/5.0 (Linux; U; Android 2.3.7; en-us; HTC Vision Build/GRI40) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1" }, "zone": "0955151879139204689" }
```

Can you tell what format this is?

SQL Injection Demo: Mobile JSON



Hacking Mobile Apps: Examples

- NT OBJECTives Research team
 - SQL Injection 101 against the media – AP Mobile

Request

```
GET /rest/v1/appush/getbreakinglist?count=10&page=1&fromdate=2012-10-09 HTTP/1.1
Host: services.apmobileapps.com
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 0
Connection: close
Accept: application/json
User-Agent: AP%20Mobile/5.4 CFNetwork/609 Darwin/13.0.0
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Cache-Control: no-cache
Vary: Accept-Encoding
Date: Tue, 23 Oct 2012 19:35:36 GMT
Server: Google Frontend
Connection: close
```

```
{"breakingnewsitem": [{"apbreakingnewsid": "8TT44mAO", "apbreakingnewstext":
"Appeals Court says Indiana can't end Medicaid funding for Planned Parenthood
over abortions.", "apbreakingnewsstoryid": null, "apbreakingnewscreationdate":
"2012-10-23 16:30:50", "apalerttype": "banner", "apcategorytags": null}<snip>
```



Hacking Mobile Apps: Examples

- NT OBJECTives Research team
 - SQL Injection 101 against the media – AP Mobile

Request

```
GET /rest/v1/appush/getbreakinglist?count=10%27//136&page=1&fromdate=2012-10-09 HTTP/1.1
Host: services.apmobileapps.com
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 0
Connection: close
Accept: application/json
User-Agent: AP%20Mobile/5.4 CFNetwork/609 Darwin/13.0.0
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Cache-Control: no-cache
Vary: Accept-Encoding
Date: Tue, 23 Oct 2012 19:35:36 GMT
Server: Google Frontend
Connection: close

Query failed -> SELECT * FROM APPushNotification WHERE apbreakingnewscreationdate > DATE('2012-10-09')
ORDER BY apbreakingnewscreationdate DESC LIMIT 10'//136
```

Hacking Mobile Apps: Examples

- NT OBJECTives Research team – Mall Day
 - Wanted a place with little or no “laptop traffic”
 - Setting up rouge hotspots (attwifi, panera, hhonors)



Hacking Mobile Apps: Examples

- NT OBJECTives Research team – Mall Day
 - Collect MEID's – Too many apps use for auth
 - Collect session tokens that rarely expire
 - Collect usernames & passwords
 - Plume Twitter app: Uses basic auth in each request.
 - Many apps that use SSL don't require a valid cert
 - **The Register**® 21st Oct 2012 22:05 GMT
Android apps get SSL wrong, expose personal data
Researchers find 1,000 insecure apps, pinch credit card and other data
http://www.theregister.co.uk/2012/10/21/android_app_ssl_vulnerability/

Hacking Mobile Apps: Examples

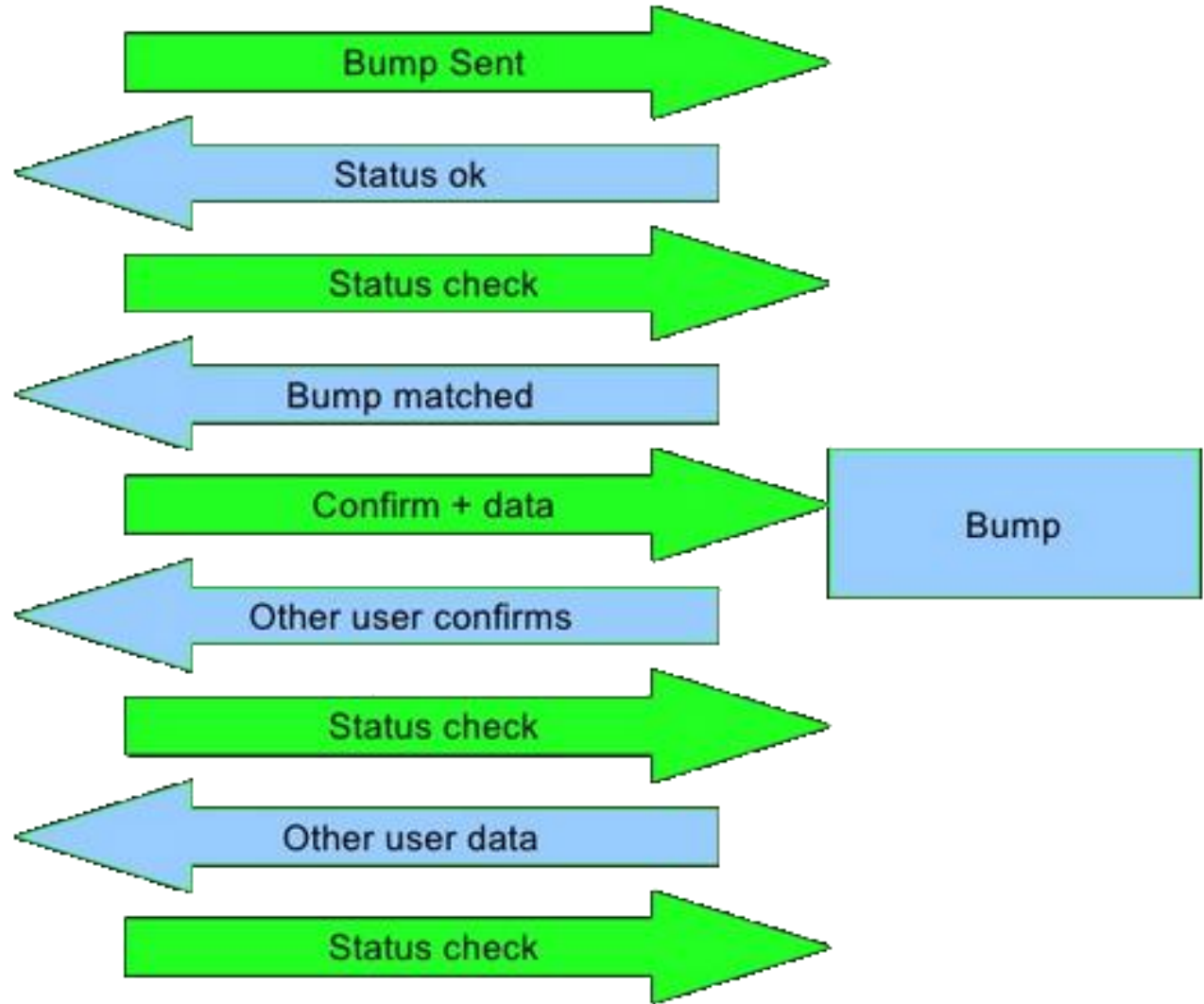
- Word with Friends: Bypassing word verification
- MJ Keith Security Researcher for Denim Group
 - MyBackup Pro: It's basic auth credentials are hardcoded user/pass in for all users.
 - Addressbook Pro: Asks for credentials, but only used to unlock the app on the phone not with servers
 - "Pwn on the go!" and hacking Bump

Hacking Mobile Apps: Examples

- MJ Keith Security Researcher for Denim Group
 - "Pwn on the go!" and hacking Bump



Hacking Mobile Apps: Bump



Overview

1. The changing landscape
2. SQL Injection 101
3. How Web Applications are changing
 - Details about: AMF, REST, JSON
 - Applying SQL Injection to these new formats
4. How this applies to Mobile Apps
- 5. Conclusions**



Conclusions



www.NTOBJECTives.com
www.ManVsWebApp.com

Conclusions

- Remember:
 - Web applications and services are the primary target of malicious hackers
 - Most web applications are vulnerable to attack
- Attack payloads:
 - We have understood SQL Injection for over 10 years yet on average 32% of applications are still vulnerable to SQL Injection
 - And now, we are going beyond HTML to new formats and technologies, the vulnerabilities are growing faster than we can resolve them
 - Learn about the different attack payloads (SQLi, XSS, LDAP injection, etc)

Conclusions

- What you can do:
 - Get to know these new technologies and formats, JSON, AMF, etc.
 - Learn to watch traffic (Burp Suite)
 - Practicing editing the traffic to force the application to throw error messages and enable you to discover vulnerabilities
- Keep in mind:
 - WAF's do not understand these formats
 - Web Application Scanners do not understand these formats
 - Until NTOSpider 6.0



Comments & Questions

www.NTOBJECTives.com
www.ManVsWebApp.com