

Are The Vendors Listening?

A Look Inside The Magic Quadrant

Abstract

In Feb 2002, 3APA3A released the first incarnation of “Bypassing Content Filtering Software”. Later on that year Andreas Marx and Mark Ackermans created a test set of some 370 emails with the purpose of testing the functionality of the industries email content filtering software.

After nearly 5 years, I thought it was time to take 3APA3A's updated paper, create a new set of malformed emails and test the attachment detection capabilities of the most popular SMTP filtering software on the market. My goal was to ascertain whether the vendors had been keeping an eye on the published works of the security community and proactively patching their products.

Introduction

Marx and Ackermans looked at 270 products across 43 different vendors. During the course of their research they notified the vendors and gave them a copy of their test cases. The majority were appreciative and patched their products to ensure the malformed emails could not be used by the underground for nefarious purposes. There were, however, a number of vendors who saw no immediate threat, and stated they would “re-evaluate the situation” if and when they saw these techniques used in the wild.

Are these vendors keeping up with the times?

The Magic Quadrant

I wanted my research to be relatively focused. I examined products with a presence in the “Gartner Magic Quadrant for E-Mail Security Boundary 2006”. To be selected for analysis and inclusion in a Gartner report, an organisation must have a signification market presence. The products selected fulfilled my requirement as they are widely deployed by some of the worlds largest organisations.

The following vendors were chosen from their respective categories:

Leaders: Symantec, Ironport,
Visionaries: Proofpoint, Clearswift, Marshall
Challengers: Sophos, TrendMicro, SurfControl

I found the following quote from the Gartner report indicative of what the engineers using these products would expect to get for their hard earned cash:

“The market is defined by vendors that provide enterprise protection against inbound e-mail threats, and fulfil outbound policy requirements at the SMTP gateway”

Method

Test File

A non-malicious VBS test file was created which displays a dialogue box (MsgBox) when executed. I chose a VBS attachment because it is hard to detect with true-type file detection signatures and is executable by default on Windows desktops.

True-type file detection does not need to rely on the content filter detecting malicious attachment extensions, it can merely look for a pattern such as MZ and strip the file encompassed within the MIME boundary. One could argue that content filtering software could look at the VBScript itself and block it based on key functions within the code. If you configure your content filtering software to check the message body for specific keywords, significant processor overhead is likely to be introduced. If one were to use this method, there are a number of obfuscation programs available on the Internet that can render the VBScript unreadable unless the content filtering software has decryption routines in place. Microsoft provides one such obfuscation tool called screnc:

```
screnc /l VBScript /e VBS test.vbs test.vbe
```

Test Cases

After determining which type of file to use, I examined the 20 attachment detection bypasses techniques detailed in 3APA3A's paper "Bypassing Content Filtering Software" and started writing my test cases. The test cases were constructed with the help of the Perl MIME::Lite library.

A control file was generated to ensure the attachment stripping was working as intended. It contained a VBScript file as an attachment without any evasion techniques.

```
Content-Disposition: attachment; filename="file.vbs"  
Content-Transfer-Encoding: base64  
Content-Type: application/octet-stream; name="file.vbs"
```

The test file was then modified a number of times for each of the following categories:

```
0x01_Encoded_Filename  
0x02_Multiple_Filenames  
0x03_Null_Byte  
0x04_Unsafe_fgets  
0x05_MIME_part_inside_MIME_part  
0x06_UUEncode  
0x07_Boundary_Additional_Space  
0x08_CR_without_LF  
0x09_Prohibited_Char_Filename  
0x0A_Skipped_File_Extension  
0x0B_Endless_UUEncode  
0x0C_Different_filename_Content-Type_and_Content_Disposition  
0x0D_Case_Sensitivity  
0x0E_Additional_dot_in_filename  
0x0F_RFC_2231_encoding_for_filenames  
0x10_Missed_MIME_Version_Header  
0x12_Empty_Boundary
```

In total, 104 unique samples were generated across the categories. The files were modified with the hope that if/when they reached the mail client at the other end the attachment would be recognised by the client and be executable.

Environment

My testing environment consisted of a Linux host running Postfix and Dovecot and a VMware Workstation installation running Windows 2003, Windows XP and RedHat EL3. The procedure for testing was as follows:

1. Install and configure each Email Boundary Security Product under VMWare to strip VBS attachments.
 - A number of the clients came configured out of the box to block malicious attachment types
 - Spam and virus filtering was disabled on all products as I was not testing this functionality
 - Configuration was done as per the documentation that came with the product

The following email boundary security products were tested:

- Proofpoint Messaging Security Gateway 4.0.7.67
 - Ironport Email Security 5.1.2-005
 - Mail Marshall SMTP 6.2.1.3252
 - TrendMicro IMSS 7.0 WIN32 5547
 - SurfControl E-Mail Security 6.0.0.39
 - Sophos PMX 5.3.3.310218
 - Symantec Mail Security 5.0.1
 - Clearswift MIMESweeper for SMTP 5.2
2. Configure each product to deliver filtered messages to Postfix
 - Postfix was configured as the next-hop relay

3. Send test cases to SMTP filtering product
 - The testing process was scripted so that all 104 test cases could be reliably repeated
4. Retrieve the filtered messages from Dovecot with the email clients
 - Each email client was configured to retrieve the messages from Dovecot via pop3

The following email clients were tested:

- Outlook Express 6.00.2900.2180
- Mozilla Thunderbird (win32) 2.0.0.6
- Outlook 2000 SR-1 9.0.0.3821
- Outlook 2003 11.6567.6567
- Lotus Notes 7.0.3.08152007

To get an initial baseline I sent all emails directly to postfix unmodified, and retrieved them with five different email clients to determine which attachments would render correctly.

Findings

Valid Tests

Of the 104 tests ran, only 58 emails were considered valid for the final test suite. I excluded 46 of the test messages which were either detected by all filtering engines and/or invalid in all of the mail clients tested.

The tests which successfully bypassed a content filtering engine on at least one occasion were:

Test Category	Number of Tests
0x01_Content_Disposition/Type	18
0x02_Multiple_Filename	4
0x03_Null_Byte_Filename	7
0x06_UUEncode	3
0x07_Boundary_Additional_Space	1
0x09_Prohibited_Char_Filename	5
0x0B_Endless_UUEncode	3
0x0C_Different_Filename	4
0x0D_Case_Sensitivity	1
0x0E_Additional_Dot_In_Filename	2
0x0F_RFC2231_Encoding_For_Filename	9
0x12_Empty_Boundary	1
Total	58

Table 1 – Successful Test Cases

Results

The results are collated below in a matrix detailing how each test fared against the email clients and boundary security products.

a) For each of the email clients (high-lighted in grey), a “1” indicates the .VBS attachment is able to be saved and executed. A “0” indicates the attachment is either not rendered or unable to be successfully saved and executed as a .VBS file.

b) For each of the boundary security products, a “1” indicates the attachment was stripped or the message dropped due to malformed content. A “0” indicates the message passed through the content filter unabated.

		Outlook Express 6	Mozilla Thunderbird	Outlook 2000	Outlook 2003	Lotus Notes 703	Proofpoint Messaging	Ironport C-Series	Mail Marshall SMTP	TrendMicro IMSS	SurfControl E-Mail	Sophos PMX	Symantec Mail Security	MIMEsweeper for SMTP	
0x01_Content_Disposition/Type															18
	3	1	0	1	1	0	0	1	1	0	0	0	1	1	
	4	1	0	1	1	1	0	0	1	1	0	0	0	1	
	5	1	0	1	1	1	0	1	1	1	1	1	0	1	
	6	1	1	1	1	0	1	1	1	1	1	1	1	1	
	8	1	1	1	1	1	1	1	1	1	1	0	1	1	
	9	1	0	1	1	1	0	1	1	1	1	1	0	1	
	10	1	1	1	1	1	1	0	1	1	1	1	1	1	
	11	1	1	1	1	0	1	0	1	1	1	1	1	1	
	12	1	1	1	1	1	1	0	1	1	1	1	1	1	
	13	1	1	1	1	1	1	0	1	1	1	1	1	1	
	16	1	1	1	1	1	1	1	1	1	1	0	1	1	
	17	1	0	1	1	1	0	1	1	1	1	1	0	1	
	22	1	1	1	1	1	1	1	1	1	1	0	1	1	
	23	0	1	1	0	0	0	0	0	0	1	1	0	0	
	24	1	1	1	1	1	1	0	1	1	1	1	1	1	
	25	0	1	1	0	1	1	0	1	1	0	0	0	1	
	26	1	1	1	1	1	1	0	1	1	0	0	0	0	
	27	0	1	0	0	0	1	0	1	1	0	0	0	1	
0x02_Multiple_Filename															4
	1	1	0	1	1	0	0	0	1	1	0	1	0	1	
	2	0	1	0	0	1	1	1	1	1	1	0	1	1	
	3	1	0	1	1	0	0	0	1	1	0	0	0	1	
	4	0	1	0	0	1	1	1	1	1	1	1	1	1	
0x03_Null_Byte_Filename															7
	2	1	1	1	1	0	1	1	0	1	1	1	0	0	
	3	0	1	0	0	0	0	1	1	1	0	1	0	1	
	4	1	0	1	1	1	1	0	0	0	1	0	0	0	
	5	0	0	0	0	1	0	0	0	0	1	0	0	0	
	6	1	1	1	1	0	1	1	1	1	0	1	0	1	
	7	0	1	0	0	0	1	1	1	1	0	1	0	1	
	8	1	1	1	1	0	1	1	1	1	0	1	1	1	
0x06_UUEncode															3
	1	1	1	1	1	1	0	0	1	1	1	0	1	1	
	2	1	1	1	1	1	0	0	1	1	1	0	1	1	
	3	1	1	1	1	1	0	0	1	1	1	0	1	1	
0x07_Boundary_Additional_Space															1
	3	1	1	1	1	1	1	1	1	1	1	0	1	1	

	Outlook Express 6	Mozilla Thunderbird	Outlook 2000	Outlook 2003	Lotus Notes 703	Proofpoint Messaging	Ironport C-Series	Mail Marshall SMTP	TrendMicro IMSS	SurfControl E-Mail	Sophos PMX	Symantec Mail Security	MIMEsweeper for SMTP	
0x09_Prohibited_Char_Filename														5
1	0	0	0	0	1	0	0	1	0	0	0	0	1	
2	0	0	1	0	0	0	0	1	0	0	0	0	1	
3	0	0	1	0	1	0	0	1	0	0	0	0	0	
4	0	0	1	0	0	0	0	0	0	0	0	0	0	
5	1	0	1	1	1	0	0	1	0	0	1	0	1	
0x0B_Endless_UUEncode														3
1	1	1	1	1	0	0	0	1	1	1	0	1	1	
2	1	1	1	1	0	0	0	1	1	1	0	1	1	
3	1	1	1	1	0	0	0	1	1	1	0	1	1	
0x0C_Different_Filename														4
1	1	0	1	1	0	0	0	1	1	0	1	0	1	
2	0	1	0	0	1	1	1	1	1	1	1	0	1	
3	1	0	1	1	0	0	0	1	1	0	1	0	1	
4	0	1	0	0	1	1	1	1	1	1	1	0	1	
0x0D_Case_Sensitivity														1
2	1	1	1	1	1	1	1	1	1	0	1	1	1	
0x0E_Additional_Dot_In_Filename														2
1	1	1	1	1	1	0	0	0	0	1	0	0	0	
2	0	1	0	0	1	1	0	0	1	1	1	0	0	
0x0F_RFC2231_Encoding_For_Filename														9
1	0	1	0	0	0	0	1	0	1	0	1	0	1	
2	0	1	0	0	0	0	1	0	1	0	1	0	1	
3	0	1	0	0	0	0	1	0	1	0	0	1	1	
4	0	1	0	0	0	0	1	0	1	0	0	1	1	
5	0	1	0	0	0	0	1	0	1	0	0	0	1	
6	0	1	0	0	0	0	1	0	1	0	0	0	1	
7	0	1	0	0	0	0	1	0	1	0	1	0	1	
8	0	1	0	0	0	0	1	0	1	0	1	0	1	
9	0	1	0	0	0	0	1	0	1	0	1	1	1	
0x12_Empty_Boundary														1
1	0	1	0	0	1	0	0	1	1	1	1	1	1	
Total	33	42	38	33	30	24	28	42	48	30	30	25	49	58
% Success Rate	57	72	66	57	52	41	48	72	83	52	52	43	84	

Table 2 – Results

High totals and success rates for email clients can be seen as either a positive or negative. The programmers who wrote these clients had probably dealt with a number of broken applications and wanted to make their email client as user friendly as possible... as opposed to RFC complaint. If everyone conformed to the RFCs the majority of these test cases wouldn't be viable.

The cases where the emails are actually valid and not detected by a number of the vendors are the RFC2231 and UUEncoded messages, this is a worrying trend.

For the SMTP boundary products, a high success rate is preferential. I am loathed to state who the “winners and losers” of this test were for fear of what their respective marketing department will do with the results, rather I have constructed a necromancers quadrangle to represent my findings.

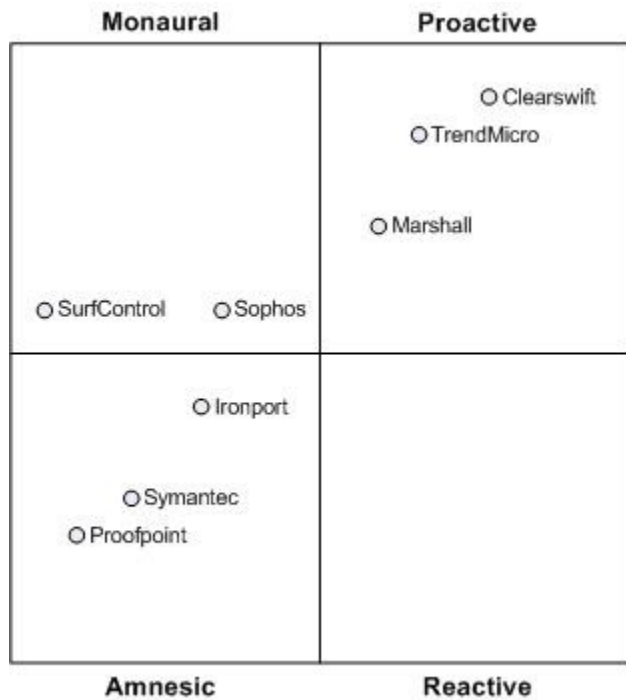


Figure 1 – Necromancers Quadrangle

Each of the four sections are segregated as follows:

Proactive:

The vendors in this quadrant are actively looking at techniques to bypass content filtering software and developing mitigation strategies for their products.

Monaural:

These vendors are listening somewhat, but haven't been keeping an open ear to developments in content filtering bypass techniques.

Amnesic:

Vendors they aren't actively looking at content filtering bypass techniques in the wild. Perhaps they forgot.

Reactive:

Vendors that only patch their products when a virus/worm/spam starts using a bypass technique to get around content filtering software. I foresee vendors in the amnesic section moving back and forth to the reactive section as threats are realised.

Recommendations

All vendors have been notified of these findings, it is beyond the scope of this document to detail their responses.

When considering purchasing a email filtering solution, a number of factors need to be taken into consideration,

often there is a trade off between usability, features and security.

There is talk in the industry of best-of-breed, perhaps people need to choose multiple boundary products to get the level of coverage they require.

Push your vendors to get these vulnerabilities patched.

Conclusion

I am sure that a number of the techniques presented in this paper can be worked around with undocumented or modified filters. Some of the vendors have replied with recommendations that will work, or partially work to mitigate the threats. In the end though, out of the box with documented configuration, over 50% of the vendors surveyed did not provide adequate protection. This leads me to believe, the vendors aren't listening.

References

3APA3A, 11 July 2007, Bypassing content filtering whitepaper, Securiyt.NNOV, Available at: <http://securityvulns.com/advisories/content.asp>

Andreas Marx, November 2002, Malformed Email Project – Part 1, Virus Bulletin, Available at: http://www.av-test.org/down/papers/2002-11_vb_malformed1.pdf

Andreas Marx, Mark Ackermans, February 2003, Malformed Email Project – Part 2, Virus Bulletin, Available at: http://www.av-test.org/down/papers/2003-02_vb_malformed2.pdf

File Format, 30 October 2007, Wikipedia, Available at: http://en.wikipedia.org/wiki/File_format#Magic_number

Ned Freed, Keith Moore, November 1997, MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations, RFC2231, Available at: <http://www.faqs.org/rfcs/rfc2231.html>

Peter Firstbrook, Arabella Hallawell, 25 September 2006, Magic Quadrant for E-Mail Security Boundary 2006, Gartner, Available at: <http://www.gartner.com/DisplayDocument?id=496544>

Ricardo Signes, 29 July 2007, MIME::Lite, CPAN, Available at: <http://search.cpan.org/dist/MIME-Lite/>

Script Encoder, 21 February 2003, Microsoft, Available at: <http://www.microsoft.com/downloads/details.aspx?FamilyId=E7877F67-C447-4873-B1B0-21F0626A6329>