Andres Monroy

U0733037

Gzip is an application designed for file compression and decompression. It comes packaged into the standard linux distributions, and is available for several other platforms. The format of gzip is described in the RFC 1951 and 1952. The gzip deflate() algorithm (compression) is based, in part, by the LZ77 (Abraham Lempel and Jacob Ziv in 1977) algorithm, and the inflate() algorithm is based on the Huffman tree. The algorithm of gzip is best used on files with relatively uniform data. The worst performance is with files of completely random data. The version of gzip used for testing is gzip-1.3.12.

Three methods were used to test the solidity and reliability of gzip. A manual code review, valgrind for memory management, and fuzz testing.

Beginning with the code review, gzip is a tool which has been in development for approximately 22 years (released Oct. 31, 1992), and should be expected to have been heavily reviewed and tested. The code shows many aspects of solid code design. Any outside data is validated and asserted several times throughout. All data structures are asserted to be in working order before moving on to manipulating the data. The code is well commented and readable. All signs point to a code base which has been throughly reviewed by many.

Using valgrind proved quick and easy. For a project in development for 22 years, there should be no memory leaks of any kind, and there were none.

Lastly fuzz testing. Gzip is a data compression tool, and is designed to handle input of any kind in the case of compression. It would seem pointless to fuzz input data for compression. However, decompressing a gzip file is expected to be in a specific format in order to function. Naturally, the best option is to fuzz input data for decompression. Fortunately, gzip's years of development has managed to produce a reliable product. Fuzzing a single bit of any size file (anywhere from a few bytes to 3GB) resulted in an error handled by gzip gracefully. No matter

how many different files formats or even random data, fuzzing a single bit (not to mention several MB or a GB) would result in an error condition handled gracefully.

There is however one catch. The version of gzip used for this testing contains a buffer overflow vulnerability when used on an ftp server. This has been patched in gzip-1.4, but remains in the current version for consideration.

Therefore, my conclusion is gzip-1.4 is a reliable and solid tool. Gzip-1.3.12 is also a solid tool, so long as it is never used on an ftp server, but otherwise proves to be too great a risk for company wide adoption.