



# Cleaning Up AJAX Development

---

**Lars Ewe, CTO & VP of Engineering**

- **Ajax Deployment**
- **What is AJAX?**
- **AJAX and the Same Origin Policy**
- **AJAX and Web App Security**
- **AJAX and Test Automation**
- **Vulnerability Examples**
- **AJAX Best Practices**
- **Q & A**



- **A**synchronous **J**avaScript **A**nd **X**ML
- **AJAX** allows for a new generation of more dynamic, more interactive, faster Web 2.0 applications, hidden useful background tasks.
- **AJAX** leverages existing technologies, such as Dynamic HTML (DHTML), Cascading Style Sheets (CSS), Document Object Model (DOM), JavaScript Object Notation (JSON), etc., and the (a)synchronous **XMLHttpRequest** (XHR)

- XHR allows for (a)synchronous server requests without the need for a full page reload
- XHR “downstream” payload can be
  - XML, JSON, HTML/JS snippets, plain text, serialized data, basically pretty much anything...
- Response often results in dynamic web page content changes through DOM modifications

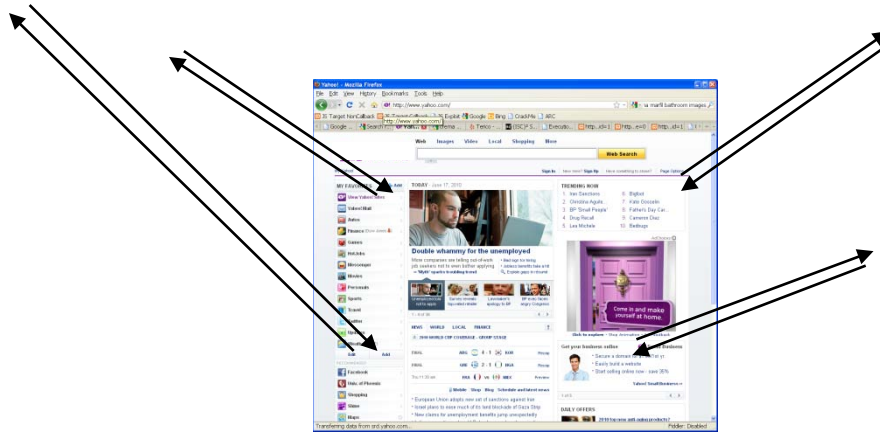
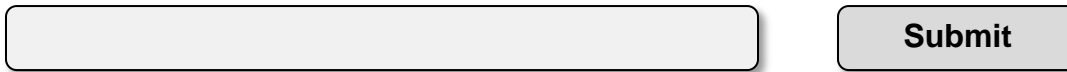


```
xhr = new XMLHttpRequest();
xhr.open("GET", "http://www.foobar.com", true);
xhr.onreadystatechange = processResponse;
xhr.send(null);

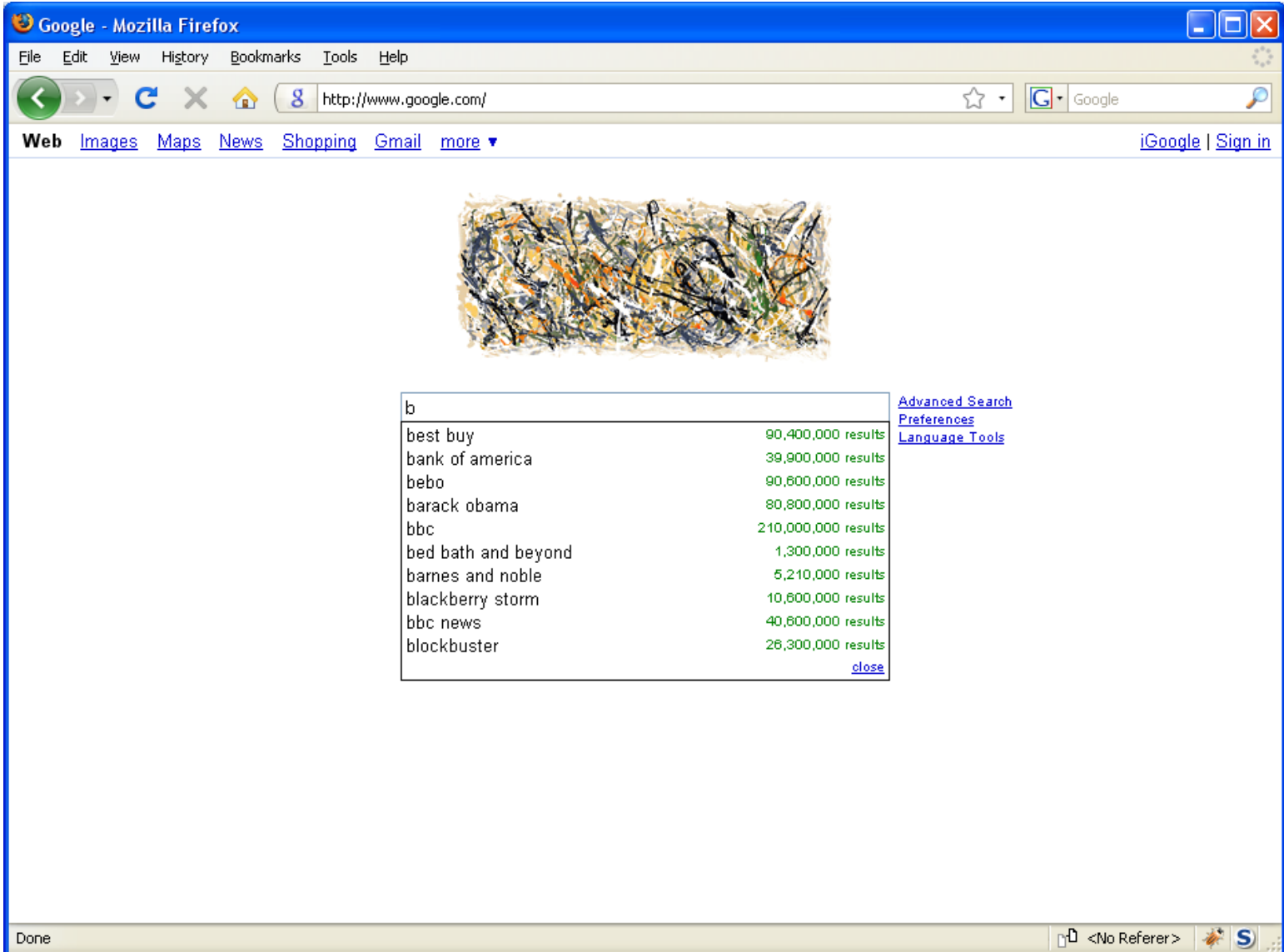
function processResponse () {
    if (xhr.readyState == 4) {
        if (request.status == 200) {
            response =
                xhr.responseText;
            .....
        }
    }
}
```

- Ajax is a new Web application development approach

- Apps no longer just fill and submit



# AJAX Example #1



The screenshot shows a Mozilla Firefox browser window with the Google homepage. The search bar contains the letter 'b', and a dropdown menu is open, displaying a list of search suggestions. The suggestions include 'best buy', 'bank of america', 'bebo', 'barack obama', 'bbc', 'bed bath and beyond', 'barnes and noble', 'blackberry storm', 'bbc news', and 'blockbuster'. Each suggestion is followed by the number of results in green text. To the right of the dropdown menu, there are links for 'Advanced Search', 'Preferences', and 'Language Tools'. The browser's status bar at the bottom shows 'Done' and '<No Referer >'.

Suggestion	Results
best buy	90,400,000 results
bank of america	39,900,000 results
bebo	90,600,000 results
barack obama	80,800,000 results
bbc	210,000,000 results
bed bath and beyond	1,300,000 results
barnes and noble	5,210,000 results
blackberry storm	10,600,000 results
bbc news	40,600,000 results
blockbuster	26,300,000 results

# AJAX Example #1



**Tamper Data - Ongoing requests**

Start Tamper Stop Tamper Clear Options Help

Filter  Show All

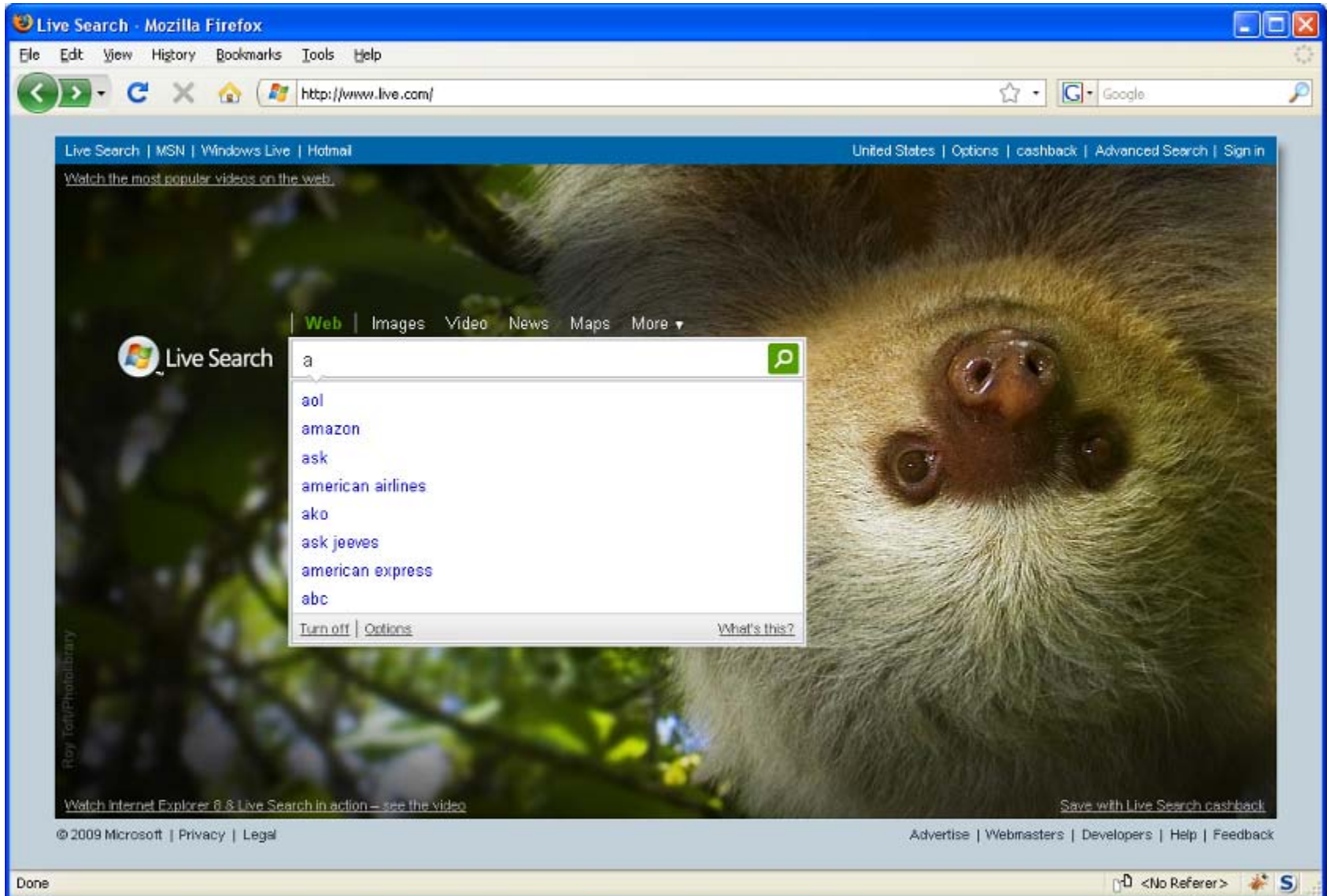
Time	URL	D...	Total ...	Size	M...	Status	
21:47:49.749	http://clients1.google.com/complete/search?hl=en&gl=us&q=b	210...	210 ms	220	GET	200	t... LOA...

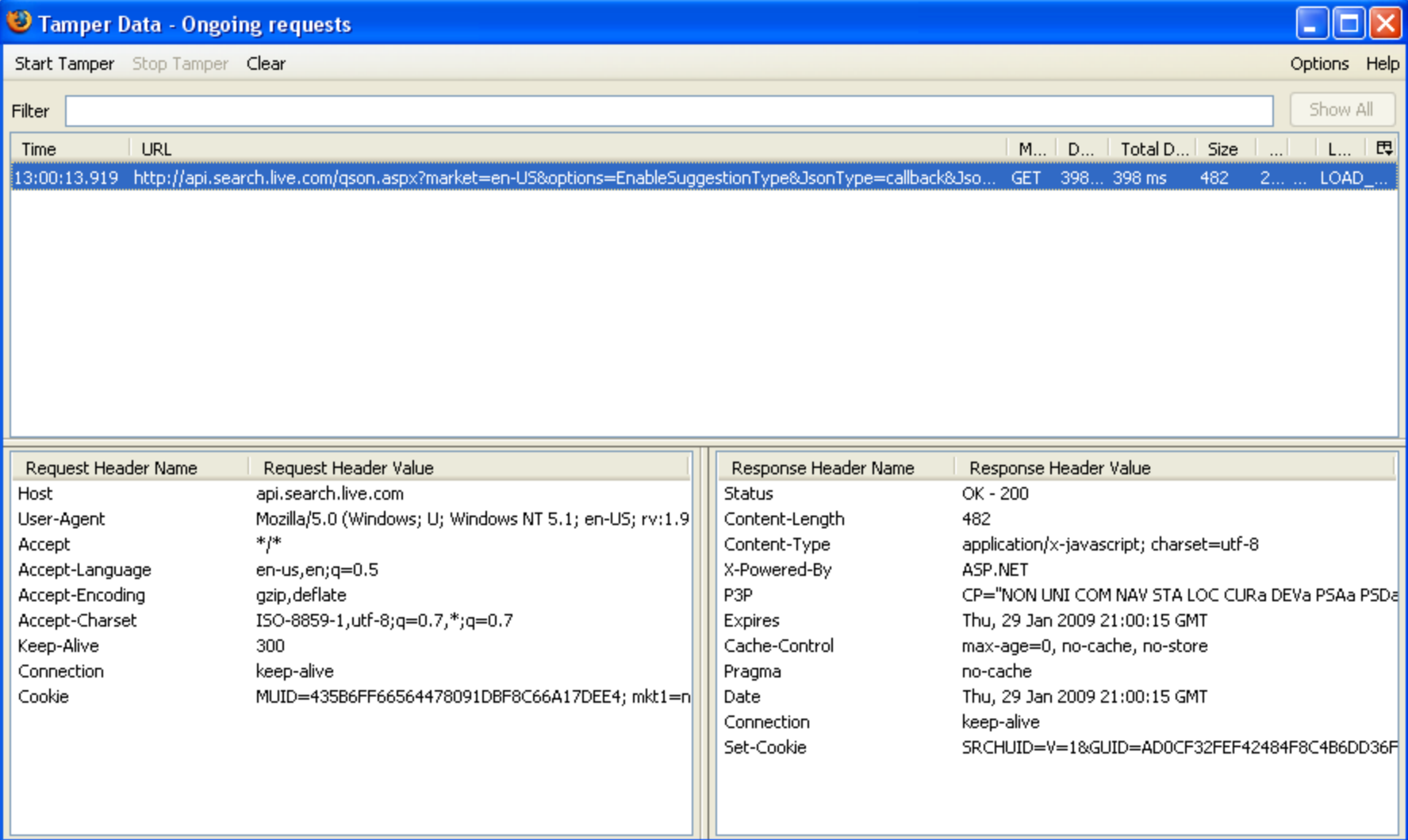
Request Header Name	Request Header Value	Response Header Name	Response Header Value
Host	clients1.google.com	Status	OK - 200
User-Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0...	Content-Type	text/javascript; charset=utf-8
Accept	*/*	Date	Thu, 29 Jan 2009 05:47:49 GMT
Accept-Language	en-us,en;q=0.5	Expires	Thu, 29 Jan 2009 06:47:49 GMT
Accept-Encoding	gzip,deflate	Cache-Control	public, max-age=3600
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7	Content-Encoding	gzip
Keep-Alive	300	Server	Auto-Completion Server
Connection	keep-alive	Content-Length	220
Cookie	PREF=ID=5511c27a765046e6:TM=1203085323:LM=1233...		



# AJAX Example #2



# AJAX Example #2



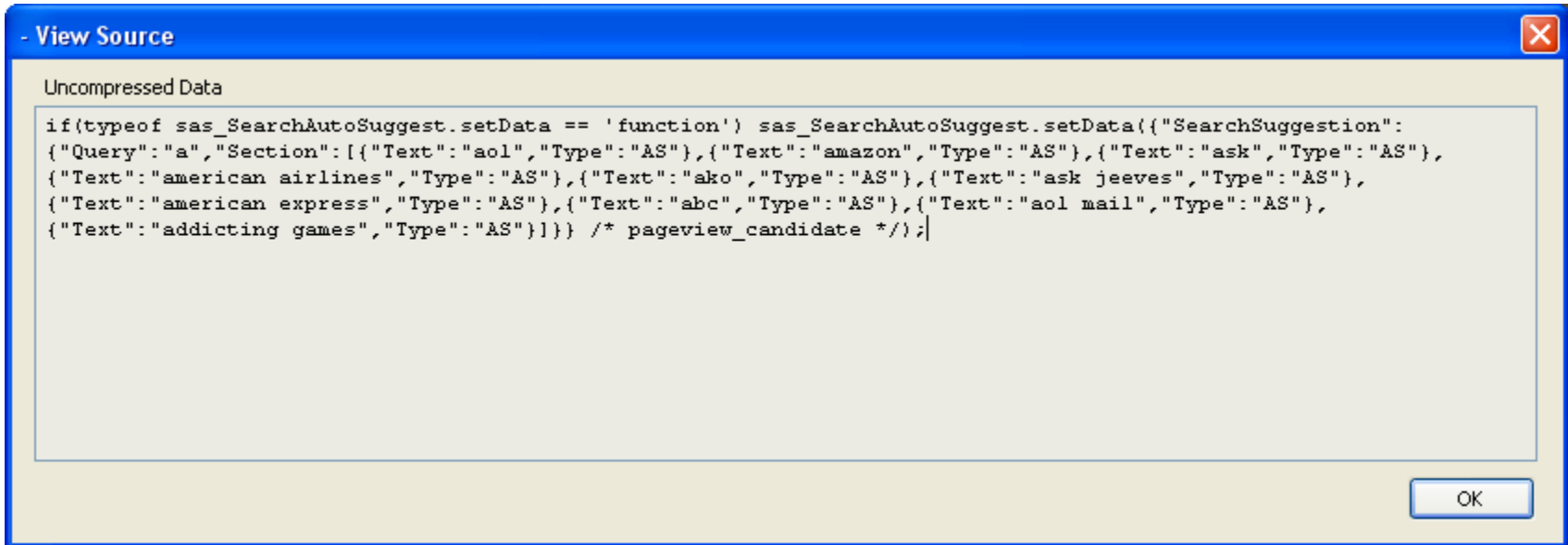
Tamper Data - Ongoing requests

Start Tamper Stop Tamper Clear Options Help

Filter  Show All

Time	URL	M...	D...	Total D...	Size	...	L...	
13:00:13.919	http://api.search.live.com/qson.aspx?market=en-US&options=EnableSuggestionType&JsonType=callback&Jso...	GET	398...	398 ms	482	2...	...	LOAD_...

Request Header Name	Request Header Value	Response Header Name	Response Header Value
Host	api.search.live.com	Status	OK - 200
User-Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9	Content-Length	482
Accept	*/*	Content-Type	application/x-javascript; charset=utf-8
Accept-Language	en-us,en;q=0.5	X-Powered-By	ASP.NET
Accept-Encoding	gzip,deflate	P3P	CP="NON UNI COM NAV STA LOC CURa DEVa PSAa PSDe
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7	Expires	Thu, 29 Jan 2009 21:00:15 GMT
Keep-Alive	300	Cache-Control	max-age=0, no-cache, no-store
Connection	keep-alive	Pragma	no-cache
Cookie	MUID=435B6FF66564478091DBF8C66A17DEE4; mkt1=n	Date	Thu, 29 Jan 2009 21:00:15 GMT
		Connection	keep-alive
		Set-Cookie	SRCHUID=v=1&GUID=AD0CF32FEF42484F8C4B6DD36F



The screenshot shows a 'View Source' dialog box with a blue title bar and a close button. The main area is titled 'Uncompressed Data' and contains the following JavaScript code:

```
if(typeof sas_SearchAutoSuggest.setData == 'function') sas_SearchAutoSuggest.setData({"SearchSuggestion":  
{"Query": "a", "Section": [{"Text": "aol", "Type": "AS"}, {"Text": "amazon", "Type": "AS"}, {"Text": "ask", "Type": "AS"},  
{"Text": "american airlines", "Type": "AS"}, {"Text": "ako", "Type": "AS"}, {"Text": "ask jeeves", "Type": "AS"},  
{"Text": "american express", "Type": "AS"}, {"Text": "abc", "Type": "AS"}, {"Text": "aol mail", "Type": "AS"},  
{"Text": "addicting games", "Type": "AS"}]}) /* pageview_candidate */);
```

An 'OK' button is located at the bottom right of the dialog box.



The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://demo.script.aculo.us/ajax/autocompleter_customized`. The page title is "script.aculo.us - web 2.0 javascript demos". The main content area features a green header with the "script.aculo.us" logo and navigation links for "back to site" and "demos". A yellow box contains the text: "This site demonstrates the **Ruby on Rails** AJAX helpers, which use script.aculo.us effects, drag-and-drop and auto completion."

The "ajax auto completion demo" section shows a "To:" label above a text input field containing the letter "a". A dropdown menu is open, listing several email addresses with their names in bold: "Ada Noel", "Adlai Cathy", "Adrian Audrey", "Adrian Clyde", "Adrian Ramneek", "Adrienne Amos", "Adrienne Conrad", and "Agatha Lesley".

To the right, a "rails demos" section lists several links: "script.aculo.us helpers", "Autocompleting text fields (basic)", "Autocompleting text fields (customized)", "Shopping cart", "Sortable elements", "New AJAX features", "Error handling", and "Update element helper".

Below the links is a red "RAILS" logo and a paragraph of text: "Rails is a full-stack, open-source web framework in Ruby for writing real-world applications with joy and less code than most frameworks spend doing XML sit-ups. That quite sums it up. So, if you're still working late hours writing definitions of what's in your database because the framework you use works against and not for you -- do yourself a favour, and check out [Ruby on Rails](#)."

At the bottom of the page, there is a "View source" link and a footer with copyright information: "© 2005-2007 thomas fuchs license mir.aculo.us". The browser's status bar at the bottom shows "Done" and a security warning: "<No Referer>".

# AJAX Example #3



**Tamper Data - Ongoing requests**

Start Tamper Stop Tamper Clear Options Help

Filter  Show All

Time	URL	M...	D...	Total D...	Size	...	Load Flags
13:09:21.218	http://demo.script.aculo.us/ajax/auto_complete_for_message_to	POST	326...	326 ms	-1	2...	t... LOAD_BYPASS_C...
13:09:21.547	http://demo.script.aculo.us/demos/images/contacts/5.jpg	GET	139...	139 ms	-1	4...	t... LOAD_FROM_CA...
13:09:21.548	http://demo.script.aculo.us/demos/images/contacts/8.jpg	GET	205...	205 ms	-1	4...	t... LOAD_FROM_CA...
13:09:21.549	http://demo.script.aculo.us/demos/images/contacts/3.jpg	GET	210...	210 ms	-1	4...	t... LOAD_FROM_CA...
13:09:21.550	http://demo.script.aculo.us/demos/images/contacts/1.jpg	GET	213...	213 ms	-1	4...	t... LOAD_FROM_CA...

Request Header Name	Request Header Value	Response Header Name	Response Header Value
Host	demo.script.aculo.us	Status	OK - 200
User-Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1...	Server	nginx/0.5.33
Accept	text/html,application/xhtml+xml,application/xml;q=0.9...	Date	Thu, 29 Jan 2009 21:09:23 GMT
Accept-Language	en-us,en;q=0.5	Content-Type	text/html; charset=utf-8
Accept-Encoding	gzip,deflate	Transfer-Encoding	chunked
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7	Connection	keep-alive
Keep-Alive	300	Status	200 OK
Connection	keep-alive	X-Runtime	0.21905
X-Requested-With	XMLHttpRequest	Etag	"ad2a0b746d83f962da315e12acaaaadd"
X-Prototype-Version	1.3.0	Cache-Control	private, max-age=0, must-revalidate
Content-Type	application/x-www-form-urlencoded; charset=UTF-8	Content-Encoding	gzip
Content-Length	20		
Cookie	__scriptaculous-demos_session=BAh7BiIKZmxhc2hJQzo...		
Pragma	no-cache		
Cache-Control	no-cache		
POSTDATA	message%5Bto%5D=a&_ =		



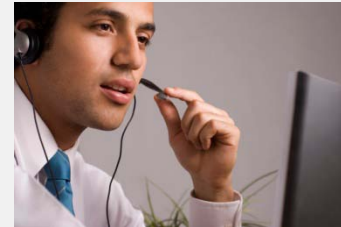
- **Cenzic CTS (SaaS): ~30% of recently tested applications use AJAX**
- **>50% AJAX developer growth year-over-year – Evans Data**
- **~3.5 million AJAX developers worldwide – Evans Data**
- **60% of new application projects will use Rich Internet Application (RIA) technologies such as AJAX within the next three years – Gartner**

- **Same origin policy is a key browser security mechanism**
  - **To prevent any cross-domain data leakage, etc.**
  - **With JavaScript it doesn't allow JavaScript from origin A to access content / data from origin B**
  - **Origin refers to the domain name, port, and protocol**
- **In the case of XHR, the same origin policy does not allow for any cross-domain XHR requests**
  - **Developers often don't like this at all!**

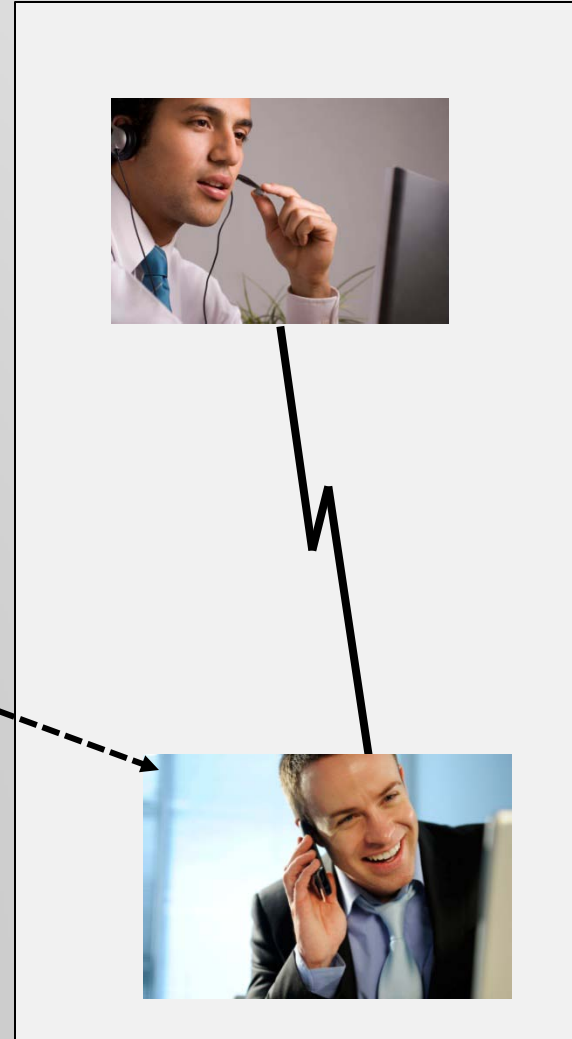
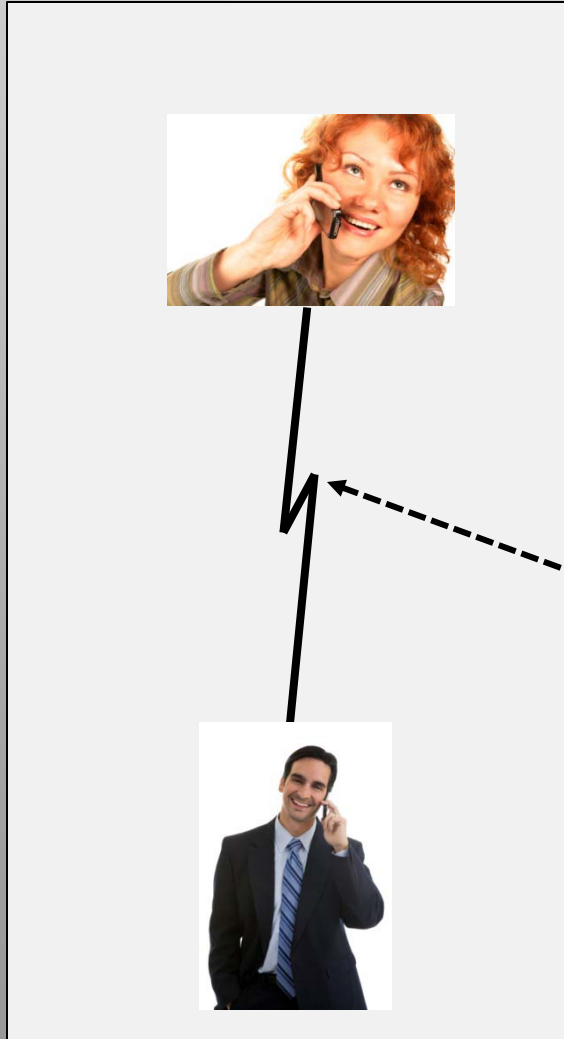
# Same Origin Policy



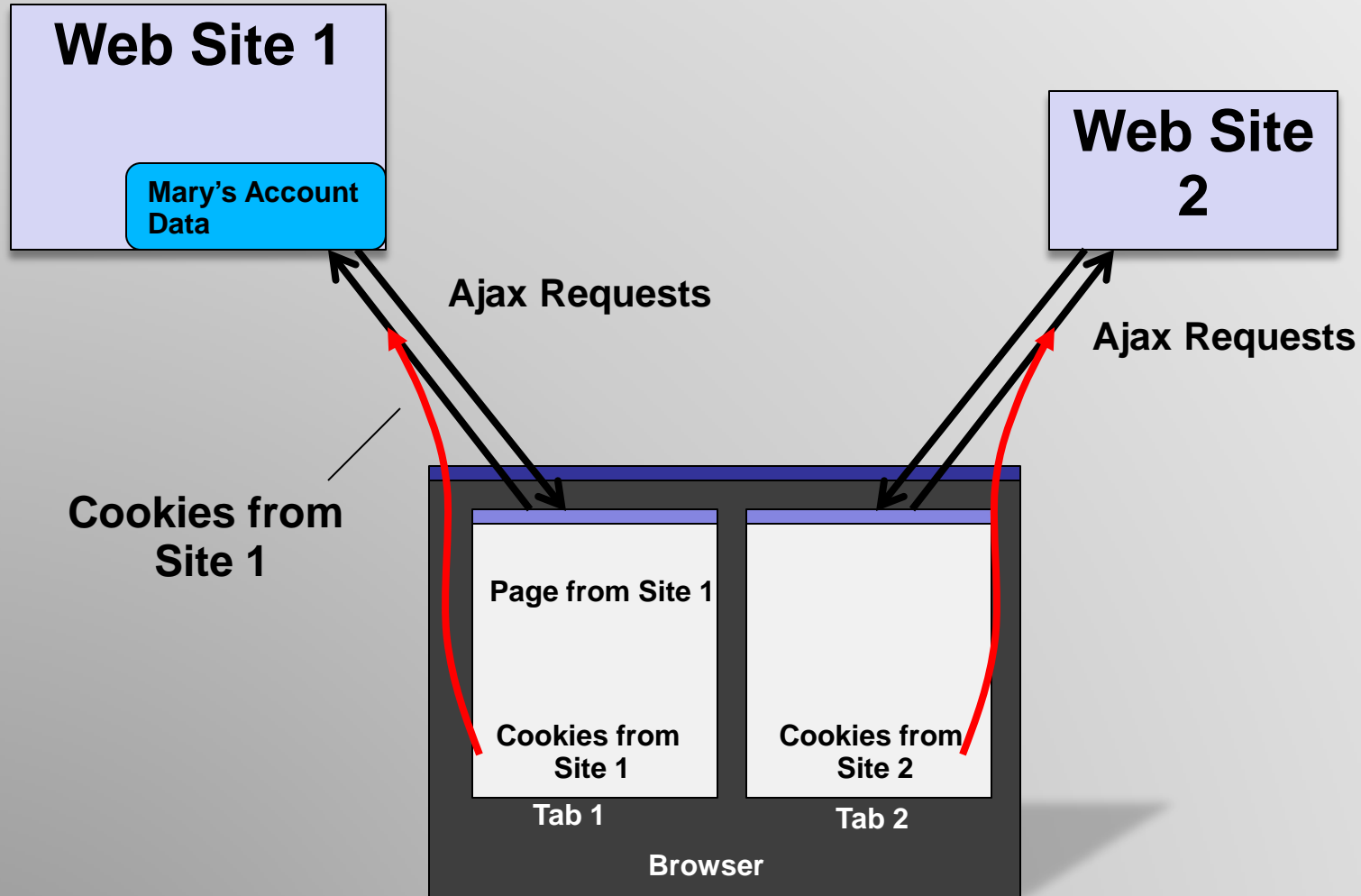
**We expect a private conversation**



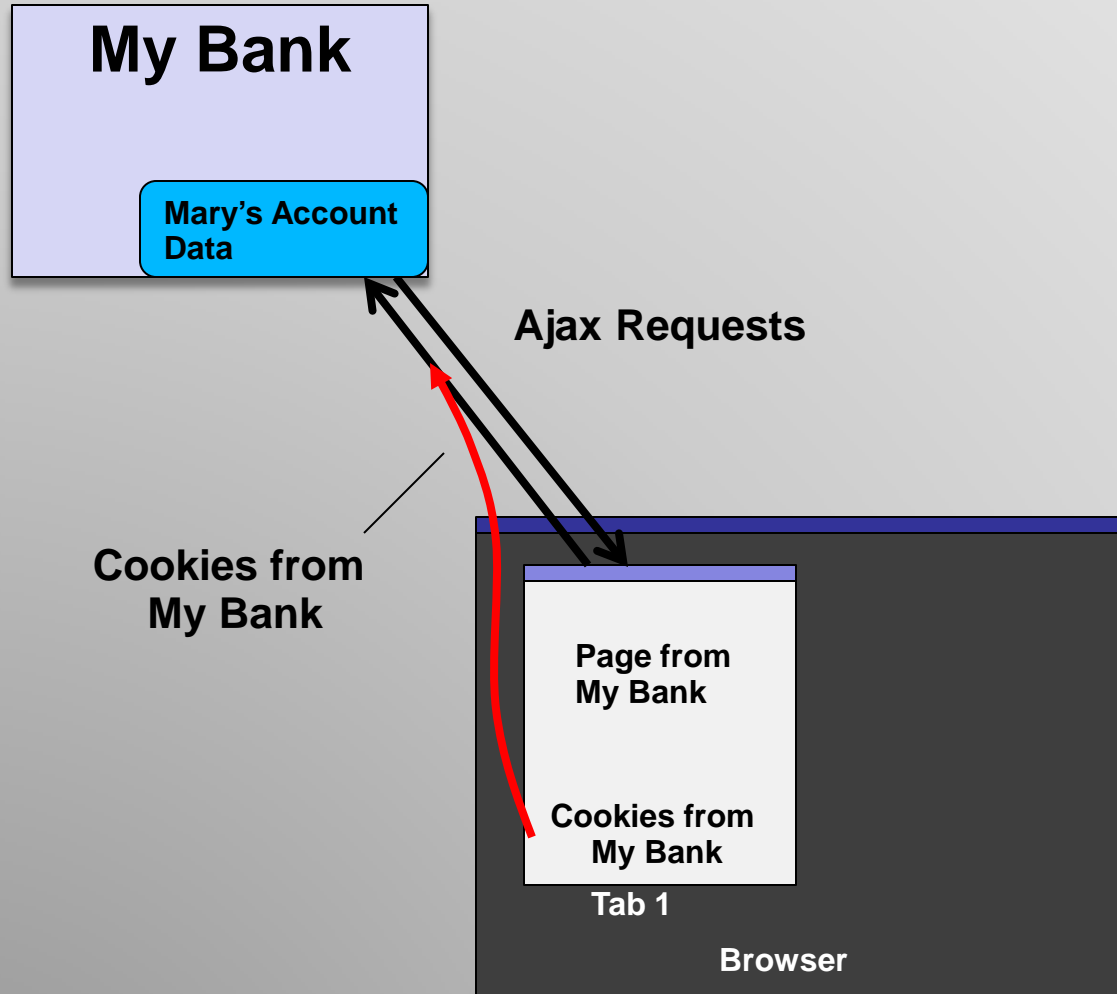
# Same Origin Policy



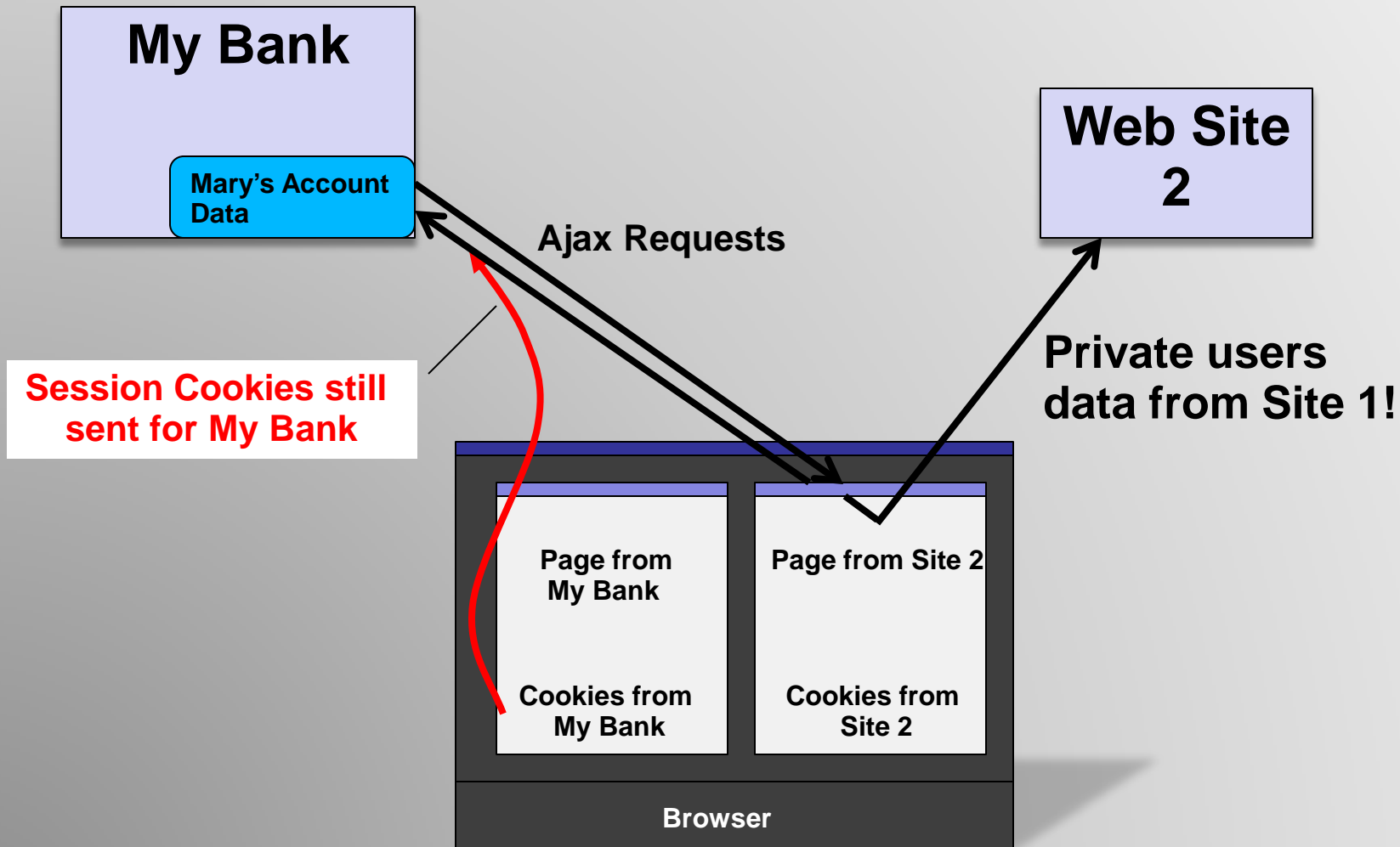
# Same Origin Policy



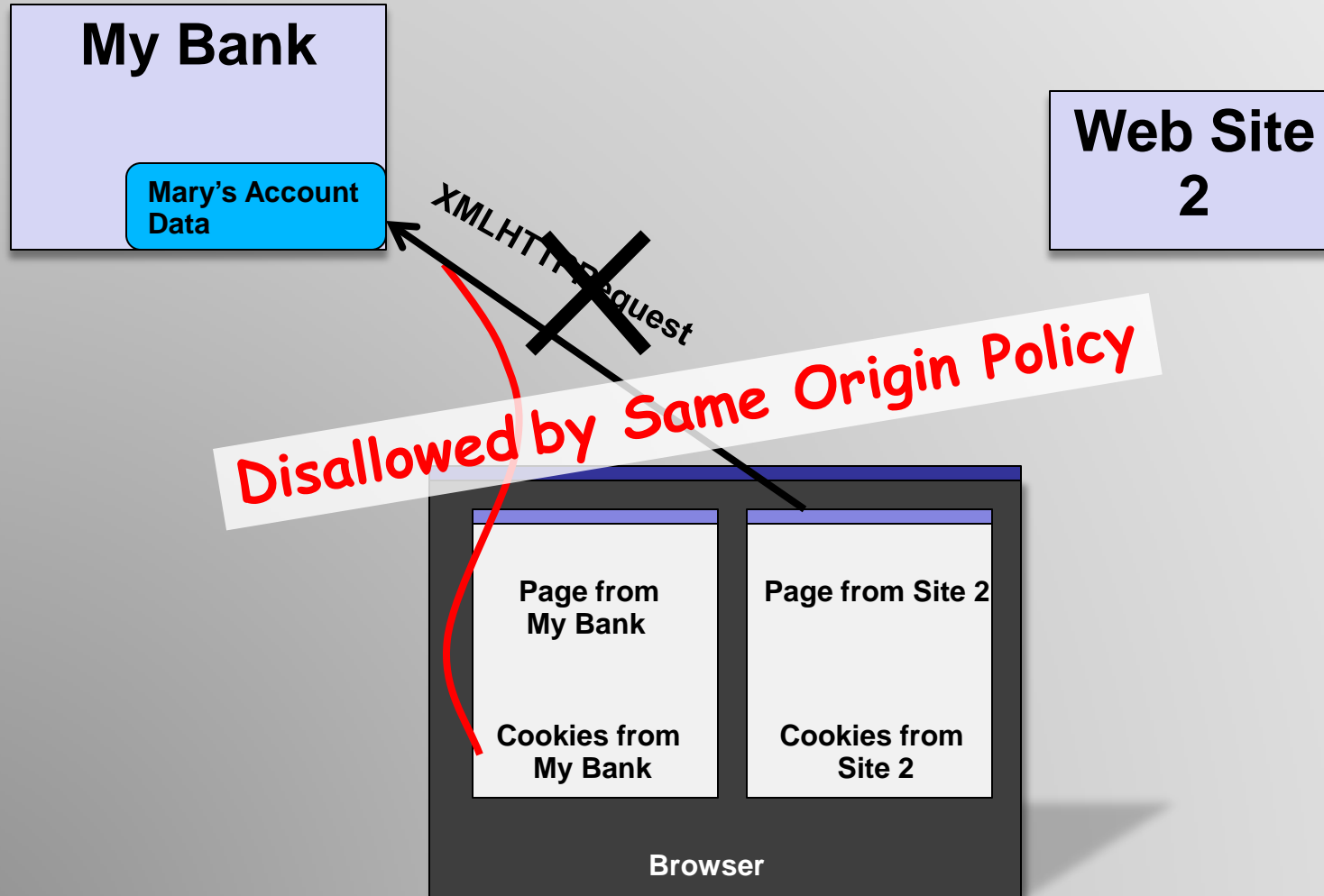
# Without Same Origin Policy



# Without Same Origin Policy



# With Same Origin Policy



- Mashups are a legitimate use and need for cross-domain Ajax processing



**Cross-domain access is often still implemented by various means, such as**

- **Open / Application proxies**
- **Flash & Java Applets (depending on `crossdomain.xml` config)**
  - **E.g. `FlashXMLHttpRequest` by Julien Couvreur**
- **RESTful web service with JavaScript callback and JSON response**
  - **E.g. `JSONscriptRequest` by Jason Levitt**

- **AJAX frameworks often categorized as either “Client” or “Proxy/Server” framework**
- **“Proxy/Server” frameworks often result in unintended method / functionality exposure**
- **Beware of any kind of “Debugging mode” (e.g. Direct Web Remoting (DWR) debug = true)**
- **Remember: Attackers can easily “fingerprint” AJAX frameworks**

# Direct Web Remoting

- Home
- Download
- Tutorials and Examples ...
  - Index
  - Who is using DWR
- Client Side ...
  - Index
  - DWR and TIBCO
  - engine.js ...
    - Index
    - Caching engine.js
    - Call Batching
    - Call Ordering
    - Errors and Timeouts
    - Remoting Hooks
    - Remoting Options
  - gi.js
  - util.js ...
    - Index
    - List Manipulation
    - \$()
    - Table Manipulation
    - addOptions
    - addRows
    - byId()
    - escapeHtml
    - getText

## Using debug/test mode

You put DWR into debug/test mode by adding the following parameter:

```
<init-param>
  <param-name>debug</param-name>
  <param-value>true</param-value>
</init-param>
```

DWR will generate test pages for each of the allowed classes (see dwr.xml below) in debug mode. These can be very useful in seeing what DWR can do and how it works. This mode can also alert you to problems like javascript reserved word clashes or overloading problems.

However this mode should not be used in live deployment as it could give an attacker a lot of information about the services that you export. If you have designed your website properly then this extra information will not help an attacker exploit your website however it is generally wise not to give anyone a route map to exploit any mistakes you might have made.

DWR is provided 'as is', without any warranty, so the security of your website is your responsibility. Please take care to keep it secure.

- **AJAX potentially increases the attack surface**
  - **More “hidden” calls mean more potential security holes**
- **AJAX developers sometimes pay less attention to security, due to it’s “hidden” nature**
  - **Basically the old mistake of security by obscurity**
- **AJAX developers sometimes tend to rely on client side validation**
  - **An approach that is just as flawed with or without AJAX**



- **Mash-up calls / functionality are often less secure by design**
  - **3<sup>rd</sup> party APIs (e.g. feeds, blogs, search APIs, etc.) are often designed with ease of use, not security in mind**
  - **Mash-ups often lack clear security boundaries (who validates, who filters, who encodes / decodes, etc.)**
  - **Mash-ups often result in untrusted cross-domain access workarounds**
- **AJAX sometimes promotes dynamic code (JavaScript) execution of untrusted response data**

**AJAX adds to the problem** of well-known Web application vulnerabilities.



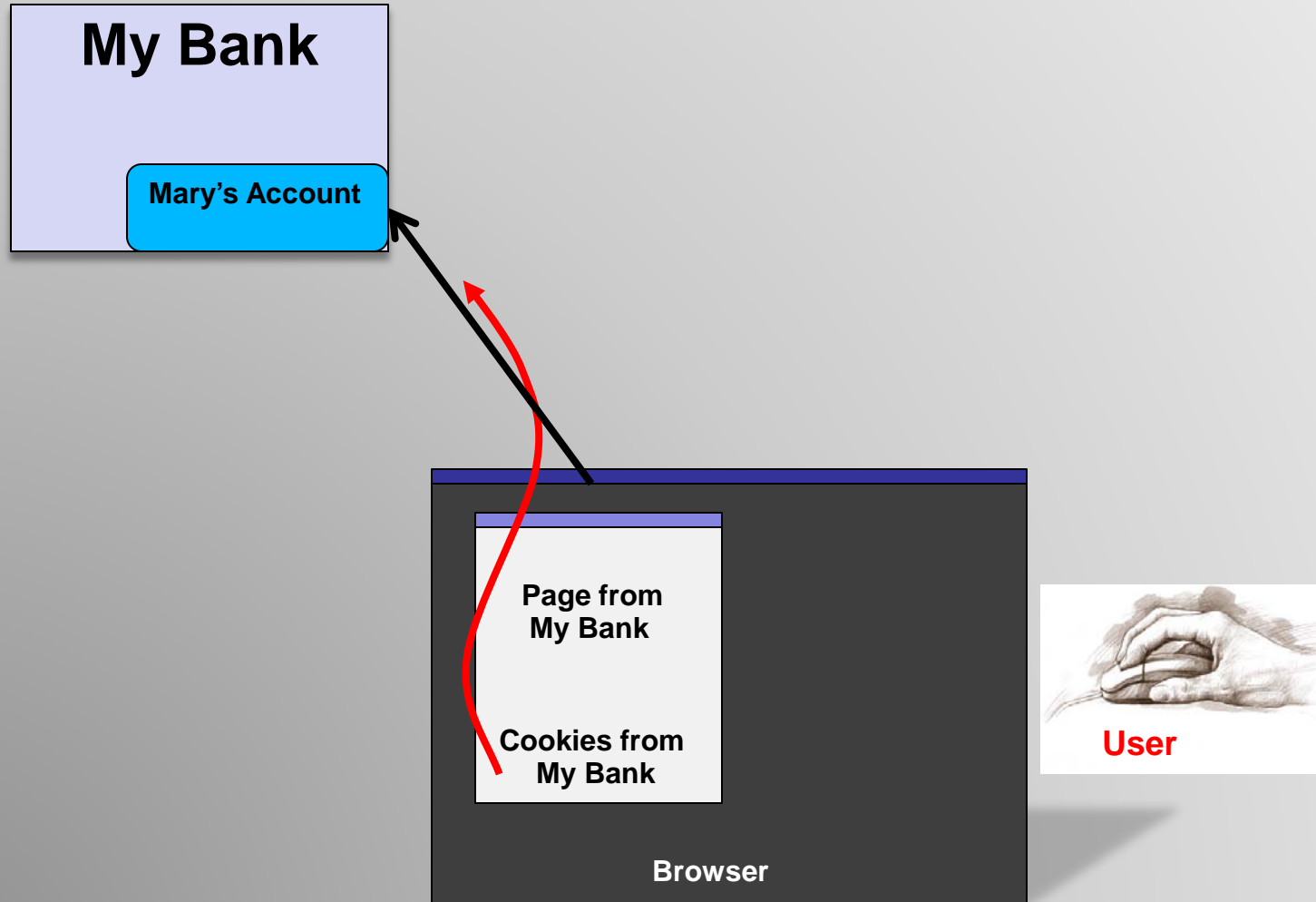
- **Spidering is more complex than just processing ANCHOR HREF's; various events need to be simulated (e.g. mouseover, keydown, keyup, onclick, onfocus, onblur, etc.)**
- **Timer events and dynamic DOM changes need to be observed**
- **Use of non-standard data formats for both requests and responses make injection and detection hard to automate**
- **Page changes after XHR requests can sometimes be delayed**
- **In short, you need to have browser like behavior (JavaScript engine, DOM & event management, etc.)**

- **What is it?:** Database contents are compromised or disclosed by the use of specially crafted input that manipulates SQL Query Logic.
- **Root Cause:** Failure to properly scrub, reject, or escape domain-specific SQL characters from an input vector.
- **Impact:** Data confidentiality, integrity, and availability with the ability to read, modify, delete, or even drop database tables.
- **Solution:** Use parameterized SQL statements. Define accepted character-sets for input vectors, and enforce these white lists rigorously. Force input to conform to specific patterns when other special characters are needed: dd-mm-yyyy. Validate data length of all inputs.

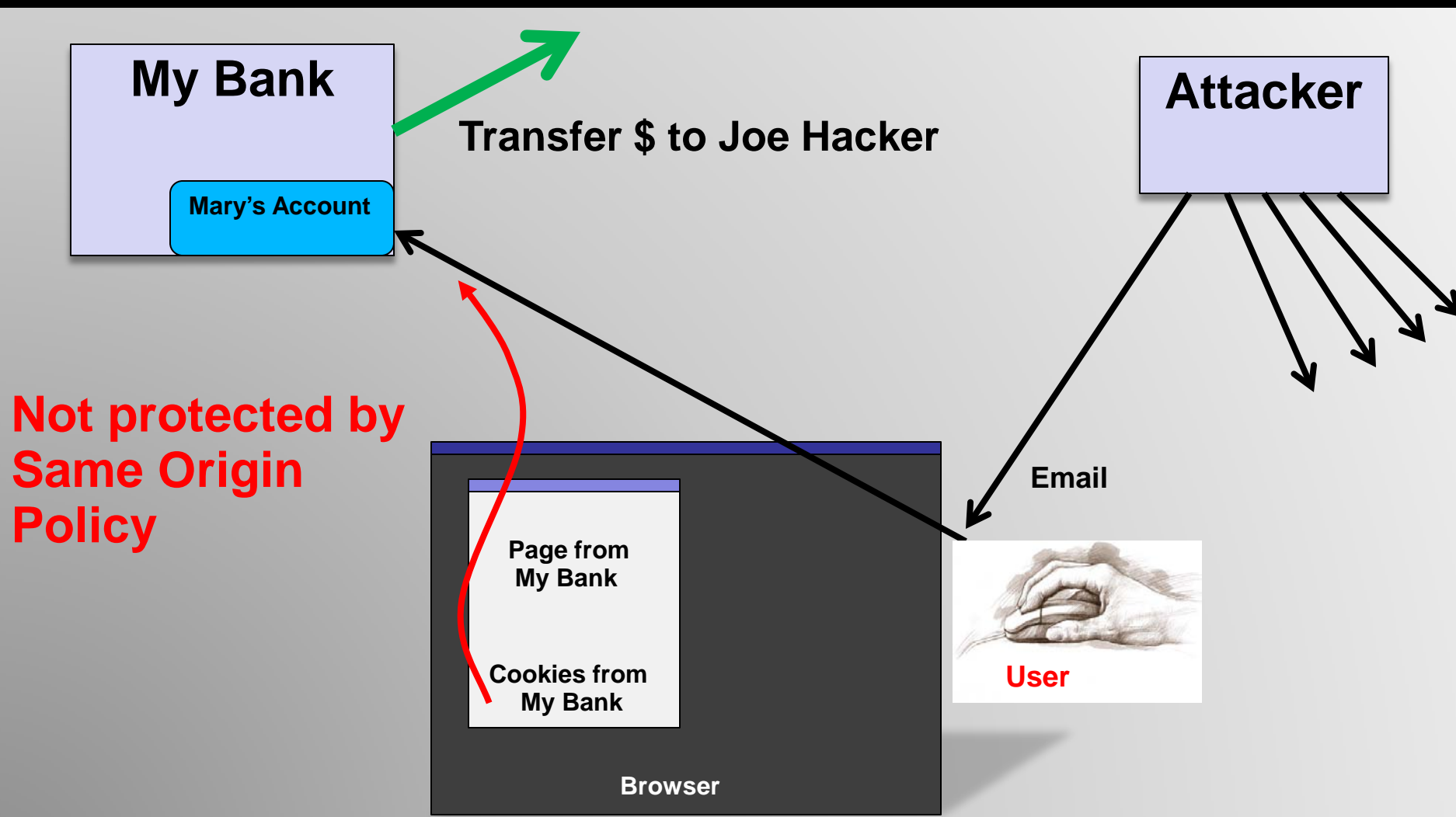
- **What is it?:** The Web Application is used to store, transport, and deliver malicious active content to an unsuspecting user.
- **Root Cause:** Failure to proactively reject or scrub malicious characters from input vectors.
- **Impact:** Persistent XSS is stored and executed at a later time, by a user. Allows cookie theft, credential theft, data confidentiality, integrity, and availability risks. Browser Hijacking and Unauthorized Access to Web Application is possible using existing exploits.
- **Solution:** A global as well as Form and Field specific policy for handling untrusted content. Use white lists and regular expressions to ensure input data conforms to the required character set, size, and syntax.

- **What is it?:** Basic Web Application session management behavior is exploited to make legitimate user requests without the user's knowledge or consent.
- **Root Cause:** Basic session id management that is vulnerable to exploitation (e.g. cookie-based).
- **Impact:** Attackers can make legitimate Web requests from the victim's browser without the victim's knowledge or consent, allowing legitimate transactions in the user's name. This can result in a broad variety of possible exploits.
- **Solution:** Enhance session management by using non-predictable "nonce" or other unique one-time tokens in addition to common session identifiers, as well as the validation of HTTP Referrer headers.

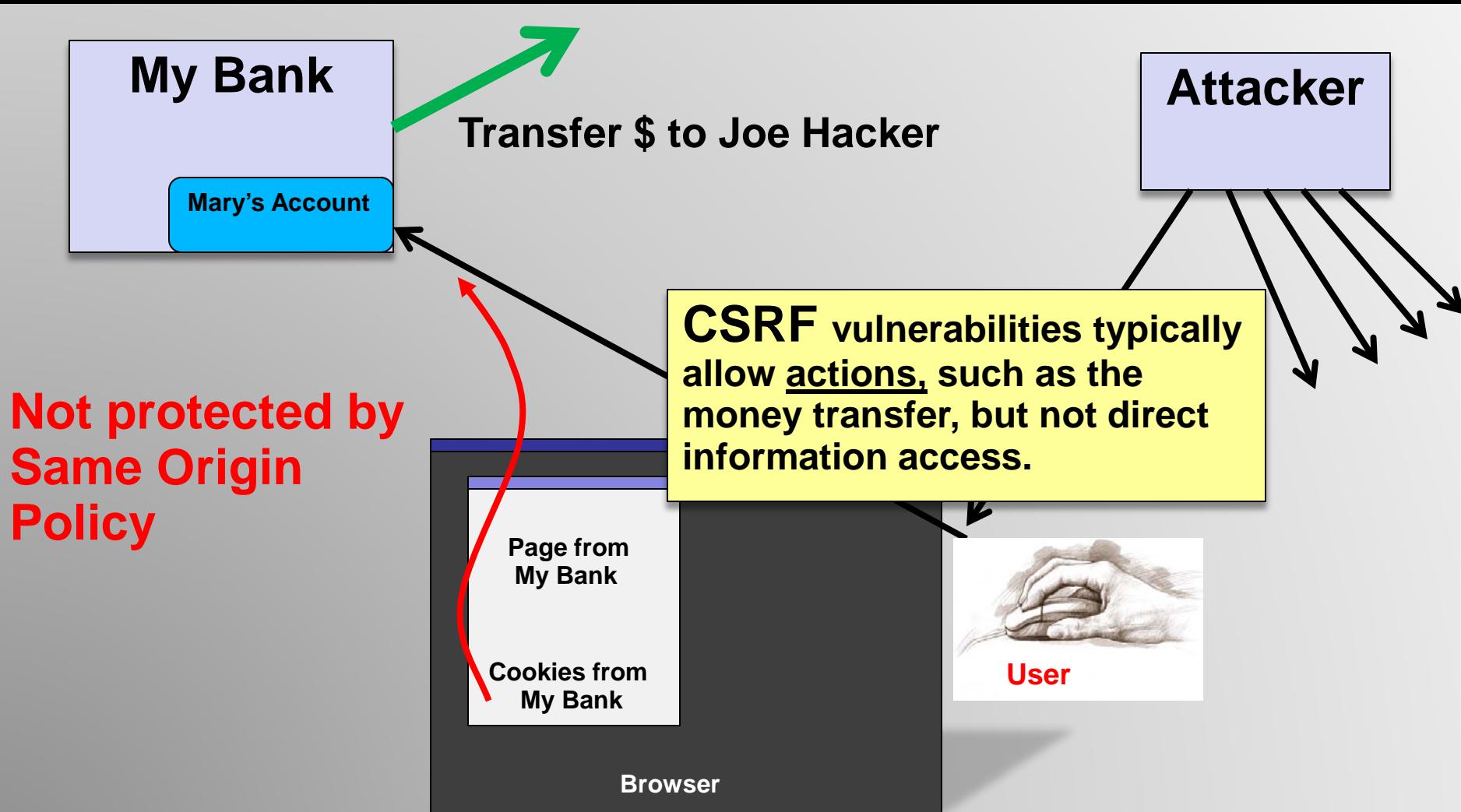
# Cross Site Request Forgery



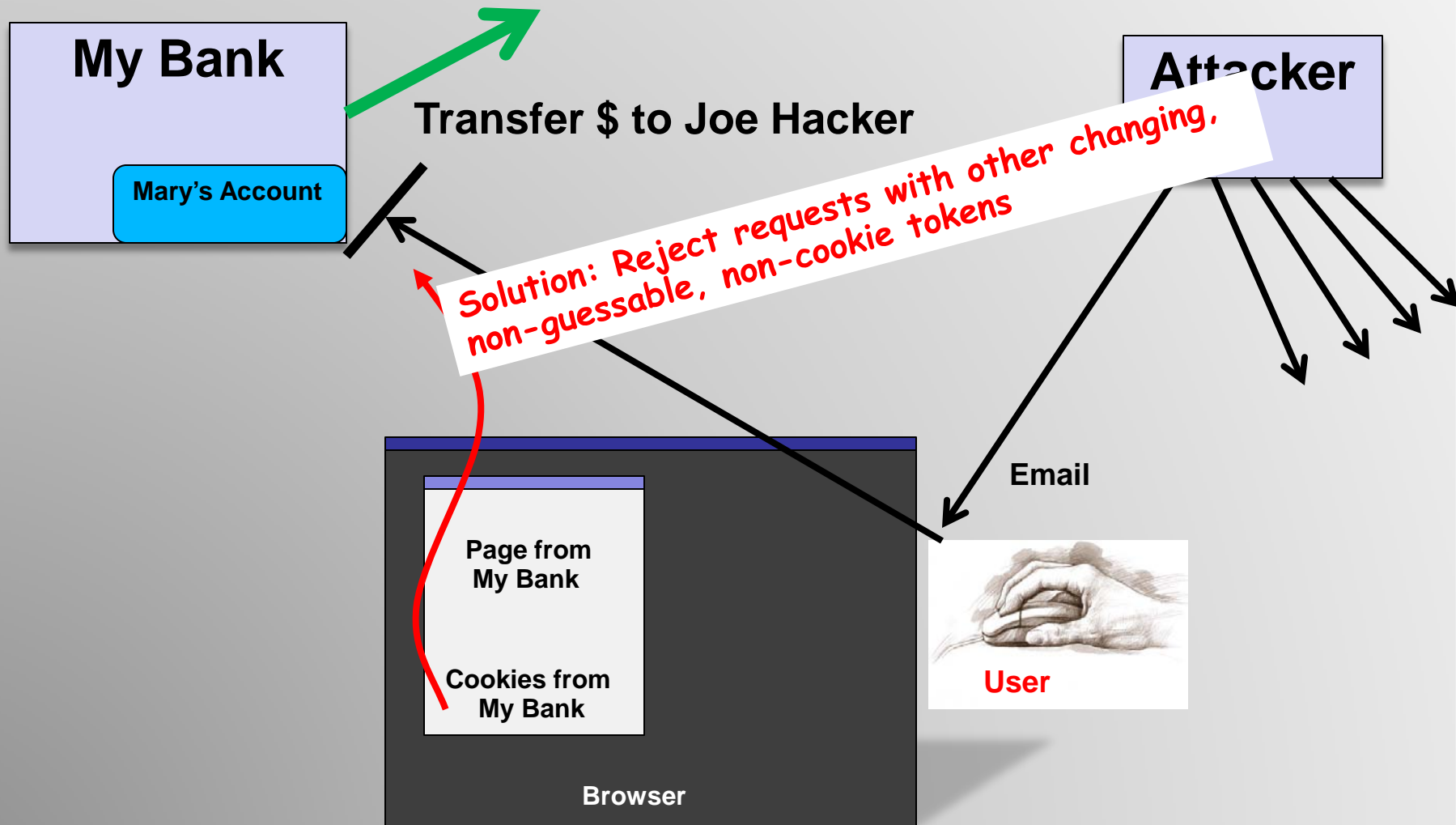
# Cross Site Request Forgery



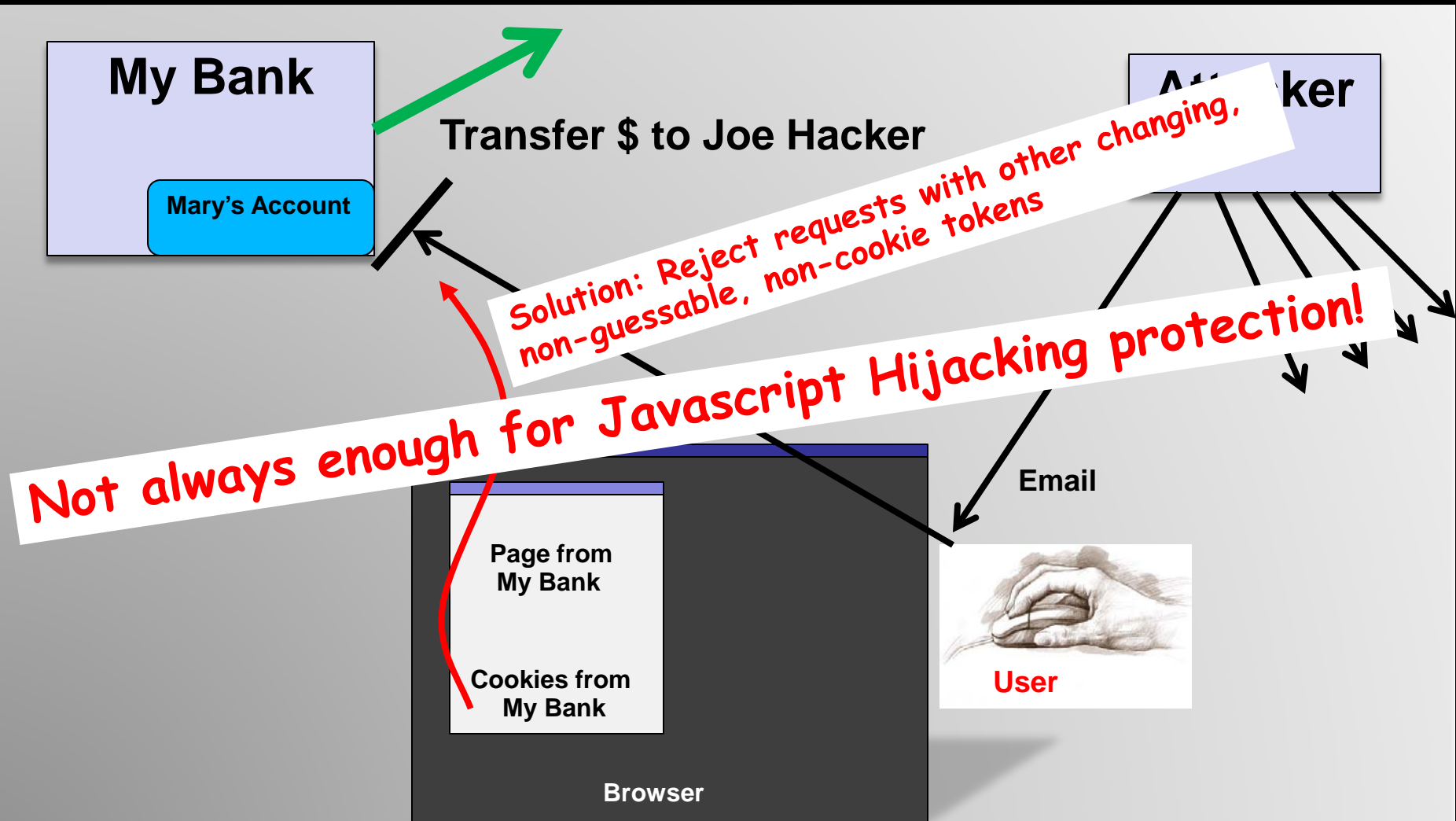
# Cross Site Request Forgery



# Cross Site Request Forgery



# Cross Site Request Forgery



- **What is it?:** An attack vector specific to JavaScript messages. Confidential data contained in JavaScript messages is being accessed by the attacker despite the browser's same origin policy.
- **Root Cause:** The `<script>` tag circumvents the browser's same origin policy. In some cases the attacker can set up an environment that lets him or her observe the execution of certain aspects of the JavaScript message. Examples: Override/implement native Object constructors (e.g. Array) or callback function. This can result in access to the data loaded by the `<script>` tag.
- **Impact:** Data confidentiality, integrity, and availability with the ability to access any confidential data transferred by JavaScript.
- **Solution:** Implement CSRF defense mechanisms; prevent the direct execution of the JavaScript message. Wrap your JavaScript with non-executable pre- and suffixes that get stripped off prior to execution of the sanitized JavaScript. Example: Prefix your JavaScript with `while(1);`

### Attacker code (override Array constructor)

### Attacker's Client Code

```
<script>
function Array(){
/* Put hack to access Array elements here */
}
</script>
```

### AJAX Call

```
<script src="http://targetsite.com/getacctinfo.php "
type="text/javascript"></script>
```

### Example AJAX response

```
[ "foo1", "bar1" ], [ "foo2", "bar2" ]
```

### Attacker code (implement callback)

### Attacker's Client Code

```
<script>
function callback(foo){
/* Put hack to access callback data here */
}
</script>
```

### AJAX Call

```
<script src="http://targetsite.com/getacctinfo.php"
type="text/javascript"></script>
```

### Example AJAX response

```
callback(["foo","bar"]);
```

# Preventing JavaScript Hijacking

## A simple code example



```
var object;  
var xhr = new XMLHttpRequest();  
xhr.open("GET", "/object.json",true);  
xhr.onreadystatechange = function () {  
    if (xhr.readyState == 4) {  
        var txt = xhr.responseText;  
        if (txt.substr(0,9) == "while(1);") {  
            txt = txt.substring(10);  
            Object = eval("(" + txt + ")");  
        }  
    }  
};  
xhr.send(null);
```

Remember, the attacker cannot sanitize the JavaScript, since they are relying on the `<script>` tag

Also see

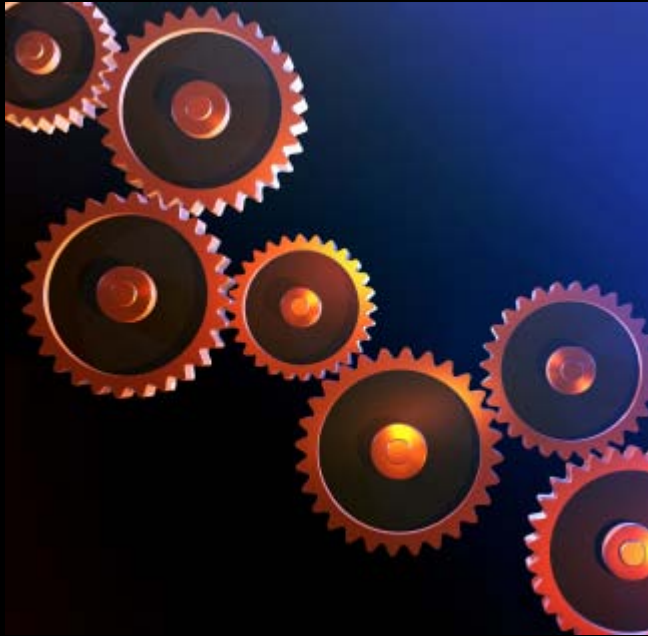
[http://www.fortifysoftware.com/servlet/downloads/public/JavaScript\\_Hijacking.pdf](http://www.fortifysoftware.com/servlet/downloads/public/JavaScript_Hijacking.pdf)

- **Consider avoiding JavaScript payloads**
- **Don't use HTTP GET for “upstream”**
- **Prefix “downstream” JavaScript with `while(1);`**
- **Avoid / Limit the use of dynamic code / `eval()`**
- **Enforce CSRF Protections comprehensively**
- **Be extremely careful when circumventing same origin policy**

**Pretty much all the usual Web app security best practices apply:**

- **Analyze and know your security boundaries and attack surfaces**
- **Do not rely on client-side security measures**
  - **Always implement strong server side input & parameter validation (black & whitelisting)**
  - **Test against a robust set of evasion rules**
  - **Remember: The client can never be trusted!**
- **Assume the worst case scenario for all 3<sup>rd</sup> party interactions**
  - **3<sup>rd</sup> parties can inherently not be trusted!**

- **Escape special characters before sending them to the browser (e.g. `<` to `&lt;` ;)**
- **Leverage HTTPS for sensitive data, use `HTTPOnly` & `Secure` cookie flags**
- **Use parameterized SQL for any DB queries**
- **Also see [owasp.org](http://owasp.org) and OWASP dev guide**



## Cenzic Product Suite

---

**Software & SaaS products that detect vulnerabilities**

# Who Is CenZic?



- Founded in June 2000 - Privately held
- CenZic provides **software & SaaS products** to **protect Web applications against hacker attacks**
  - **Software** (CenZic Hailstorm Enterprise ARC & CenZic Hailstorm Professional)
  - **Managed Service** (CenZic ClickToSecure Managed)
  - **Services** (training courses and assessment methodology)
- Stateful Assessment technology makes CenZic unique in the Web vulnerability scanning market
- Winner of numerous industry awards



“CenZic emulates a hacker and looks for real-time responses at the browser level. This approach provides an accurate solution with **less than 1% false positives.**”

**Charles Kolodgy**  
IDC



## Software

### Hailstorm Enterprise ARC



Cenzic Hailstorm Enterprise ARC  
Download Datasheet »

### Hailstorm Professional



Cenzic Hailstorm Pro  
Download Datasheet »

## SaaS / Cloud

### ClickToSecure Managed



Cenzic ClickToSecure Managed  
Download Datasheet »

### ClickToSecure Cloud



Cenzic ClickToSecure Cloud  
Download Datasheet »

## Professional Services

### Assessment Methodology



Cenzic Assessment Methodology  
Download Datasheet »

### Training



Cenzic Training Courses  
Download Datasheet »

## Hybrid

### Hybrid Model



Cenzic Hailstorm Enterprise ARC  
Download Datasheet »

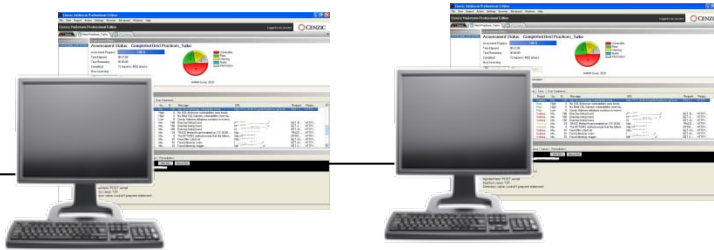


Cenzic ClickToSecure Managed  
Download Datasheet »

# Cenzic Hailstorm Pro & Cenzic Enterprise ARC Architecture



**ARC Desktop Client**



**ARC Web Users**



**Web Application**



**ARC Execution Engine**



**ARC Execution Engine**



**ARC Execution Engine**



**Centralized Database**



# Risk Management Dashboard

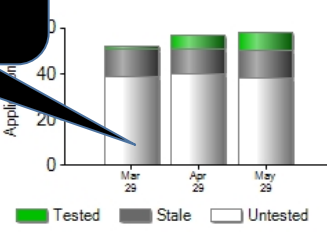


## Dashboard

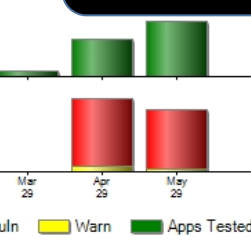
Summary of the last 3 months ending today

### Security Summary: Mar 01, 2009 through May 29, 2009

#### Coverage Trend

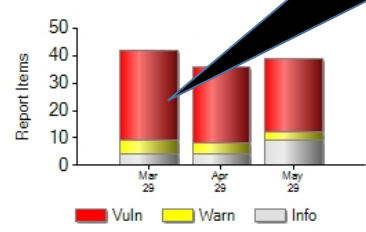


Web Interface



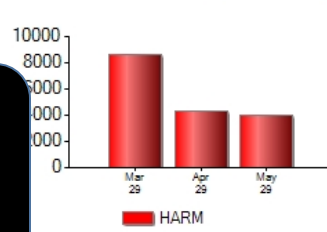
Tells vulnerability levels

#### Vulnerability Trend



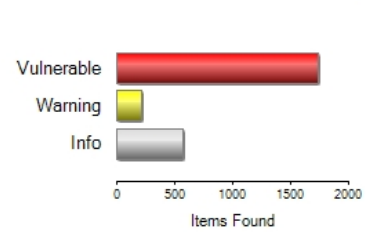
Tells which apps have been tested

#### HARM Trend (AVG)

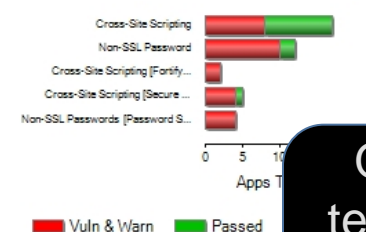


Finds and lists all applications

#### Overall Status



#### Top SmartAttacks



Quantitatively tells how severe the risk is for each app

Customize Charts

### Top 10 Applications by HARM: Mar 01, 2009 through May 29, 2009

Application	URL	Vuln	Warn	HARM Score	Status	Action
Crackme Bank	http://localhost:8081/	98 94 25	3 2 24	45083	not queued	report
Learning Portal	http://wordcircle.cenzicarc.com/	109 33	1 8	40062	queued	report
HacmeBank	http://172.16.17.7/HacmeBank_v2_Website/asp/Login.aspx?lmsg...	4 63 5	2 3 3	32164	not queued	report
WebGoat	http://172.16.18.18:8080/WebGoat/attack...	58 6 129	1 1 10	30012	not queued	report
Hacme Casino	http://172.16.18.18:3000/	1 59 10	--	29446	not queued	report
Sample Web Application	http://localhost:8081/kelev/view/cleardb.php	62 36 24	4 1 42	28176	not queued	report

## Accuracy With Broad Coverage

Most comprehensive attack library

## Enterprise Level Scalability

Ability to test all web applications

## Unparalleled Support

Cenzic responds to your needs within 24 hours

## Integrated Product

### Suite: Software + SaaS

Standardized platform for ultimate product flexibility

## End To End Security Offering

Cenzic partners with leading security vendors (WAF, remediation, etc.)



# Questions?

---

Lars Ewe

[www.cenzic.com](http://www.cenzic.com) | 1-866-4-CENZIC (1-866-423-6942)