

The Making of a Spam Zombie Army

Dissecting the Sobig Worms

On 25 June 2003, a new mass-mailing worm emerged. We'd hardly think it worth noting except for a few important details: The worm was highly successful at infecting computer systems, is the fifth revision of a worm that first raised its ugly head in January

started making use of a built-in list of network time protocol (NTP) servers to obtain the current date and time, upon which it makes time-based decisions. This is likely an attempt to enforce the time restrictions on infected systems that have the incorrect time.

Multistage malicious code

Sobig also includes an upgrade mechanism. Sobig variants A–C downloaded files hosted at geocities.com from static embedded URLs. The downloaded files contained other URLs that pointed to files that Sobig downloaded and executed. By controlling the Geocities-hosted file contents, the worm author could direct the worm to download and execute files located anywhere on the Internet.

This mechanism's weakness is that the pointing to Geocities are

ELIAS LEVY
Symantec

of this year, and there is strong evidence that spammers use infected computers to relay unsolicited commercial email.

Like its predecessors, Sobig.E is unremarkable in many ways. It's a piece of malicious code that targets Microsoft Windows operating systems. Written in Microsoft Visual C++, it makes use of threads, its executable is compressed with either UPX or TeLock, it collects email addresses by harvesting files (such as Windows Address Book [WAB], Outlook Express mailbox [DBX], HTM, HTML, Mail message [EML], or text [TXT]), and attempts to infect new systems by sending them an infected email message or by copying itself to an open network share. The worm also includes its own simple mail transport protocol (SMTP) engine, spoofs its emails' source address, encrypts and decrypts text strings as needed, and creates a mutual exclusion object (mutex) on infected systems to ensure they are not infected more than once.

A more curious characteristic is that the worm is programmed to stop infecting new systems after 14 July 2003. Note that the worm

does not terminate itself; other functionality, explained next, continues to execute. In fact, all previous variants of Sobig, except the first (Sobig.A), have a built-in date after which they stop propagating (see Table 1). From the continued release of new versions of Sobig with built-in propagation-termination dates, we can surmise that a single individual or organization with access to the worm's source code is responsible for all the variants. With Sobig.D, the worm also

Calling all malware reporters

Welcome to the premier of Malware Recon. We envision using this intermittently appearing department to address virus, worm, and other malware issues on an ad hoc, issue-by-issue basis. The emergence and impact of the Sobig.x worm dissected here is a good example of the topics we could be covering. Just as we were going to press, author Elias Levy was able to provide a good run-down on Sobig and its variants. The idea is to get an authoritative overview of a

threat's strengths and weaknesses, dig a little deeper into how it works, and discuss how it fits or doesn't fit into ongoing trends. Other topics might include first-hand practices and patches, attack-encounter reports, and real-world strategies against malware and their spawn.

If you are interested in making a contribution, contact Attack Trends department editors Iván Arce (ivan.arce@corest.com) or Elias Levy (aleph1@securityfocus.com) to discuss potential topics.

hard coded. Once they identified the URLs, security researchers could access the sites to determine what programs were being downloaded and executed, analyze their functionality, and devise detection mechanisms. The worm's author attempted to hide this weakness by placing a decoy URL in the Geocities-hosted file for most of the time. When a security researcher downloaded the file to examine the URL, in most instances, it would be invalid. The worm's author only updates the valid URL file for short periods of time, but the worm continuously attempts to download the URL, so it has a high probability of hitting the valid URL and downloading and executing the file it pointed to.

The valid URL pointed to a new malicious code, named Backdoor.Lala (<http://securityresponse.symantec.com/avcenter/venc/data/backdoor.lala.html>). This backdoor installed a keystroke logger and permitted the attacker to steal sensitive information, such as stored passwords and cookies, and to download and execute files.

According to Joe Stewart of LURHQ Corporation (www.lurhq.com/sobig.html), the worm's author then seems to have used the backdoor to disinfect Sobig's system and install the Wingate (www.wingate.com) proxy program on infected computers. The Wingate proxy server lets spammers launder their connections through the infected hosts, hiding the spams' real source.

Independence from Geocities

Another problem with the hard-coded URL upgrade mechanism is that Geocities became very efficient at shutting down the offending URLs. Starting with Sobig.D, however, the upgrade mechanism changed (www.lurhq.com/sobig-e.html).

Sobig.D and Sobig.E included a list of hosts that they attempted to contact by sending UDP packets to port 8998 between 7:00pm and

Table 1. Sobig variation self-propagation stop dates.

VARIANT	DISCOVERY DATE	PROPAGATION STOP DATE
Sobig.A	9 January 2003	None
Sobig.B	18 May 2003	31 May 2003
Sobig.C	31 May 2003	8 June 2003
Sobig.D	18 June 2003	2 July 2003
Sobig.E	25 June 2003	14 July 2003

11:59pm UTC. As with the Geocities update mechanism, the response was often invalid, but these systems occasionally responded with an encrypted URL that Sobig could retrieve and execute.

Once again, the next malicious download was a variant of Backdoor.Lala, which also disinfected the system of Sobig and installed the Wingate proxy, although this time the ports changed, probably because the earlier port number had been compromised.

In addition, the new variants opened and listened on UDP ports 995–999. By sending a UDP packet to these ports, the author could update the list of servers to contact for the URL to download. Thus, even if the hosts in the hard-coded host list still listening to UDP port 8998 became disinfected or were shut down, the author could attempt to recover them using Wingate on the infected hosts to scan for hosts with UDP ports 995–999 open and sending them a new host list.

The quick spread of some variants and the receipt of infected messages in long-ago decommissioned email addresses has led to the speculation that spamming seeded at least some variants of Sobig—ironic because a worm used to improve the author's ability to spam was itself spread through spam.

Sobig is not the only malicious code spammers use to infect computers and launder their connections. For more on this breed, see Kevin Poulsen's "Rise of Spam

Zombies" (www.securityfocus.com/news/4217). Other spam codes include Backdoor.Guzu (<http://securityresponse.symantec.com/avcenter/venc/data/backdoor.guzu.html>) and Backdoor.Jeen (<http://securityresponse.symantec.com/avcenter/venc/data/backdoor.jeen.html>). Backdoor.Jeen is reportedly very successful.

These attack approaches outline a disturbing trend. The growing anger and ongoing war against spam—and the temptation of easy money—has led some spammers to start using computer-hacker and malware author techniques. Spammers are moving from the legal gray area of using open relays and proxies to send their messages to the decidedly illegal commandeering of unknowing users' computers to continue with their activities. We can expect the problem to become worse as efforts increase to stamp out spam.

Still, new security measures are not needed. Existing ones, such as personal firewalls and antivirus measures, are sufficient. More importantly, good security practices—such as not executing attachments and keeping up with security patches—are paramount. □

Elias Levy is an architect with Symantec. Previously, he was the chief technology officer and cofounder of SecurityFocus and the moderator of Bugtraq, a vulnerability disclosure mailing list. His research interests include buffer overflows and networking protocol vulnerabilities. He is also a frequent commentator on computer security issues and participates as a technical advisor to a number of security related companies. Contact him at aleph1@securityfocus.com.