



Securing Virus Code
infectionvectors.com
July 2004

Overview

The number of compression and encryption tools available to hide the inner workings of a program rises everyday. These tools are valuable to companies trying to keep their code safe from reverse engineering and would-be pirates. Virus writers use numerous tools to keep their code safe as well, from unique encryption routines to commercially available packers. This brief review of various methods will help administrators better understand how viruses are protected and what types of utilities (found during workstation inspections or during post-compromise forensics) may indicate nefarious activity on a computer.

Compression and Encryption Tools

Much more dominant on the closed-source front (for obvious profit-protection purposes), application packers help keep executables' code proprietary. Examples include the very popular UPX (or, Ultimate Packer for eXecutables), which is distributed under the GPL. UPX compresses application code and then decompresses it as the program is run. Information on UPX can be found at its website: <http://upx.sourceforge.net/>.

UPXScrambler, a tool that adds an encryption routine to the code, is often associated with viral applications. It is detected with ease by many virus scanners and flagged as a "hack tool."

Pest Patrol UPXScrambler Information

http://www.pestpatrol.com/pestinfo/v/virtool_win32_upxscrambler.asp

Additional cryptors and packers of note:

FSG - (Fast Small Good) has been a popular choice due to a low level of recognition of its packing method by antivirus software.

PEX - An executable protector that also appears as the packer of choice for many viruses, including later versions of Beagle.

ASPack - A commercial product that compresses and encodes Win32 applications, available at www.aspack.com.

The list of packing and enciphering tools is long and grows every week. The reason so many are developed and used is simple: they hinder detection and analysis. When an

application is packed with a new compression method it may be invisible to antivirus programs designed to catch the code. Once it is running, the code must be decompressed (which may expose the virus to the security software) before it is run. If there is any encryption used, this must also be undone so that the computer can understand the instructions passed to it. Once a virus is running on a host however, it may employ a routine to disable the antivirus protection before the malicious software is caught and cleaned. Either way, the code is harder to disassemble once it is identified as a virus. The packing and ciphering must be undone, a process that can be sped up by simulating the execution of the program and dumping the unencrypted assembly instructions to a file that the analyst can examine.

One final note in this overview on packing tool features and use: Once compressed, any application is made more efficient for network delivery, a key element in efficiently distributing a worm across the Internet.

Application of Compression Tools in the Wild

At some point, virtually every packer and encryption tool has been employed by a virus coder to package his or her wares. A few recent examples:

Beagle UPX/PEX

MiMail LCC/Petite

Sasser PECompact

WallonUPX and ASPack 2.12 (1 of the EXEs is UPX, the other 2 are ASPack)

DabberUPXScrambler

SDBot FSG 1.33 (among numerous others)

Along with the compression tool, a number of decompressors exist. These are generally written and used by reverse engineers, both legitimate researchers and criminal elements.

In addition to these tools, some virus code is written to employ other secure methods, hiding communication and intentions from users. Wallon used a trick that many spammers employ: encrypted URLs to hide destinations from casual observers. Phatbot built upon the WASTE code, which had the capability to cipher all P2P communications (note, however, that the Phatbot in circulation today does not tap the encryption capabilities of that code).

Secure coding is vastly different from securing code, however. Prior to obfuscating what the program does, many virus writers are employing techniques of testing and development that allow them to produce software in much the same way commercial outfits do. For a look at these practices in action, see the report “The Beagle Worm History Through April 24, 2004.”

http://downloads.securityfocus.com/library/Beagle_Lessons.pdf

Weaknesses in viral code have been exploited many times, as in the case of Sasser's FTP server buffer overflow which allowed Dabber to infect and control a Sasser-compromised machine. Often a virus needs to get into circulation quickly, the faster the better when it comes to exploiting newly discovered vulnerabilities. In these cases, a solid means of obfuscating the code, so it cannot be analyzed expediently, is the only course for a worm writer.

Copyright © 2004 infectionvectors.com. All rights reserved.
For reprint rights contact@infectionvectors.com