TrueCrypt Search and Decrypt Tool (tcsandd)

User Guide 1.0

2013.08.27.

Csaba Fitzl

Contents

[Overview 3](#_Toc365404275)

[Search operation 3](#_Toc365404276)

[Speed and decryption 3](#_Toc365404277)

[Logging 3](#_Toc365404278)

[CLI and GUI Overview 5](#_Toc365404279)

[File options 6](#_Toc365404280)

[Password options 6](#_Toc365404281)

[Keyfile support 7](#_Toc365404282)

[Usage examples: 7](#_Toc365404283)

[Start GUI 7](#_Toc365404284)

[Decrypt a single file with a given password 7](#_Toc365404285)

[Brute force a single file based on a password list: 8](#_Toc365404286)

[Search for files in the current directory 8](#_Toc365404287)

[Search for files in the current directory and try to decrypt them with a password 8](#_Toc365404288)

[Decrypt a single file using keyfiles, and single password 8](#_Toc365404289)

[Decrypt a single file using keyfiles, and dictionary 8](#_Toc365404290)

[Decrypt a single file using keyfiles, and charaters for password bruteforcing 8](#_Toc365404291)

[Decrypt files from a list using dictionary 8](#_Toc365404292)

# Overview

## Search operation

The tool is very fast in searching TC volumes. The search logic is the following:

* 1. The suspect file size modulo 512 must equal zero.
  2. The suspect file size is at least 256kB in size (this is the size of the headers + backup headers)
  3. The suspect file must not contain a common file header.
  4. The suspect file has entropy more than 7.6.

The search is actually looking for encrypted files, as it’s impossible to tell if a file is a TC volume until the correct password is supplied. Thus it can be used to look for other encrypted files like FreeOTFE.

Based on these rules, the search will find any possible encrypted file, not only TC. Proving that a file is actually a TC volume is impossible without decryption. If running it on the entire file system, it will find about 300 files, which are not real TC volumes at all, which is a very good false positive rate, considering that there are more than 200.000 files on a normal computer. (**This is only if we have the provided foremost configuration file set, to filter out known headers)**

An example Foremost header configuration file provided with the source code.

## Speed and decryption

The password tries are very slow compared to other tools like OTFBrutus (<http://www.tateu.net/software/dl.php?f=OTFBrutusGUI>), and the reason is that the hash and encryptions implemented in python are not so optimal. If we have only a couple of passwords to try, then the tool is good, but if not it will run for long time.

The tool can decrypt an entire TC volume (hidden as well) once the password is found. It’s again slow, however there is no other tool which can do that.

Because of speed, the recommended usage is the following:

1. Use this tool to find possible encrypted files
2. In case we have many passwords to try, use another tool to find the password, if not, use this tool
3. Use this tool, to make a decrypted copy of the volume.

Other notes:

* TrueCrypt keyfiles are supported for the decryption.
* Only files created with TrueCrypt version 6 and above are supported
* Hidden volumes supported

## Logging

The tool will log everything to the standard output, and to a file called “tcsandd.log”, so it can be searched later. It has a timestamp for each event, a log level as well. For example:

**2013-08-26 06:47:45,839 - tcsandd - INFO - Starting GUI**

**2013-08-26 06:51:59,762 - tcsandd.compute - ERROR - The specified path: doesn't exists**

**2013-08-26 06:52:45,026 - tcsandd.tcsearch.search\_file - DEBUG - C:/Users/csaby/Downloads/502\502.zip: File size or mod doesn't match**

**2013-08-26 06:52:45,028 - tcsandd.tcsearch.search\_file - DEBUG - C:/Users/csaby/Downloads/502\a.txt: File size or mod doesn't match**

# CLI and GUI Overview

The program has the following command line options:

**c:\tcsandd>tcsandd.exe -h**

**Usage: tcsandd.exe [options]**

**Options:**

**-h, --help show this help message and exit**

**-G, --gui Start GUI**

**-S, --search\_only Search only**

**-v, --verbose Verbose output**

**-s SINGLE, --single=SINGLE**

**Specify a single file**

**-f FILELIST, --filelist=FILELIST**

**File for file list**

**-p PATH, --path=PATH Path to search, default is the current directory**

**-c CONFIG, --config=CONFIG**

**Path to Foremost / Scalpel config file**

**-P PASSWORD, --password=PASSWORD**

**Single password to try**

**-d DICTIONARY, --dictionary=DICTIONARY**

**Dictionary to use for password attacks**

**-r REGULAR, --regular=REGULAR**

**Characters for password generation**

**-l LENGTH, --length=LENGTH**

**Maximum password length, default: 1**

**-k KEYFILEDIR, --keyfiledir=KEYFILEDIR**

**Directory of keyfiles**

**-B, --brutekeys Bruteforce keyfiles**

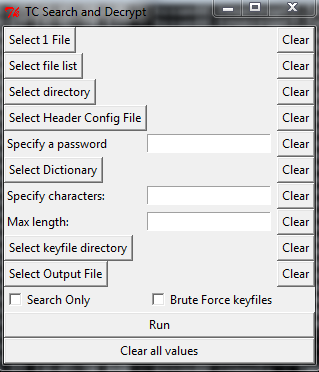
**-o OUTPUT, --output=OUTPUT**

**Output file (only in case of a single file specified)**

Here is the detailed meaning of each one:

* -h / --help: lists the help menu as shown above, other options will be ignored
* -G / --gui: starts the tool in GUI mode, other options will be ignored
* -S / --search\_only: the tool will search for TC files, without decrypting them, and will create a “foundfiles.txt” file, with all found files. This can be feed later to the tool for decryption. password, dictionary, regular, length, output, filelist, single options will be ignored.
* -v / --verbose: very detailed output about the progress, recommended if we want to see every step the tool makes
* -s / --single: the toll will try to decrypt a single, given file, filelist and path options will be ignored if used
* -f / --filelist: give a txt file with a lit of file (1 line / file) to try to decrypt, path options will be ignored if used
* -p / --path: give a path where to search for TC files, by default it will search in the current directory
* -c / --config: file for the scalpel / formost configuration file, if not given it will use some built in headers only
* -P / --password: give a password for TC file decryption, dictionary and regular options will be ignored if used
* -d / --dictionary: give a password list for TC file decryption, regular option will be ignored
* -r / --regular: provide a list of characters, what to test for decryption. All possible combinations will be tried.
* -l / --length: Max password lengths to try with the charters. Default is 1.
* -k / --keyfiledir: Directory of the keyfiles
* -B / --brutekeys: try all possible combinations of the keyfiles found int he keyfile directory
* -o / --output: Output file for decryption, valid only in case only a single file specified.

GUI mode:



We can see that we have the very same options (except verbose and GUI, which must be specified at the beginning) as with the command line. We can click on the buttons to select the specific files / destination. We can use the clear buttons to clear values. Once everything is set we can press on the “Run” button to start the search / decryption.

## File options

We can choose three methods for searching for files:

1. Single file, in that case only that file will be decrypted (CLI: -s, GUI: “Select 1 File”)
   1. If we choose this, we have the option to specify the output file name, in other cases, it’s ignored. (CLI: -o, GUI: “Select Output file”)
2. File list, where we give a text file containing 1 file / line (CLI: -f, GUI: “Select file list”)
3. Search a directory for encrypted files (CLI: -p, GUI: “Select directory”)
   1. If we want to search only, without trying to decrypt, we can specify a search only option. In that case the tool will save the list of found files to a file called “foundfiles.txt” (CLI: -S, GUI: “Search Only”)
   2. We can specify a Foremost / Scalpel header configuration file, so that the tool recognize known headers during search (CLI: -c, GUI: “Select Header Config File”)

By default all decrypted files got the following name: original filename + .decrypted extension, and they are placed in the directory he file was found. If the output exists, then the tool won’t overwrite it.

## Password options

We can specify the following options for passwords:

1. 1 password, the tool will try only that single password (CLI: -P, GUI: “Specify password” textbox)
2. dictionary file, the tool will try all password found in the file (CLI: -d, GUI: “Select dictionary”)
3. characters to brute force (CLI: -r, GUI: “Specify characters”)
   1. we can specify the max length (CLI: -l, GUI: “Max length” textbox)

The tool will use only one from all in the order as listed above. If none listed, it will throw an error.

## Keyfile support

The tool supports keyfiles with two options:

1. Read all files in a directory, and use all of them as a keyfile for a volume. **The order of the files are important, thus they have to be in the correct alphabetical order for processing.** (CLI: -k, GUI: “Select keyfile directory”)
2. Try to brute force all possible combinations of the keyfiles found in the previously given directory (CLI: -B, GUI: “Brute Force keyfiles”)

## Usage examples:

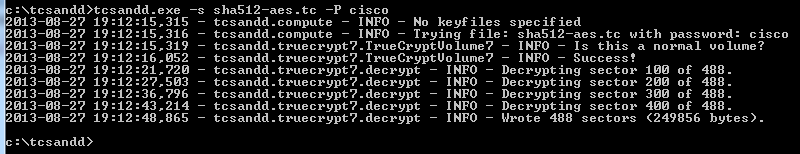
### Start GUI

**c:\tcsandd>tcsandd.exe -G**



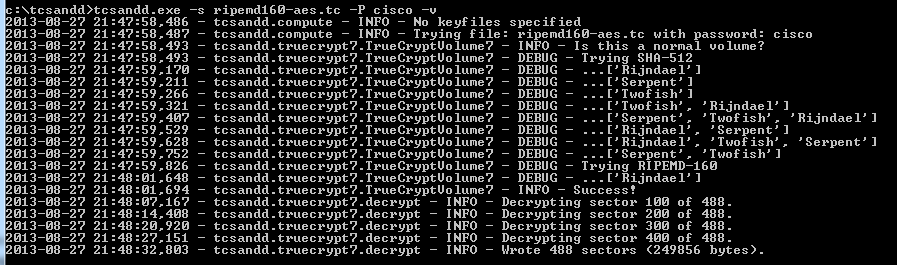
### Decrypt a single file with a given password

**tcsandd.exe -s sha512-aes.tc -P cisco**



The same in verbose mode:

**tcsandd.exe -s ripemd160-aes.tc -P cisco –v**

****

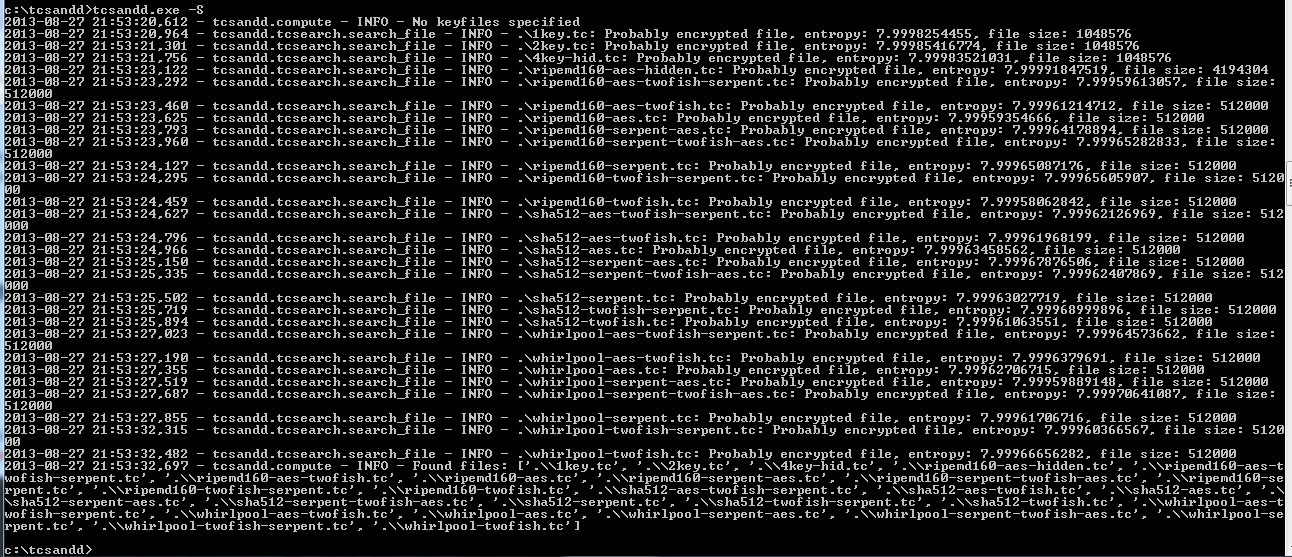
### Brute force a single file based on a password list:

**tcsandd.exe -s ripemd160-twofish.tc -d password.lst**

****

### Search for files in the current directory

**tcsandd.exe –S**

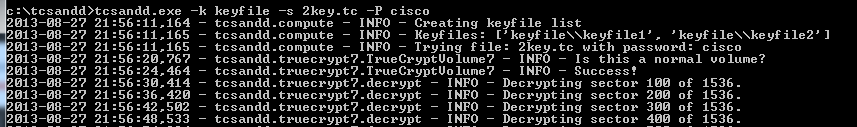


### Search for files in the current directory and try to decrypt them with a password

**tcsandd.exe -p . -P cisco**

### Decrypt a single file using keyfiles, and single password

**tcsandd.exe -k keyfile -s 2key.tc -P cisco**



### Decrypt a single file using keyfiles, and dictionary

**tcsandd.exe -k keyfile -s 2key.tc –d password.lst**

### Decrypt a single file using keyfiles, and charaters for password bruteforcing

**tcsandd.exe -k keyfile -s 2key.tc -r ciso**

### Decrypt files from a list using dictionary

**tcsandd.exe -f foundfiles.txt -d password.lst**