Get off your AMF and don’t REST on JSON
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
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
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

@dan_kuykendall
Most Web Application Scanners

HTML4/5 & Javascript

PHP, ASP, C#, Python
Processes and validates data

Data servers, SQL, LDAP, BigData services

Presentation Layer

API

Business Logic

Data Layer

Web App1

Mobile App1

Web App2 (AJAX)

Mobile App2 (Binary)

Web Service

XSS

XSRF

SQLi

LDAP

Overflows Datatypes

JSON/REST/AMF

SOAP

Thin pass-through

@dan_kuykendall
Automated tools have fallen behind

The Widening Coverage Gap

Application Scanners

Application Complexity & Scanner Coverage

- Static Pages (HTML)
- CGI
- Scripted Pages
- Web 2.0 (JavaScript/AJAX)
- Application Frameworks
- Dynamic Application Security Scanners
- Network Scanners (known vuln checks)
- Re-architected for JavaScript
- Flash/Flex, Silverlight, JSON, REST, AMF, SOAP
- Web 3.0 & Mobile

Incremental improvements only

@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
SQL Injection 101
Hacking the database
Hacking Traditional WebApps - Traffic

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

```plaintext
request | response
--- | ---
GET /crosstraining/linkout.php?name=Pake 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; NP9)
Accept-Encoding: gzip, deflate
Host: www.webscantest.com
Cookie: SESSIONID_VULN_SITE=p5v5h7qp75ccqkj7bp9qep0e92
```

- Submitting a FORM, generates a POST requests

```plaintext
request | response
--- | ---
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; NP9)
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: www.webscantest.com
Pragma: no-cache
Cookie: SESSIONID_VULN_SITE=p5v5h7qp75ccqkj7bp9qep0e92
Content-Length: 76
```
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='%name%'
```

`%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=‘%name%’
```

%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=‘O’Brian’
```

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

  ```sql
  SELECT * from tPeople WHERE lastname='\%name\%
  ```

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

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

  ```sql
  SELECT * from tPeople WHERE lastname='O'Brian'
  ```

  invalid SQL statement

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

  ```sql
  SELECT * from tPeople WHERE lastname='O\'Brian'
  ```

@dan_kuykendall
Standard SQL Injection Example

• Often errors will be created and displayed

![Login page with SQL injection error](https://www.johndoe.com/login.php?user=admin'&password=Password1)

```
Invalid User
Error 1064: You have an error in your SQL syntax near '"' at line 1 of
SELECT * FROM accounts WHERE username='admin' AND password='Password1'
```

```
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
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/Actionscript Messaging Format
    ```xml
    &lt;null&gt;
    &lt;flex.messaging.messages.RemotingMessage source=operation=destination=time
    admin junk
    DSIdIBA981B7-5E1C-6007-6D98-2C3EF173C5AF0D3Endpointomy-amfIOB6
    ```

- **AJAX**: Asynchronous JavaScript and XML
  - **REST**: REpresentational State Transfer
    ```http
    http://example.com/catalog/item/17
    ```

- **JSON**: JavaScript Object Notation
  ```json
  {“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

```html
+ 'This content requires the Adobe Flash Player.'
+师事务所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
/svflash.cab">
    <param name="movie" value="testHelloWorld1.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#869ce7" />
    <param name="allowScriptAccess" value="sameDomain" />
    <embed src="testHelloWorld1.swf" quality="high" bgcolor="#869ce7"
        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">

</object>
</noscript>
</body>
</html>
```

@dan_kuykendall
Hacking AMF – Raw traffic

- Looks like binary garbage

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw</td>
<td></td>
</tr>
<tr>
<td>Params</td>
<td></td>
</tr>
<tr>
<td>Headers</td>
<td></td>
</tr>
<tr>
<td>Hex</td>
<td></td>
</tr>
<tr>
<td>AMF</td>
<td></td>
</tr>
</tbody>
</table>

```
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/3null/null

Oflex.messaging.messages.RemotingMessagesource.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

<table>
<thead>
<tr>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>/3</td>
</tr>
<tr>
<td>response method</td>
<td>null</td>
</tr>
<tr>
<td>data</td>
<td>array</td>
</tr>
<tr>
<td>Source</td>
<td>null</td>
</tr>
<tr>
<td>Body</td>
<td>array</td>
</tr>
<tr>
<td>[0] Operation</td>
<td>admin</td>
</tr>
<tr>
<td>[1] Operation</td>
<td>junk</td>
</tr>
<tr>
<td>Operation</td>
<td>sayHello</td>
</tr>
<tr>
<td>RemoteUsername</td>
<td>null</td>
</tr>
<tr>
<td>RemotePassword</td>
<td>null</td>
</tr>
<tr>
<td>Timestamp</td>
<td>number 0</td>
</tr>
<tr>
<td>Headers</td>
<td>map</td>
</tr>
<tr>
<td>TimeToLive</td>
<td>number 0</td>
</tr>
<tr>
<td>Destination</td>
<td>string HelloWorld</td>
</tr>
</tbody>
</table>
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 placeholder 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.

Don't 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.
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:** Asynchronous Javascript And XML
  - JSON is native for AJAX frameworks, including jQuery

- **Mobile apps use JSON**
•Serialized array – Shown nicely formatted

```json
{"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

```json
{"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;" } }}
```
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 placeholder 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.
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...
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
● 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":"3MN/31p3yaN5wWHUjebSGuMA1BScjKszFvAv6htgMwY==","bundleId":"com.zynga.words","ccs":"310260;T-Mobile;GSM","cct":2,"cctDetailed":"","clientDateTime":"4,248","cookie":{"maxAge":2592000,"name":"fc8211","value":"ref=1202221048"},"maxAge":2592000,"name":"vw","value":"ref=1202221048","maxAge":172800,"name":"lc","value":"147744"},"maxAge":86400,"name":"iuqiS7ZzRhvUej10-oTBREFA","value":"x"},{"maxAge":172800000,"name":"v","value":"ePU9ZJF4UidalJ1pc5SLA"},"crFarms":"","deviceFlags":0,"deviceId":"22c009a281dd2d30f110b9c63d95ac44f283d8f","encDeviceId":"1323d4+FSlguVoOQS1cVliAmN4M5H8Bo+Afi33vCwd=","ipAddress":"","noTrack":0,"placement":"","pubTargeting":"Millennial=1","publisher":"ql37ZzRhvUej10-oTBREFA","rvcR":"13942","type":"","userId":"","userAgentInfo":{"Build":"1.11.0.0942","Carrier":"T-Mobile","Density":"High","Device":"HTCVision","DeviceFamily":"htc_wwe","MCC":"310","MNC":"260","Platform":"Android","PlatformVersion":"2.3.7","ScreenWidth":"480x800","v":"1","webUserAgent":"Mozilla/5.0 (Linux; U; Android 2.3.7; en-us; HTC Vision Build/GR140) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1"},"zone":"095515:879139204689"}

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-Encodings: 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)

@dan_kuykendall
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

@dan_kuykendall
Hacking Mobile Apps: Bump

1. Bump Sent
2. Status ok
3. Status check
4. Bump matched
5. Confirm + data
6. Other user confirms
7. Status check
8. Other user data
9. Status check
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

● 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