**ClamAV Vulnerability Report**

Hi,

I believe I found a potential Vulnerability in the latest version 1.1.0 of ClamAV.

**Report Info:**

Report Title: ClamAV 1.1.0 CRC Vulnerability Report

Report Date: August 1st, 2023

Report Author: Alec Yu, Weiqi Wang and David Lie from David Lie Group, University of Toronto ECE Department <https://security.csl.toronto.edu/>

Contact Information: [alec.yu@mail.utoronto.ca](mailto:alec.yu@mail.utoronto.ca)

**Vulnerability Details:**

Vulnerability Description:

This Vulnerability originates in the XzDec.c file in the function SRes XzUnpacker\_Code at the lines 759, 780, 792, 816, 831, 849, 869, 879, basically all instances where XzUnpacker\_Code identifies a CRC Error after it’s passed through the case XZ\_STATE\_BLOCK. Although XzUnpacker\_Code correctly identifies that an altered byte fails CRC, the function it returns to, cli\_XzDecode in the file xz\_iface.c, also takes into account the fact that the field XZ->status had been switched from CODER\_STATUS\_NOT\_SPECIFIED to CODER\_STATUS\_FINISHED\_WITH\_MARK in the XZ\_STATE\_BLOCK case of XzUnpacker\_Code. The cli\_XzDecode function, even though it knows that there was a CRC Error, first checks for XZ->status and decides that the value of this field holds precedence over the fact that XzUnpacker\_Code has returned a CRC\_ERROR. This ultimately results in ClamAV deciding that the compressed file it scanned contains no virus although CRC has determined that the data is corrupted.

Although footer parts of files don’t contain executable code, this can still be dangerous as a specially crafted input file with a payload hidden in the footer parts of a compressed file will be able to avoid the ClamAV scans. We believe that ClamAV should still be returning a data error instead of returning a scan passed with no issues found, as faulty decompression softwares may be exploited after thinking that the there is no chance for there to be viruses due to ClamAV’s scan.

**Reproduction Steps:**

Step-by\_Step instructions:

* Variable
* Function
* File Name

1. Load the input file xz-5.2.2-clamav-0.99.2-tgt-0-ivtpos-61-subfuzz-1-dis-1.xz that I’ve attached to the e-mail.
2. Place breakpoints in files XzDec.c at lines 648 and 758 (XzUnpacker\_Code), in xz\_iface.c at line 84 (cli\_XzDecode) and in scanners.c at lines 1402 and 1403 (cli\_scanxz).
3. Start the program in gdb and continue, which will take you to line 1402 in cli\_scanx.
4. Step into the function cli\_XzDecode in the file xz\_iface.c. We are now preparing to decompress the archive xz input file. Use the next command until line 69, which takes in a variable crucial to the bug, &XZ->status. Step into the XzUnpacker\_Code in the file XzDec.c and use the continue command to jump to line 648.
5. This is the end of the XZ\_STATE\_BLOCK, which runs from bytes 16-24 and consists of the main contents of the compressed XZ file. After decompressing this section, ClamAV sets the variable XZ->status from CODER\_STATUS\_NOT\_SPECIFIED to CODER\_STATUS\_FINISHED\_WITH\_MARK, presumedly because it has finished decompressing the main part of the file and does not think that there can be any viruses hidden in the footer parts of the file.
6. We then enter the command continue into gdb, taking us to line 758 in XzUnpacker\_Code.
7. We encounter a SZ\_ERROR\_CRC due to data corruption on byte 25, which is part of XZ\_STATE\_BLOCK\_FOOTER, and returns it to cli\_XzDecode in xz\_iface.c under the variable res. This would presumedly trigger a DATA\_ERROR return from ClamAV, however the following code suggests otherwise.
8. Enter the continue command which will take you to line 84 in cli\_XzDecode. This is in my opinion the root of the error, where instead of first considering the value of res which indicates that there was an error in the footer section of the archived xz file, it cares more about the fact that it has finished decompressing and storing the main block of the archived xz file which is indicated by the variable XZ->status being set to CODER\_STATUS\_FINISHED\_WITH\_MARK. Thus, instead of returning a XZ\_RESULT\_DATA\_ERROR to cli\_scanxz in scanners.c, it returns XZ\_STREAM\_END which is stored in the rc variable.
9. Enter the continue command, which should take you to line 1403 in cli\_scanxz in scanners.c. We are expecting ClamAV to tell us there is a data/decompression error which can still potentially happen even if cli\_XzDecode has returned XZ\_STREAM\_END, which would mean that ClamAV does not have a bug. We will soon see that this isn’t the case.
10. If variable rc were actually equal to XZ\_RESULT\_DATA\_ERROR , line 1403 would detect the issue and have clamAV return an error message. This is however not the case, meaning that after entering the next command to gdb, it will skip to line 1416.
11. Since one of the conditions is that rc == XZ\_STREAM\_END, it will go into the condition block upon entering the next command and go to line 1417.
12. Now, inside this block, there are two checks that can potentially make ClamAV return an error message that are at lines 1423 and 1428. Before going through these two checks however, ClamAV will first process two key variables, towrite and size.
13. towrite is set to the result of CLI\_XZ\_OBUF\_SIZE - strm.avail\_out, which results in 0 as they are the same value. This is because strm.avail\_out is set to CLI\_XZ\_OBUF\_SIZE once the main block of the archived file is decompressed.

A screenshot of a computer

Description automatically generated

1. Size is initially 0 and is updated to add the value of towrite, meaning that it remains as 0.
2. Now the interesting part comes with the cli\_writen function from file others\_common.c that takes three inputs: fd, buf and towrite
3. There are two cases:
   1. \*cli\_writen(fd, buf, towrite) != 0 🡪 returns an error message; this means that the actual ClamAV library functions as intended as does actually consider failed CRC checks to be an error.
   2. \*\*cli\_writen(fd, buf, towrite) == 0 🡪 does not return an error message; this means that there is probably a misimplementation in the ClamAV library as even though it detects the failed CRC check, it doesn’t consider it as an error even though it clearly is one
4. Now when we look at the cli\_writen function, we can see that it is based on the built-in c function write(), as that will decide whether or not it will return an error message and -1, which would cause case a\* from 16

A screen shot of a computer program

Description automatically generated

1. However, remember that we know towrite == 0, and count is simply towrite. todo is also initialized with the value of count, which is 0, and when we use the built-in c function write() with its 3rd argument (number of bytes it will process) being 0, write() will automatically return 0.
2. Thus, in the code inside the red box in the picture above, retval == 0 which is not smaller than 0, and cli\_writen will not return -1 with an error message.
3. Since todo is still 0, it isn’t bigger than 0 and thus we exit the do-while loop. Finally, we return count which still retains a value of 0 to the scanners.c file, indicating that case b\*\* from 16 might happen.
4. The next condition at line 1428 checks for file size, which is not an issue here.
5. We exit the loop since rc == XZ\_STREAM\_END without ever modifying ret from the initial value of CL\_CLEAN (== 0), meaning that ClamAV does not report any problems with the file even if there is one.

I have also confirmed this finding by simply running the version 1.1.0 of ClamAV on the corrupted file xz-5.2.2-clamav-0.99.2-tgt-0-ivtpos-61-subfuzz-1-dis-1.xz, and ClamAV finds nothing wrong with it:

A computer screen with white text

Description automatically generated

Test Environment using clamconf:

Config file: clamd.conf

-----------------------

AlertExceedsMax disabled

PreludeEnable disabled

PreludeAnalyzerName disabled

LogFile disabled

LogFileUnlock disabled

LogFileMaxSize = "1048576"

LogTime disabled

LogClean disabled

LogSyslog disabled

LogFacility = "LOG\_LOCAL6"

LogVerbose disabled

LogRotate disabled

ExtendedDetectionInfo disabled

PidFile disabled

TemporaryDirectory disabled

DatabaseDirectory = "C:\Users\alecy\Documents\Career\Jobs\Summer 2023 - DLGroup\clamav-1.1.0.win.x64\database"

OfficialDatabaseOnly disabled

FailIfCvdOlderThan disabled

LocalSocket disabled

LocalSocketGroup disabled

LocalSocketMode disabled

FixStaleSocket = "yes"

TCPSocket = "3310"

TCPAddr = "localhost"

MaxConnectionQueueLength = "200"

StreamMaxLength = "104857600"

StreamMinPort = "1024"

StreamMaxPort = "2048"

MaxThreads = "10"

ReadTimeout = "120"

CommandReadTimeout = "30"

SendBufTimeout = "500"

MaxQueue = "100"

IdleTimeout = "30"

ExcludePath disabled

MaxDirectoryRecursion = "15"

FollowDirectorySymlinks disabled

FollowFileSymlinks disabled

CrossFilesystems = "yes"

SelfCheck = "600"

ConcurrentDatabaseReload = "yes"

DisableCache disabled

VirusEvent disabled

ExitOnOOM disabled

AllowAllMatchScan = "yes"

Foreground disabled

Debug disabled

LeaveTemporaryFiles disabled

GenerateMetadataJson disabled

User disabled

Bytecode = "yes"

BytecodeSecurity = "TrustSigned"

BytecodeTimeout = "10000"

BytecodeUnsigned disabled

BytecodeMode = "Auto"

DetectPUA disabled

ExcludePUA disabled

IncludePUA disabled

ScanPE = "yes"

ScanELF = "yes"

ScanMail = "yes"

ScanPartialMessages disabled

PhishingSignatures = "yes"

PhishingScanURLs = "yes"

HeuristicAlerts = "yes"

HeuristicScanPrecedence disabled

StructuredDataDetection disabled

StructuredMinCreditCardCount = "3"

StructuredMinSSNCount = "3"

StructuredSSNFormatNormal = "yes"

StructuredSSNFormatStripped disabled

ScanHTML = "yes"

ScanOLE2 = "yes"

AlertBrokenExecutables disabled

AlertBrokenMedia disabled

AlertEncrypted disabled

StructuredCCOnly disabled

AlertEncryptedArchive disabled

AlertEncryptedDoc disabled

AlertOLE2Macros disabled

AlertPhishingSSLMismatch disabled

AlertPhishingCloak disabled

AlertPartitionIntersection disabled

ScanPDF = "yes"

ScanSWF = "yes"

ScanXMLDOCS = "yes"

ScanHWP3 = "yes"

ScanArchive = "yes"

ForceToDisk disabled

MaxScanTime disabled

MaxScanSize = "419430400"

MaxFileSize = "104857600"

MaxRecursion = "17"

MaxFiles = "10000"

MaxEmbeddedPE = "41943040"

MaxHTMLNormalize = "41943040"

MaxHTMLNoTags = "8388608"

MaxScriptNormalize = "20971520"

MaxZipTypeRcg = "1048576"

MaxPartitions = "50"

MaxIconsPE = "100"

MaxRecHWP3 = "16"

PCREMatchLimit = "100000"

PCRERecMatchLimit = "2000"

PCREMaxFileSize = "104857600"

OnAccessMountPath disabled

OnAccessIncludePath disabled

OnAccessExcludePath disabled

OnAccessExcludeRootUID disabled

OnAccessExcludeUID disabled

OnAccessExcludeUname disabled

OnAccessMaxFileSize = "5242880"

OnAccessDisableDDD disabled

OnAccessPrevention disabled

OnAccessExtraScanning disabled

OnAccessCurlTimeout = "5000"

OnAccessMaxThreads = "5"

OnAccessRetryAttempts disabled

OnAccessDenyOnError disabled

DevACOnly disabled

DevACDepth disabled

DevPerformance disabled

DevLiblog disabled

DisableCertCheck disabled

AlgorithmicDetection = "yes"

BlockMax disabled

PhishingAlwaysBlockSSLMismatch disabled

PhishingAlwaysBlockCloak disabled

PartitionIntersection disabled

OLE2BlockMacros disabled

ArchiveBlockEncrypted disabled

Config file: freshclam.conf

---------------------------

LogFileMaxSize = "1048576"

LogTime disabled

LogSyslog disabled

LogFacility = "LOG\_LOCAL6"

LogVerbose disabled

LogRotate disabled

PidFile disabled

DatabaseDirectory = "C:\Users\alecy\Documents\Career\Jobs\Summer 2023 - DLGroup\clamav-1.1.0.win.x64\database"

Foreground disabled

Debug disabled

UpdateLogFile disabled

DatabaseOwner = "clamav"

Checks = "12"

DNSDatabaseInfo = "current.cvd.clamav.net"

DatabaseMirror = "database.clamav.net"

PrivateMirror disabled

MaxAttempts = "3"

ScriptedUpdates = "yes"

TestDatabases = "yes"

CompressLocalDatabase disabled

ExtraDatabase disabled

ExcludeDatabase disabled

DatabaseCustomURL disabled

HTTPProxyServer disabled

HTTPProxyPort disabled

HTTPProxyUsername disabled

HTTPProxyPassword disabled

HTTPUserAgent disabled

NotifyClamd = "C:\Users\alecy\Documents\Career\Jobs\Summer 2023 - DLGroup\clamav-1.1.0.win.x64\clamd.conf"

OnUpdateExecute disabled

OnErrorExecute disabled

OnOutdatedExecute disabled

LocalIPAddress disabled

ConnectTimeout = "30"

ReceiveTimeout = "60"

Bytecode = "yes"

clamav-milter.conf not found

Software settings

-----------------

Version: 1.1.0

Optional features supported: MEMPOOL AUTOIT\_EA06 BZIP2 LIBXML2 PCRE2 JSON RAR

Database information

--------------------

Database directory: C:\Users\alecy\Documents\Career\Jobs\Summer 2023 - DLGroup\clamav-1.1.0.win.x64\database

bytecode.cvd: version 334, sigs: 91, built on Wed Feb 22 16:33:21 2023

daily.cvd: version 26973, sigs: 2039213, built on Tue Jul 18 03:28:30 2023

main.cvd: version 62, sigs: 6647427, built on Thu Sep 16 08:32:42 2021

Total number of signatures: 8686731

Platform information

--------------------

uname: Microsoft Windows 6.2 SP0.0 Build 9200

OS: Windows, ARCH: AMD64, CPU: AMD64

zlib version: 1.2.13 (1.2.13), compile flags: 65

platform id: 0x1025b4b4080000000000077c

Build information

-----------------

Microsoft Visual C++: (0.7.124)

sizeof(void\*) = 8

Engine flevel: 180, dconf: 180

Fuzzer Used:

Academic research tool described here: <https://security.csl.toronto.edu/wp-content/uploads/2022/08/wwang_mascthesis_2022.pdf>

We are happy to share the source code privately if you are interested.

Additionally, we have many more input files that reveal this bug as there are many CRC checks that are implemented but ignored in the XzUnpacker\_Code function of the XzDec.c file. I simply picked one that was affected by the first CRC check.

**Mitigation:**

Proposed Solution:

I would first invert the checks at lines 84 and 88 in order to check for errors redundancy check errors even if the main parts of the file has been decoded.

Since I am not very familiar with antiviruses in general, I do not know whether the footer parts of the file should also be scanned, as ClamAV does not seem to be doing it right now.