

VIRUS ANALYSIS 1

SOBIG, SOBIGGER, SOBIGGER

Peter Ferrie

Symantec Security Response, USA

W32/Sobig is big, its code is bad, and its style is ugly. In the absence of correct information, there has been speculation and wrong information in abundance. Let us restrict ourselves to the facts.

INITIALISATION

All known variants of Sobig begin by initialising their random number generator, using the current time as the seed. The first line of code, and here's the first bug already: the initialisation is carried out only once, in the main thread, but the random number generator supports multi-threading (using Thread Local Storage – see *VB*, June 2002, p.4), and the worm uses multiple threads. Thus, the random number generator is not initialised in those threads (the seed always begins at zero), resulting in the same sequence whenever the worm is executed. The same bug exists in some other multi-threaded viruses, such as W32/Welchia (described on p.10), and appears to be a common programming error.

Sobig.A checks the current date at this point. The second line of code, and – yes – here's the second bug. The worm converts the date to 'yyyy.mm.d' format, and compares the date against '2003.1.23' (23 January 2003). The problem is that specifying 'mm' requests the month in a two-digit form, which *Windows* dutifully supplies, using a leading zero for the months prior to October (the 10th month). This results in a date format of, for example, '2003.01.23' so the comparison always fails. The format should have been 'yyyy.m.d'. Later variants of Sobig check the date in a different way, which works correctly.

Sobig then checks whether it has been run with any command-line parameters. If it has been run without parameters, it assumes that it was launched by the user (either as an email attachment, or as a file that was copied across the network, for the variants that exhibit this behaviour).

When run without command-line parameters, the worm will compare its path name against the common path name that it uses on a compromised machine. If the two differ, the worm will attempt to make a copy of itself using the common path name. What might be considered a bug exists here too – the comparison of the name is case-sensitive, but *Windows* does not alter the case of a filename when copying over an existing file, so if the case of the worm filename is ever altered, the worm will attempt to copy itself every time it is executed without command-line parameters.

All known variants of Sobig attempt to copy themselves to the *Windows* directory (as specified by the %windir% environment variable), however the filename has been changed with each version. The list follows:

Sobig.A: winmgm32.exe	Sobig.D: cftrb32.exe
Sobig.B: mscn32.exe	Sobig.E: winssk32.exe
Sobig.C: mscvb32.exe	Sobig.F: winppr32.exe

If the copy fails (for example, if the user created a directory using the worm filename, as an attempt at a counter-measure), the worm will use the name of the currently executing file instead.

REGISTER NOW!

If the path names match, or after the copy is attempted, and, in the case of Sobig.D, no other copies of the variant seem to be running (another bug, see below), the worm will add itself to the registry, by altering the Software\Microsoft\Windows\CurrentVersion\Run key in both the 'LocalMachine' and 'CurrentUser' hives. This ensures that the worm will be executed whenever *Windows* is rebooted. The value has been changed with each version. The list follows:

Sobig.A: WindowsMGM	Sobig.D: SFtrb Service
Sobig.B: System Tray	Sobig.E: SSK Service
Sobig.C: System MScvb	Sobig.F: TrayX

All known variants of Sobig prior to Sobig.F have no command-line parameter in their registry data, so the initialisation code is executed whenever *Windows* reboots, adding to the overhead of the system. Sobig.F adds a command-line parameter ('/sinc') to its registry data.

After the registry has been altered, the worm executes itself again, this time with a command-line parameter, in order to proceed with the main execution. The contents of the parameter are never checked, only its presence or absence. Each version of the worm passes a specific parameter to itself, and the parameter has been changed with each version except Sobig.D and Sobig.E. The list follows:

Sobig.A: start	Sobig.D: dwaqr
Sobig.B: xcvfd	Sobig.E: dwaqr
Sobig.C: dwaqr	Sobig.F: /sinc

THE BIG EVENT

All known variants of Sobig use a named event in an attempt to prevent multiple instances of a variant from running at the same time. The name of the event has been changed with each version:

Sobig.A: Worm.X

Sobig.D: Nibs.X

Sobig.B: Mnkx.X

Sobig.E: Nuiro.X

Sobig.C: Poss.X

Sobig.F: TrayX

Sobig.B: 31 May 2003

Sobig.E: 14 July 2003

Sobig.C: 8 June 2003

Sobig.F: 10 September 2003

Sobig.D: 2 July 2003

Unfortunately, the author(s) of the worm seem(s) to be incapable of mastering the required algorithm, despite several variations on the code in different variants. When a named event is created, the name is added to the global namespace of *Windows*, which means that the `CreateEvent()` API will not return a failure for an event that exists already (created by another process or even by a thread within the same process) – the same valid handle will be returned each time. The existence of the named event is that which should be checked. Instead, the worm checks if the event has been set (the worm sets the event whenever the initialisation code has completed). This results in a race condition that can, in turn, allow several copies of the worm to run at the same time – launched, for example, by someone clicking several times on the email attachment that ‘doesn’t seem to do anything’.

SOCKET 2.ME

After the worm’s initialisation has completed, the worm will initialise Winsock support, requesting version 2.2, but ignoring the version that is returned, perhaps assuming that any version will be sufficient (which begs the question: why request such an advanced version?). Several threads are created at this point.

The thread details changed in Sobig.B and again in Sobig.F. Sobig.A creates a thread to notify someone (perhaps the author of the worm) by *ICQ* pager whenever the worm runs. That thread sends a mail from ‘mail@mail.com’ to ‘0@icq.pager.com’, with a subject of ‘Notify’ and a body text of ‘Hello’. This code is not present in Sobig.B and later variants, although the thread is still created in the variants prior to Sobig.F. Interestingly, the *ICQ* user name and body text were removed completely in Sobig.B, restored in Sobig.C–E (though the body text was changed to ‘Worm started’, and the ‘0’ was changed to ‘1’ in Sobig.E), and removed again in Sobig.F.

The remaining threads that are created are for the checking of updates, spreading by email, and spreading by network shares. In Sobig.F, the number of email threads was increased from one to seven.

BEST IF USED BEFORE ...

Once the threads are created, Sobig.B and later variants will check the date on the local computer clock to see if their date of ‘expiry’ has been reached. The ‘expiry’ dates are as follows:

If the expiration date has not been reached, the worm will enumerate the drive letters from A: to Z:, with a three-second delay between each drive, searching for non-removable drives. For each non-removable drive that is found, the worm will search recursively through all subdirectories, looking for files whose suffix matches one of those on the list that the worm carries.

All known variants of Sobig carry a list that contains: txt, eml, html, htm, dbx and wab. Additionally, the list in Sobig.F contains mht and hlp. The contents of files whose suffix matches one of those on the list will then be searched for texts that resemble email addresses. The worm uses OLE Automation to drive the VBScript regular expression engine to find these texts. The expression used by the worm is

```
[A-Za-z0-9]+[A-Za-z0-9_.-]+@[([A-Za-z0-9\-\
])+[.])+[A-Za-z]+
```

which translates to: must contain at least one letter/number, followed by at least one letter/number/underscore/dot/hyphen, immediately preceding a ‘@’, followed by at least one letter/number/hyphen and a dot (and this combination can appear multiple times), followed by at least one letter.

Since the VBScript regular expression engine returns the number of unique strings found, multiple addresses can be extracted from a single file. The worm checks its list before adding each address, and will not add duplicates. This list is used by the email thread(s). Additionally, Sobig.F keeps a list of the first 1000 filenames whose suffix is one of: jpeg, jpg, gif, htm, txt, doc, xls, mpg, eml, bmp, fax. This list is used by the network enumeration thread in Sobig.F.

Another bug exists here: when the drive searching routine completes, the worm will exit, regardless of what the other threads are doing.

UP, UP AND UPDATE

The update thread attempts to download a file from a server on the list that the worm carries. The thread contains no date check, so it continues to function even after the expiration date if, and only if, the drive searching routine is running to keep the worm active (as described above). Otherwise, no part of Sobig will function at all after the expiration date.

Sobig.A and Sobig.B will attempt to contact servers whenever the update thread begins to execute. Sobig.C and later variants synchronise themselves first with network time protocol (NTP) servers, and will attempt to download

files only during certain hours of certain days. The list of NTP servers to be contacted is carried by the worm:

129.132.2.21 (swisstime.ee.ethz.ch)
 137.92.140.80 (chronos.ise.canberra.edu.au)
 200.19.119.69 (server2.pop-df.rnp.br)
 142.3.100.2 (clock.uregina.ca)
 128.233.3.101 (non-existent)
 193.5.216.14 (metasweb01.admin.ch)
 131.188.3.220 (ntp0-rz.rrze.uni-erlangen.de)
 131.188.3.222 (ntp2-rz.rrze.uni-erlangen.de)
 212.242.86.186 (ogps.freebsd.dk)
 chronos.cru.fr
 138.96.64.10 (ntp-sop.inria.fr)
 193.204.114.232 (unreachable)
 133.100.11.8 (clock.tl.fukuoka-u.ac.jp)
 193.67.79.202 (ntp0.nl.net)
 193.79.237.14 (ntp1.nl.net)
 132.181.12.13 (pukeko.cosc.canterbury.ac.nz)
 150.254.183.15 (vega.cbk.poznan.pl)
 62.119.40.98 (ntp1.sp.se)
 200.68.60.246 (non-existent)

The server to contact is chosen randomly, however Sobig.E contains a bug that restricts the choice to only the first four entries. The list of days and hours is as follows:

Sobig.C: Monday, Thursday, and Saturday, from 8pm to 11pm UTC
 Sobig.D: Thursday and Saturday, from 7pm to 12am UTC
 Sobig.E: Monday and Friday, from 7pm to 12am UTC
 Sobig.F: Sunday and Friday, from 7pm to 11pm UTC

Every two hours Sobig.A attempts to download the file from <http://www.geocities.com/reteras/reteral.txt>.

Sobig.B traverses its list by one row every two hours, and attempts to download from these sites:

<http://www.geocities.com/fjgoplsnjs/jane.txt>
<http://www.geocities.com/lfhcpsnfs/mdero.txt>
<http://www.geocities.com/dnggobhytc/nbvvhf.txt>
<http://www.geocities.com/bntdfkghvq/nbdef.txt>

Sobig.C traverses its list by one row every 59 minutes, and attempts to download from these sites:

<http://www.geocities.com/vbifhdgs/aadfa.txt>

<http://www.geocities.com/vbhhrtok/axcfa.txt>

<http://www.geocities.com/vbhcbhptok/axccfa.txt>

Sobig.D attempts approximately every 50 minutes to download from these sites:

63.187.136.207	65.92.185.105
218.145.251.172	4.46.216.107
63.139.177.178	68.158.97.35
68.119.94.107	65.96.174.173
211.172.37.81	80.133.8.182
203.218.1.205	65.96.134.32
80.193.162.47	68.160.246.76
217.86.31.254	68.51.149.158
24.159.40.38	24.101.46.49
24.199.119.153	67.85.144.168

Sobig.E attempts every hour to download from only the first five (because of a bug) of these sites:

67.164.250.26	80.145.119.84
129.244.36.194	61.41.223.43
67.73.60.121	218.158.43.206
218.146.139.246	67.168.13.135
66.169.84.77	209.34.8.147
64.229.253.52	65.69.221.166
65.95.29.173	67.74.161.243
203.252.75.45	80.136.150.140
217.230.224.66	69.22.34.186
65.95.91.31	62.47.6.238
217.228.235.145	24.96.26.108

Sobig.F attempts every hour to download from these sites:

68.50.208.96	65.92.80.218
12.232.104.221	63.250.82.87
218.147.164.29	65.92.186.145
24.33.66.38	65.95.193.138
12.158.102.205	65.93.81.59
24.197.143.132	65.177.240.194
24.206.75.137	66.131.207.81
24.202.91.43	67.9.241.67
24.210.182.156	68.38.159.161
61.38.187.59	67.73.21.6

Sobig.D and Sobig.E also listen on ports 995–999 for incoming data that can be used in addition to (or instead of)

the download server list. This is in the same format as the list which is downloaded from the servers.

Sobig.A–C will connect to the servers on port 80, using a standard *Windows* API to download the data; Sobig.D–F will connect to the servers on port 8998 to download the data. In all cases, the data expected to be received is the URL of a file to download and execute. Sobig.A uses no encryption at all; Sobig.B uses simple bit-twiddling of nybbles. Sobig.C–F use, according to Frédéric Perriot, a variant of DES which has the first and last steps removed. The lack of these steps does not seem to reduce the strength of the encryption. Additionally, the key generation algorithm was changed in Sobig.F, resulting in an incompatibility with previous variants.

Once the data has been downloaded, the worm will create or open a local download log file in the %windir% directory, then search the log file for a match of the data. Sobig.A attempts to match the entire URL, Sobig.B–F attempt to match only the filename part of the URL. If the data is not found in the log, it will be added to the log, after which the requested file will be downloaded and executed. The name of the download log file has been changed with each version. The list follows:

Sobig.A: dwn.dat	Sobig.D: dftrn32.dat
Sobig.B: msdbrr.ini	Sobig.E: msrrd32.dat
Sobig.C: msddr.dll	Sobig.F: winstf32.dll

EVER HEARD OF THE MAILMAN?

The email thread begins by creating or opening a local file in the %windir% directory that contains the ‘sent’ list. The name of the sent list file has been changed with each version. The list follows:

Sobig.A: smtmls.ini	Sobig.D: rssp32.dat
Sobig.B: hnks.ini	Sobig.E: msrrf.dat
Sobig.C: msddr.dat	Sobig.F: winstt32.dat

The worm will send one mail to each person who is not on the sent list. In the case of Sobig.F, however, the email threads are not synchronised, so it is highly likely that more than one thread (and potentially all seven of them) will send mail to each person not on the sent list. Additionally, if copies of the worm are running, all of those copies could potentially be sending mail to the same people.

For the sender’s address, the worm will use its default address if only one email address is found or is remaining on the list; otherwise the choice will be made randomly from the list of email addresses, excluding the recipient’s address. No attempt is made to discover the email address of the real sender, so if that address exists on the list, it could

be used. The default address has been changed with each version:

Sobig.A: big@boss.com
 Sobig.B: support@microsoft.com
 Sobig.C: bill@microsoft.com
 Sobig.D: admin@support.com
 Sobig.E: support@yahoo.com
 Sobig.F: admin@internet.com

The subject of the mail is chosen at random from a list that the worm carries. In the case of Sobig.E, a bug restricts the choice to only the first two entries. The list has been changed with each version. The list follows:

Sobig.A:
 Re: Here is that sample Re: Sample
 Re: Document Re: Movies

Sobig.B:
 Re: My application Approved (Ref: 38446-263)
 Re: Movie Re: My details
 Cool screensaver Your password
 Screensaver Your details
 Re: Approved (Ref: 3394-65467)

Sobig.C:
 Re: Application Re: Screensaver
 Re: Your application Re: Movie
 Re: 45443-343556 Approved
 Re: Submitted (004756-3463) Re: Approved

Sobig.D:
 Re: Application Application Ref: 456003
 Your Application Re: Movies
 Re: Accepted Re: App. 00347545-002
 Re: Screensaver Re: Documents
 Re: Your Application (Ref: 003844)

Sobig.E:
 Re: Application Application.pif
 Re: Movie Applications.pif
 Re: Movies movie.pif
 Re: Submitted Screensaver.scr
 Re: Screensaver submitted.pif
 Re: Documents new document.pif
 Re: Re: Application ref 003644 Re: document.pif

Re: Re: Document	004448554.pif	Screensaver.scr	app003475.pif
Your application	Referer.pif	Application844.pif	
Sobig.F:		Sobig.E:	
Re: That movie	Re: Details	Your_details.zip (contains Details.pif)	
Re: Wicked screensaver	Your details	Application.zip (contains Application.pif)	
Re: Your application	Thank you!	Document.zip (contains Document.pif)	
Re: Approved	Re: Thank you!	Screensaver.zip (contains Sky.world.scr)	
Re: Re: My details		Movie.zip (contains Movie.pif)	

The message body has been changed with each version. The list follows:

Sobig.A: Attached file: [attachment name]
 Sobig.B: All information is in the attached file.
 Sobig.C: Please see the attached file.
 Sobig.D: See the attached file for details
 Sobig.E: Please see the attached zip file for details.
 Sobig.F chooses randomly from:
 Please see the attached file for details.
 See the attached file for details

The attachment name is chosen at random from a list carried by the worm. In the case of Sobig.E, the list is useless, since a bug restricts the choice to the first entry. The list has been changed with each version:

Sobig.A:
 Sample.pif Document003.pif
 Untitled1.pif Movie_0074.mpeg.pif

Sobig.B:

application.pif	password.pif
movie28.pif	approved.pif
screen_doc.pif	ref-394755.pif
screen_temp.pif	your_details.pif
doc_details.pif	

Sobig.C:

document.pif	45443.pif
application.pif	submitted.pif [sic]
approved.pif	movie.pif
documents.pif	screensaver.scr

Sobig.D:

Application.pif	ref_456.pif
Applications.pif	movies.pif
Accepted.pif	Document.pif

Sobig.F:

movie0045.pif	your_details.pif
wicked_scr.scr	thank_you.pif
application.pif	document_all.pif
document_9446.pif	your_document.pif
details.pif	

Sobig.E is the only known variant of Sobig that sends its attachments in Zip form. It carries the Zlib Deflate library in order to compress itself (as opposed to only storing, which requires no library code). Zip is a suffix that is not blocked by the *Outlook* security patch that prevents access to attachments based on their suffix.

The email part-separator is formed by appending a random eight-character hexadecimal value to a hard-coded text. For variants A to E the text is 'CSmtpMsgPart123X456_000_'. For Sobig.F the text is '_NextPart_000_'. The size of the attachment is variable in Sobig.E and Sobig.F. The file possesses a tail that contains the text 'XC001815d', prepended by a 32-bit value containing the number of bytes appended to the original file.

Before sending the attachment, the worm will add a random number of up to 3995 bytes to the original file, then include this tail. If the new size is larger than the old size then the tail text can be visible multiple times. Emails that are sent by Sobig.F include several 'X-' headers, one of which is 'X-MailScanner: Found to be clean'. Let us hope that no one is fooled by this.

When opening the file to attach, all known variants of Sobig do so in a mode without sharing enabled. This can result in a failure to open the file. The message is sent anyway, which is a reason why some emails that are sent by Sobig will arrive without the attachment.

SCHMOOZING AND NETWORKING

The thread for spreading by network shares is fairly standard. The routine is executed every four hours by Sobig.A, and every half hour by Sobig.B and later variants.

The worm enumerates the machines on the local network, and attempts to connect to the C\$, D\$, and E\$, shares on each machine. For the variants of Sobig prior to Sobig.F, if the connection succeeds, the worm will copy itself to the root directory of the share, and to the Startup directories specific to US-English *Windows 2000/XP/2003* (Documents and Settings\All Users\Start Menu\Programs\Startup) and *Windows 9x/Me* (Windows\All Users\Start Menu\Programs\Startup). Sobig.F begins by examining its list of the first 1000 filenames gathered by the drive-searching routine. If the list exists, a filename is chosen randomly from there, and '.exe' is appended to form a double extension (e.g. file.jpg.exe). If no filename exists, 'winpr32.exe' will be used instead. After the filename is chosen ... nothing happens – the code to perform the file copy is not present in Sobig.F.

CONCLUSION

The storm of mail that was produced by Sobig.F might make it seem as though tens of millions of people were infected by the worm. This should give pause for thought, since multiple copies of the worm running at the same time, coupled with the multiple email threads, on a far smaller number of machines, could easily account for much of the contribution. Despite this, some variants of Sobig have spread quite well and very quickly. Sobig relies on minimal social engineering, and no exploits. Sobig.F does not even spread across the network, relying solely on email for its propagation. How has it become so successful? A recent comic explains (from *User Friendly* by J.D. 'Illiad' Frazer, (c) User Friendly Media Inc, userfriendly.org.):

Q. After working for an ISP for over half a decade and being immersed in the web, what is the one lesson you've learned?!

A: Click on everything.

Okay. Stop clicking. NOW.

W32/Sobig@mm

Type:	Win32 SMTP mass-mailer worm, network share crawler.
Size:	65,538 bytes (A), 50,425 bytes (B), 57,241 bytes (C), 76,003 bytes (D), 86,528 bytes (E), 72,892 bytes (F).
Payload:	Can download and execute arbitrary executables without permission.
Removal:	Fix registry, delete worm copies and its data files.