

Infectable Objects Part Four- Viruses in Archive Files and Compressed Files

Robert Vibert 2001-01-10

Infectable Objects Part Four - Viruses in Archive Files and Compressed Files

by Robert Vibert

last updated Jan. 10, 2001

No matter how quickly the speed of the Internet increases, we still find it convenient to compress files before we send them. Once a file is compressed, however, it becomes harder for a virus scanner to find any virus that may be lurking inside it. This challenge - peering inside the various compression and archival formats to discover the viruses hidden there - has not gotten easier over time. This article will discuss the implications of utilizing various forms of file compression on virus protection.

To properly understand this aspect of files that can contain viruses, we need to distinguish between archive files and compressed files, particularly compressed executable files.

Archive Files

Let's start with archive files, those most commonly used to cut down on transmission times or to save storage space. Almost everyone who uses a computer on the Internet has run into a file with the extension .ZIP, although it might not be immediately obvious what it is the first time you see it. A quick look around on the Internet for Windows 9x/NT archive file programs which deal with the .ZIP format reveals more than twenty, with names like WinZIP, TurboZIP, NetZIP, and FileWrangler. No doubt, one could find more given a little more time.

So what exactly is a .ZIP file? We can think of it, and other archive files, as a container into which we squish one or more files, reducing their size as they are added to the file.

Compression software uses complex mathematical equations to scan a file for repeating patterns in the data. It replaces this repeated data with smaller codes that occupy less space. For example, one way that compression software works is to replace repeating text characters with a code that also notes the locations of those characters in the data. With a picture, it would find all the green parts, for example, and replace them with a code.

There are many different methods of compression: over the years, PKZIP alone has used seven

different methods for the .ZIP file format. What does this mean for a virus scanner? It means that even though the file is in .ZIP format, it might contain files stored using any one of various compression methods. It also might contain files that have been previously compressed, using .ZIP or another compression format.

Over the years, a wide variety of compression formats have been developed. It is beyond the scope of this article to try to cover them all, but a quick sampling will give you some idea of what a virus scanner might face:

- .ARC - one of the older formats for archiving;
- .ARJ - although not seen much outside Europe, still a fairly common format for MS-DOS machines;
- .GZ/GZIP - the GNU Project's compression program, most commonly used for UNIX and PC files;
- .SIT - a Macintosh file that has been compressed using a program called Stuffit;
- .SEA - a Macintosh self-extracting archive file.
- .TAR/.TAR.GZ/.TAR.Z/.TGZ - A UNIX archiving scheme that is also available for PCs. Tar, which is short for Tape ARchive, can archive files but not compress them. .TAR files are often gzipped, which is why one might occasionally encounter the file extension .tar.gz.
- .Z - a UNIX compression format

Of course, this list is far from complete. There are a number of other archive formats to consider, including ACE, CAB, LHA, RAR for the PC, as well as Unix formats such as B2 and TBZ.

Because any archive file can contain one or more already-archived files inside it, a virus scanner must be able to not only decipher the contents of an archival file, but also to subsequently scan inside each of the compressed files inside the main compressed file. It could have to look inside a .ZIP file, inside an .ACE file, inside an .ARJ, file inside a .CAB file inside a LHA file, inside another .ZIP file and so on, and so on. This is called recursive scanning of multiple packed archives - although, some Anti-Virus developers simply call it a pain in the neck. However, if a virus is inside one of these files, you will want to know about it.

Archive files raise several other concerns. For instance, they require room in the system's memory in order to be decompressed, thereby bringing about the issue of where they will be decompressed - in memory, on disk, or both. In the early days, it was common for an Anti-Virus scanner to spawn a copy of a decompression utility itself to process the files and then

hand them over to the scanner for inspection. This implied that enough hard disk space was available for the decompression, which could be a major problem if the size of the decompressed files exceeded that of the free space on the hard disk.

This problem was quickly overcome by Anti-Virus vendors, who developed software that would decompress files in memory or use algorithms that can decipher the compression without needing to actually expand the file itself. Some can even go as deep as fifty layers inside an archive file.

Several issues remain to be resolved, such as password-protected archives. Most Anti-Virus programs cannot inspect an archive file that has been password protected by the compression program.

A further issue arises with self-extracting files. A number of file compression formats, including .ACE, .ARJ, .LHA, .RAR16, .RAR32, and .ZIP can be created to auto-extract themselves. Self-extracting archives usually work by including a decompression program with the archive, and this program itself can be infected.

Some programs, such as WinZIP, have an option that permits them to create archives that automatically run program files included in the .ZIP when it is decompressed. The self-extracting .ZIP file will usually create a temporary folder, unzip the files into the specified folder, run the program, wait for it to complete, then delete the temporary folder created in the first step. However, if this program was infected, it could then infect your system.

The repair of infected archives - the removal of infected files or removal of the virus itself inside an archive, is another option offered by some Anti-Virus programs. It can be a fairly complicated process, especially if the infection lies several layers deep in the archive.

Many organizations scan all archival formats at their firewall or gateway.

Executable File Compressors

The other family of file compression one needs to consider is that of executable file compressors. When programmers create programs, they are typically compiled into files with .EXE extensions. As we saw in [part one](#) and [part two](#) in this series, there are a number of types of executable files for DOS and Windows, with different formats. Virus scanners have to understand these different executable file formats.

Programmers have long used any available techniques and tools to reduce the programs size. Today, there are dozens of file compressors available. For instance, programs like Diet, Ice, LZExe, PkLite, and WWPack are used for DOS executable compression. While for Windows (32-Bit) executable compression, one can use compressors such as ASPack, CEexe, Neolite, PEPack , Petite, PKLite32, Shrink, UPX, and WWPack32.

All of these compressor utilities work in essentially the same manner. They compress the data that makes up a program, and attach a small loader / decompressor module to the compressed executable file. Each time the program is run, the loader module activates and decompresses the compressed program data into the computer's memory. Once the program is in memory, as it would be after a normal loading in its uncompressed state, the loader module passes control to the actual program to run.

Program file compressors offer a range of options and capabilities, including

- reduction of the file size of 32-bit Windows programs and libraries by as much as 70%;
- some protection of programs against reverse engineering (debuggers and dis-assemblers) via encryption;
- creation of program versions with registration keys, evaluation and trial versions;
- creation of a built-in application integrity check;
- compression of DLLs - Dynamic Link Library;
- OCX - OLE Custom Control;
- ActiveX control files, data, resources - icons, dialog boxes and other bitmaps, and DOS device driver program files (.DRV); and,
- adding virus detection to the compressed executables - they will check themselves for infection every time they are executed.

The problem for virus scanners is that once an infected program has been compressed, it usually requires a new definition to be detected, as the compressor will alter the look of the file. PrettyPark and MiniZip are examples of virus/worm programs that have been altered in this way during the past couple of years. Decompressing virus- infected programs and then re-compressing them with different compression tools is easy and can be used by virus authors to create new variants.

MiniZip, also called Worm.ExploreZip(pack), was discovered in November 1999. This worm is a variant of Worm.ExploreZip, which was discovered in June 1999. The only difference between

them is that Worm.ExploreZip(pack) has been packed by file compression to reduce the file size to be about 40% smaller than the original Worm.ExploreZip. The compression resulted in this new version being missed by many scanners.

As well, the common SubSeven Trojan backdoor tool is distributed, together with compression instructions, in an "unpacked form" specifically with the intent that hackers can compress it themselves to make detection harder.

The Russian Dolls Problem - Embedded Objects

In order to wrap up this discussion of the challenges of finding viruses in files that have been hidden inside archive and compressed executable files, we must touch briefly upon the ways in which viruses can be hidden inside other files.

It is quite common for someone using MS Office to want to insert an Excel spreadsheet or graph in a Word document or a PowerPoint presentation. The problem is that doing this with an infected Excel spreadsheet can result in the virus being embedded in the other file. Users can create a document in one application (such as MS Word) and then move, copy, or link content from other documents. Data objects that retain their native full-featured editing and operating capabilities when they are moved into another container (document) are called embedded objects. Generally, containers support nesting of embedded and linked objects to any level. For example, a user can embed a chart in a worksheet, which, in turn, can be embedded in a word-processing document. The model for interaction is consistent at each level of nesting.

The following table is just a small example of the ways in which objects can be embedded into files. Anti-Virus developers have identified close to four hundred different combinations of files and embedded objects, and identification of viruses in these is not always the biggest headache. Because of the way some embedded objects are stored in some files, they may be next to impossible to clean - PowerPoint PPT files with embedded Excel charts are a good example.

Files That Could Contain Embedded Objects	Embedded Objects					
	DOC	PPT	XLS	COM	EXE	VBS
Office 97 - DOC	X	X	X	X	X	X
Office 97 - PPT	X	X	X	X	X	X
Office 97 - XLS	X	X	X	X	X	X

Office 2000 - DOC	X	X	X	X	X	X
Office 2000 - PPT	X	X	X	X	X	X
Office 2000 - XLS	X	X	X	X	X	X
RTF	X	X	X	X	X	X
SHS	X	X	X	X	X	X
DOC-MSO	X	X	X	X	X	X
PPT-MSO	X	X	X	X	X	X
XLS-MSO	X	X	X	X	X	X

The following table, extracted from [The Enterprise Anti-Virus Book](#), details some of the more common archive file formats.

Infectable Object	Description
.ACE	ACE Archiver file
.AIN	AIN-compressed file
.ARC	PKARC Archiver file
.ARJ, .A0?, .A1?, ...	ARJ Archiver file
.b64	Encoded base64 MIME archive
.BO?	Boot sector image
.BZ	BZIP compressed file
.BZ2	BZIP2 compressed file
.CAB	MS Cabinet file
.COM	Self-extracting archives
.COM	DIET, PKLITE, CRYPTCOM, ICE, etc.
.CPIO	Unix Archive file
.CPT	Compressed MAC file
.EXE	DIET, PKLITE, LZEXE, UPX, etc.
.GZ	GZIP compressed
.ICE	ICE compressed
.IM?	Disk image
.JAR	Java Archive
.LIM	Limit compressed

.LZH	LHA compressed
.MSI	MS Windows Installer file
.PAK	PkPak compressed
.RAR, .R0?, .R1?, ...	RAR compressed
.TAR	Unix TAR compressed
.TAZ	Unix Compressed TAR file
.TDO	TeleDisk diskette images
.UU	UUEncoded Unix file
.UUE	UUEncode
.Z	Unix Compress file
.ZIP	PKZIP, WinZip
.ZOO	ZOO
.??_	MS Compress/Expand

To read **Infectable Objects, Part 5**, click [here](#).

Relevant Links

[Infectable Objects Part 1 - DOS](#)

Robert Vibert

[Infectable Objects Part 2 - Windows Infectable](#)

Robert Vibert

[Infectable Objects Part 3 - Win Apps](#)

Robert Vibert

[MiniZip worm highlights weak antivirus defenses](#)

Ann Harris, Computerworld

[AV-Test.org](#)

[Privacy Statement](#)

Copyright 2006, SecurityFocus