International Journal of Research
e-ISSN: 2348-6848 & p-ISSN 2348-795X Vol-5, Special Issue-11
International Conference on Multi-Disciplinary Research - 2017 held in February, 2018 in Hyderabad, Telangana State, India organised by GLOBAL RESEARCH ACADEMY - Scientific & Industrial Research Organisation (Autonomous), Hyderabad.

Utilization of TrueCrypt Software for Data Confidentiality through Encryption

Pushpanjali Patra[1], Prof.Dr.G.Manoj Someswar[2]

1. Research Scholar, VBS Purvanchal University, Jaunpur, U.P., India
2. Research Supervisor, VBS Purvanchal University, Jaunpur, U.P., India

Abstract

People are increasingly using data encryption software to ensure data confidentiality. One application which facilitates data encryption is the freely available and open-source software named TrueCrypt. Merely detecting encrypted data can be challenging for the digital forensic investigator as its content appears random when viewed. TrueCrypt magnitudes this difficulty by implementing two features, a hidden volume and a hidden operating system. When these features are used not only does the software provide data confidentiality through encryption, it lets people deny that data exists and this is often difficult for the forensic investigator to disprove.

Where use of data encryption is suspected, forensic investigators will typically try to gain access to the suspect's computer whilst it is powered on. In its powered on state, recovery from memory of password and key material may be possible or it could allow direct access to the data in a decrypted state. In this thesis, a security analysis of TrueCrypt, we examine a worst case scenario. In the scenario the forensic investigator only has access to the suspect computer's hard disk after the machine had been switched off for a considerable length of time and thus a memory capture or access to the data in a decrypted state was not possible. This research paper begins by evaluating existing statistical tests for their suitability in differentiating the encrypted TrueCrypt data from other non-encrypted data. A process model is defined which could be used by the forensic investigator to identify the encrypted data solely by analysis of the suspect hard disk's raw byte data content. The process model is applied to the problem of detecting a hidden volume or hidden operating system. In application and verification of the process model this thesis establishes a revised volume layout of the actual TrueCrypt volume, but ultimately the hidden volume and hidden operating system remained undetectable. Using existing forensic investigation techniques, this thesis examines the leaking of information which could aid the forensic investigator in establishing use of TrueCrypt to further strengthen the case against the suspect. Finally, I conclude that detection of the hidden volume and hidden operating system solely from analysis of the suspect computer's hard disk is still problematic for the forensic investigator.

Key words: ***Extensible Markup Language (XML),"TrueCryptVolume", Advanced Encryption Standard (AES), National Institute of Standards and Technology (NIST), TCanalyzer.***

Information leakage in TrueCrypt

Introduction

In this segment the legal picture was inspected to decide if any data spilled from the utilization of TrueCrypt which would enable the measurable agent to affirm the nearness of the shrouded volume or concealed working framework. The measurable pictures were made after a similar procedure itemized in segment. The two pictures mirrored the condition of the hard circle volume as dissected in this research paper.

Hard plate volume investigation of the concealed volume picture

The shrouded volume picture was stacked into the measurable investigation programming P2 Commander where the whole hard circle could be perused and examined at both the document framework level and at the crude plate byte level. The Windows Registry database, a various leveled database integral to the operation and design of Windows, was found. Understood zones known as hive keys were dissected for ordinarily held data.

TrueCrypt application execution confirm

The accompanying "UserAssist" Registry enter found in the NTUSER.DAT record was analyzed:
\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\75048700-EF1F-11D0-9888-006097DEACF9\Count.

This key contained subtle elements of utilizations which had already been executed on the running Windows framework. The information esteems contained in the "UserAssist" key were known to be encoded with ROT-1. Decay 13 is a Caesar figure whose move is 13 characters. The accompanying information was found under the "name" esteem: HRZR_EHACNGU:P:\Qbphzragf naq Frggvatf\Nqzvavfgengbe\Zl Qbphzragf\GehrPelcg7.1n\GehrPelcg.rkr.
Transformation of this information from ROT-13 brings about the accompanying information: UEME_RUNPATH:C:\Documents and Settings\Administrator\My Documents\TrueCrypt7.1a\TrueCrypt.exe.
This affirms an executable called TrueCrypt.exe was executed from the way appeared, regardless of the possibility that it had been along these lines erased. On the off chance that the TrueCrypt.exe document could be situated on plate at that point producing a MD5 hash of the executable and contrasting it with the executable accessible on the TrueCrypt site would enable the agent to affirm whether this was the genuine TrueCrypt executable instead of a record bearing a similar filename. The quantity of times this executable was executed is found in the "information" esteem. In this picture the esteem contains the hexadecimal esteems "02 00 09 00 C0 B4 CC 12 97 99 CE 01". The fifth hexadecimal esteem speaks to the quantity of times the application was executed, however the check begins from 05, in this way in this occasion TrueCrypt.exe has been begun 4 times.

As a component of Windows graphical UI (GUI) applications are shown in the GUI as symbols, pictures which speak to the record's substance or utilize. Windows reserves symbols on a for every client premise in a record called IconCache.db and as applications are executed

their symbols together with the owning application reference is put away inside this current document's substance.[1] The IconCache.db document was found in the accompanying area in the record framework: /Documents and Settings/Administrator/Local Settings/Application Data/.

Utilizing P2 officer to see the record information as hexadecimal characters a reference to TrueCrypt.exe is discovered, this can be seen highlighted in Figure1.

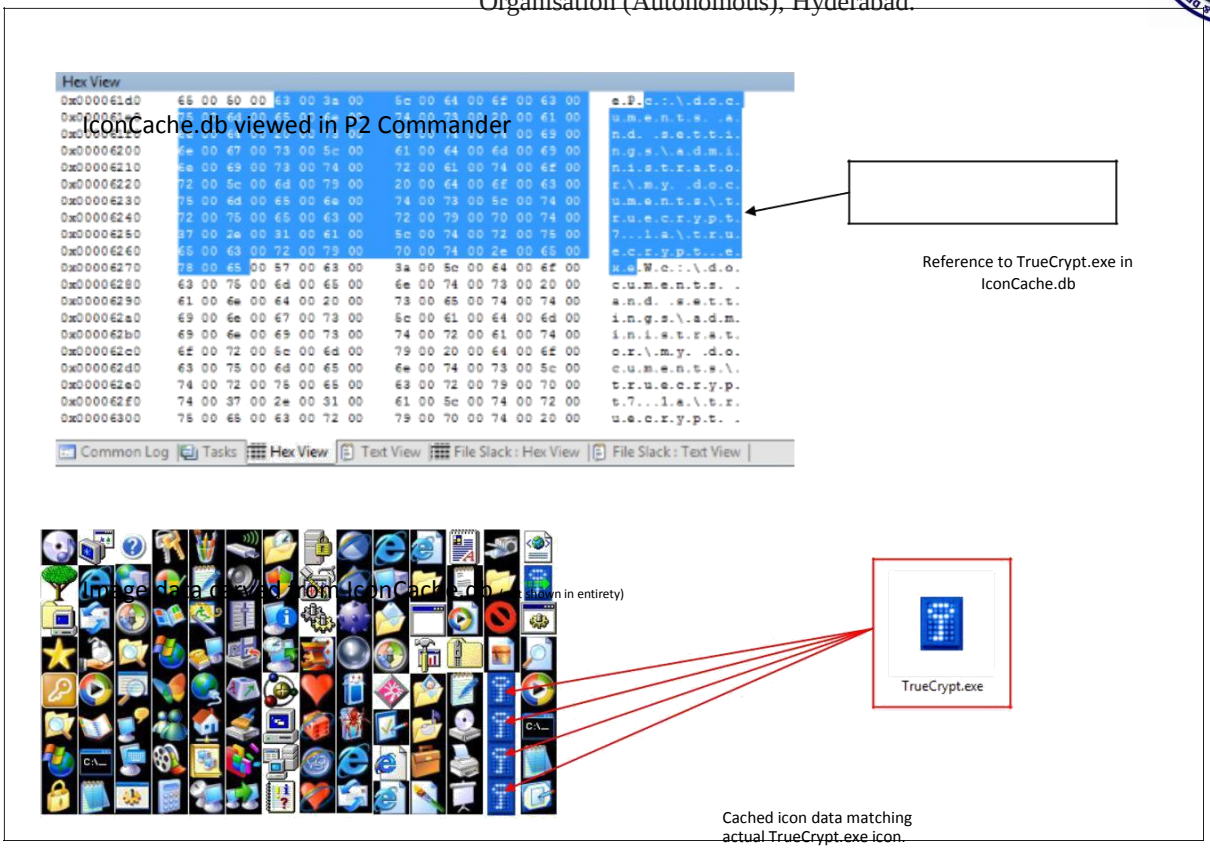

Figure 1: Analysis of IconCache.db

Utilizing a record cutting utility the bitmap picture information speaking to various symbols was separated. A correlation of the extricated symbol picture information and the TrueCrypt.exe record's symbol demonstrated various matches which can be found in Figure1. Given that the symbols and reference to TrueCrypt.exe were found in this document, it gives extra proof to the measurable examiner that TrueCrypt has sooner or later been utilized on this establishment of Windows.[2] The IconCache.db would not contain such information as a matter of course.

Evidence of mounting TrueCrypt volumes

One of the clearest signs that TrueCrypt has been utilized on this establishment of Windows could be found in the Registry area "\system\MountedDevices" of the system.dat document. This area tracks volumes which have been utilized inside Windows. Changing over the hexadecimal esteems found in the "information" key to ASCII we discover various references alluded to as "TrueCryptVolume". For this legal picture eight occurrences were found, in particular:

1. "TrueCryptVolumeT",
2. "TrueCryptVolumeZ",
3. "TrueCryptVolumeZ",
4. "TrueCryptVolumeT",
5. "TrueCryptVolumeS",
6. "TrueCryptVolumeS",
7. "TrueCryptVolumeH",

8. "TrueCryptVolumeH".

These references vary from references found for removable media added to the framework and do relate to the mounting of TrueCrypt volumes inside the Windows working framework. The letter taking after "TrueCryptVolume" compares with the letters which were known to be doled out to both the standard and shrouded TrueCrypt volumes at the season of mounting them. This confirmation demonstrates that a TrueCrypt volume had been mounted, yet it was impractical to tell whether a standard or shrouded TrueCrypt volume had been mounted.

Recently got to record prove

Generally Windows and some of its applications keep up a rundown of documents as of late got to. The as of late got to records' rundown can be found in the registry. References were found for some of the example records made inside or gotten to from the standard and concealed TrueCrypt volumes. These were found in the NTUSER.DAT registry document under the accompanying keys:

1. \Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs.

2. \Software\Microsoft\Windows\CurrentVersion\Applets\Wordpad\Recent File List.

3. \Software\Microsoft\Windows\CurrentVersion\Applets\Paint\Recent File List.

The qualities inside each key demonstrated filenames of archives gotten to from inside the mounted TrueCrypt volumes. Critically the principal key incorporated the mounted's volume mark. In the event that a suspect was required to mount a TrueCrypt volume under a lawful demonstration, for example, RIPA and mounted the less mystery standard TrueCrypt volume whose volume name did not coordinate the one shown in this registry key, they would require a conceivable clarification with respect to why the volume names were distinctive. One such clarification could be essentially that "I changed the volume names name". The documents recorded in each key just reference a volume letter doled out to the filename. The volume letter demonstrated would should be corresponded against the volume letters found in area Indeed, even with such connection it is feasible for a client to mount the standard TrueCrypt volume and allocate it a similar volume letter as one already alloted to a concealed volume, therefore giving conceivable cover to the shrouded volumes presence.

TrueCrypt XML arrangement record confirm

Utilizing P2 administrator programming the unallocated space of the Windows framework volume was inspected for TrueCrypt ancient rarities. The remaining parts of a TrueCrypt Extensible Markup Language (XML) setup document were found. The substance of the setup document are appeared in Figure2. The setup document gave additional confirmation to the measurable specialist that TrueCrypt had been utilized on this machine, yet it didn't give any further sign in the matter of whether a concealed volume was being used.

```
<?xml version="1.0" encoding="utf-8"?>
<TrueCrypt>
      <configuration>
            <config key="StartOnLogon">0</config>
            <config key="HiddenSectorDetectionStatus">0</config>
            <config key="SaveVolumeHistory">0</config>
            <config key="SecurityTokenLibrary"></config>
            <config key="Language"></config>
            <config key="OpenExplorerWindowAfterMount">0</config>
            <config key="UseDifferentTrayIconIfVolumesMounted">1</config>
            <config key="CachePasswords">0</config>
            <config key="WipePasswordCacheOnExit">0</config>
            <config key="WipeCacheOnAutoDismount">1</config>
            <config key="MountDevicesOnLogon">0</config>
            <config key="MountFavoritesOnLogon">0</config>
            <config key="MountVolumesReadOnly">0</config>
            <config key="MountVolumesRemovable">0</config>
            <config key="PreserveTimestamps">1</config>
            <config key="EnableBackgroundTask">1</config>
            <config key="CloseBackgroundTaskOnNoVolumes">0</config>
            <config key="DismountOnLogOff">1</config>
            <config key="DismountOnPowerSaving">0</config>
            <config key="DismountOnScreenSaver">0</config>
            <config key="ForceAutoDismount">1</config>
            <config key="MaxVolumeIdleTime">-60</config>
            <config key="UseKeyfiles">0</config>
            <config key="LastSelectedDrive"></config>
            <config key="CloseSecurityTokenSessionsAfterMount">0</config>
            <config key="DisableSystemCrashDetection">0</config>
            <config key="HotkeyModAutoMountDevices">0</config>
            <config key="HotkeyCodeAutoMountDevices">0</config>
            <config key="HotkeyModDismountAll">0</config>
            <config key="HotkeyCodeDismountAll">0</config>
            <config key="HotkeyModWipeCache">0</config>
            <config key="HotkeyCodeWipeCache">0</config>
            <config key="HotkeyModDismountAllWipe">0</config>
            <config key="HotkeyCodeDismountAllWipe">0</config>
            <config key="HotkeyModForceDismountAllWipe">0</config>
            <config key="HotkeyCodeForceDismountAllWipe">0</config>
            <config key="HotkeyModForceDismountAllWipeExit">0</config>
            <config key="HotkeyCodeForceDismountAllWipeExit">0</config>
            <config key="HotkeyModMountFavoriteVolumes">0</config>
            <config key="HotkeyCodeMountFavoriteVolumes">0</config>
            <config key="HotkeyModShowHideMainWindow">0</config>
            <config key="HotkeyCodeShowHideMainWindow">0</config>
            <config key="HotkeyModCloseSecurityTokenSessions">0</config>
            <config key="HotkeyCodeCloseSecurityTokenSessions">0</config>
            <config key="PlaySoundOnHotkeyMountDismount">1</config>
            <config key="DisplayMsgBoxOnHotkeyDismount">1</config>
      </configuration>
</TrueCrypt>
```

Figure 2: True Crypt XML configuration file

Conclusions

From confirmation found in the "User Assist" Registry key and Icon Cache.db record it was conceivable to unquestionably discover that the TrueCrypt application had sooner or later in history been begun on this occurrence of Windows.

In our test situation, simply utilizing standard Windows applications and usefulness, different references were discovered which logged information gotten to from areas inside a TrueCrypt volume, standard or covered up. Be that as it may, despite the fact that references were recorded, the scientific examiner couldn't conclusively credit their area to a TrueCrypt volume. This would just be derivation.

Of most use to a scientific examiner was the Registry area \system\MountedDevices. The information put away in this key could affirm the mounted volume to be a TrueCrypt volume. It doesn't however separate between the standard and concealed TrueCrypt volume sorts.

Hard plate volume investigation of the shrouded working framework volume picture

The shrouded working framework analyzed in situation 4 was stacked into the criminological investigation programming P2 Commander. The allotments containing the imitation working framework, segment 0 and the shrouded working framework, parcel 1 were both accurately shown. The substance of both allotments when seen in P2 Commander appeared to just contain irregular aimless information. Exhibit in the unallocated territory toward the begin of the circle which spoke to the ace boot record zone, confirmation of the TrueCrypt boot loader was found and is indicated highlighted in Figure3. No other information identifying with TrueCrypt could be found.

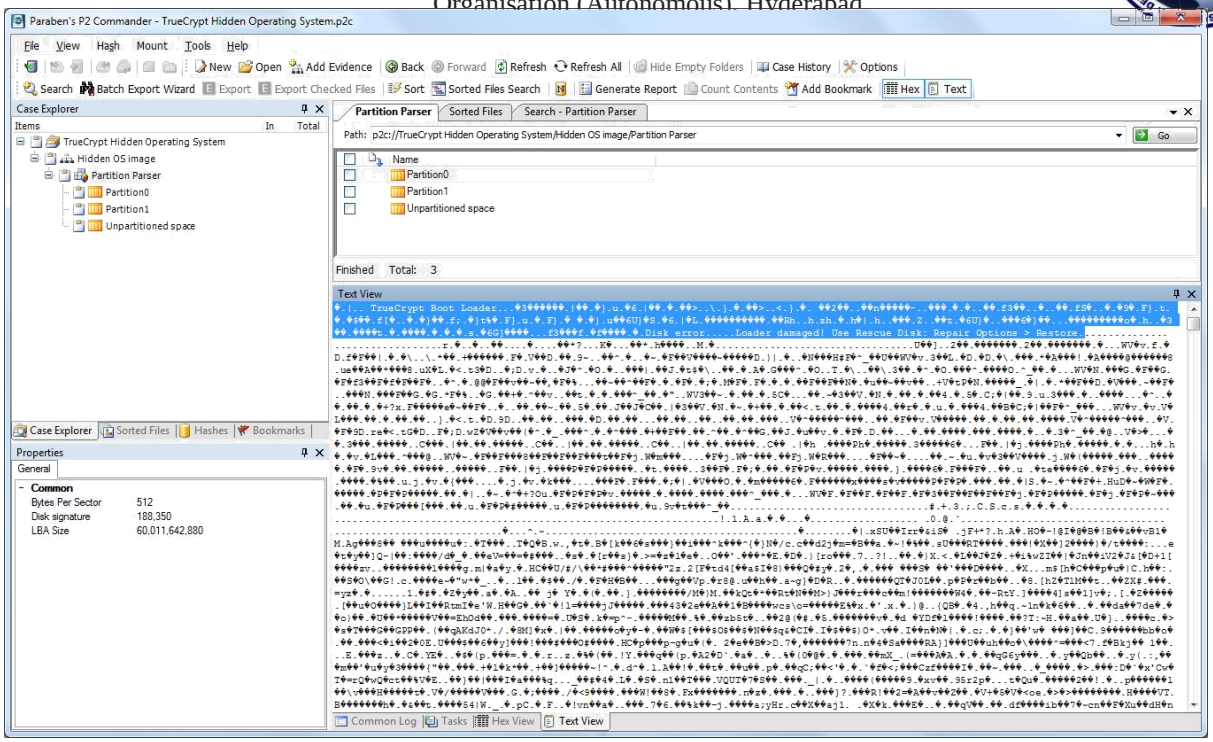

Figure 3: View of hard disk volume containing a hidden operating system

[The boot loader gave confirm that TrueCrypt was being used, yet it didn't give prove that a concealed volume existed.[3] The way that the segments just seemed to contain futile irregular information was normal and presence of an imitation or shrouded working framework was not clear.]

Conclusion

The expressed motivation behind the shrouded working framework arrangement is, to decisively ensure against the sort of data spillage seen in segment. The shrouded working framework highlight of TrueCrypt accomplishes this goal and would likewise secure against the sort of outsider application information spills which are reported in the paper by Czeskis, et al.

TrueCrypt volume header detection

This section will examine the volume headers of a standard and hidden TrueCrypt volume and headers used when TrueCrypt's hidden operating system feature is used.

Standard and hidden volume header analysis

As a feature of the investigation completed for Scenario 3, it was conceivable to reconsider our comprehension of the TrueCrypt volume design. Utilizing the program TCanalyzer the genuine volume format was resolved.[4] The format included areas of header data for the standard and shrouded TrueCrypt volumes and their individual reinforcement headers. The initial 131,072 bytes toward the begin of the parcel contained a typical standard TrueCrypt volume header of 65,536 bytes taken after by a

concealed volume header additionally of 65,536 bytes. In the event that no concealed volume existed then this second header simply contained arbitrary information when seen at the crude byte level. The last 65,536 bytes of the parcel contained a reinforcement of the concealed volume header and the procedure 65,536 bytes contained a reinforcement header of the standard TrueCrypt volume.

Again if no shrouded volume existed the reinforcement concealed volume header additionally contained irregular information. With information of the header structures and beginning counterbalanced of the concealed volume it was conceivable to overwrite quite recently the information contained in the standard TrueCrypt volume.[5] Subsequent to erasing this information, the standard volume couldn't be mounted inside TrueCrypt, however the shrouded volume stayed open. This affirmed the shrouded volume information just exists in the limits of the standard volume and not as cited by Czeskis, et al. "put inside non-concealed, general encoded volumes".

The main piece of a header which remained decoded was the initial 64 bytes of every header sort, standard, covered up or relating reinforcement headers. As can be seen from the salt esteems appeared in Table 5, extricated from the legal picture utilizing TCanalyzer, the salt esteems for reinforcement headers contrast from the first standard and concealed esteems. In the event that the qualities continued as before then it would be workable for the criminological specialist to decide likely volume degrees. Degrees could be dictated via hunting down sets of coordinating 64 bytes isolated by at any rate the base volume size of a given parcel.[6] As the salt esteems vary, this was not a practical choice.

Volume header type	Salt 64 bytes
Standard	e7 48 80 88 ef 71 bf d6 0b 01 3e b7 c2 b9 7a 26 8c 62 e9 f5 33 b7 91 1b 64 30 02 61 31 1a e7 69 55 89 e8 ea a4 1a 3f 38 d4 4d 71 89 3c ac 20 De 9c 5a 39 39 52 61 59 53 86 32 aa 88 fb d5 2b b4
Hidden	6b 29 20 db 47 cd 4a b5 2a 3f 25 2f 19 25 ed 14 47 04 00 04 7f 8f 84 b4 ad e0 8e 7e 24 a4 d4 d7 85 5c c2 6d 92 f7 6d 1b 76 f2 03 48 40 6b 25 a9 96 d9 78 5f c1 ec 35 3a 03 1d 87 f4 4c 24 6c 4c
Standard backup	a2 a1 5a 34 ea 61 f6 6f b0 9e 9f 46 01 09 83 49 83 3c 08 ae 38 0a a9 d5 6b c7 18 f3 ef 0d c6 Ea 62 e0 77 61 6a 1b b1 d8 ec d1 4a fa 1f 0a e3 8c cf 21 53 96 4c de b7 9e b9 e1 f6 61 b0 01 d8 8c
Hidden backup	30 52 4f 71 3f 5f 40 a8 85 32 f3 82 c0 85 b7 26 b0 7b ad d1 e3 fe d4 f3 73 91 37 39 35 7d 0b a8 d4 d2 46 52 a9 50 28 75 2d 1b 7d 8a 47 6d f3 f7 c2 72 53 13 e5 a8 0b 7e 7d ae 5f 34 55 b4 da 48

Table 1: Volume headers "salt"

Conclusions

A surprising outcome from the preliminary work performed to permit examination of the TrueCrypt headers has been a more prominent comprehension of the relationship of a TrueCrypt shrouded volume to the standard volume. It is off base to state the shrouded volume information zone is inside the standard TrueCrypt volume, this suggests a dependence on the standard volume. A more exact definition is say the concealed volume information zone exists in the degrees of the standard volume information range.[7] The shrouded volume is not subject to the standard volume. This was affirmed by erasing information from the standard volume yet leaving the region possessed by the concealed volume. Concealed volume operations proceeded with typically while the standard volume neglected to mount.

Despite the fact that the salt is decoded, it is produced utilizing TrueCrypt's irregular number era handle as examined before in area. Being arbitrary in nature there was no perceptible distinction between the salt and the rest of information in the TrueCrypt volume. As there was no distinction it was inconceivable for the legal agent to recognize headers of any sort. The main choice left to the agent is an animal constrain assault on the client key of a suspected TrueCrypt volume. The assault would make suspicions about the area of salt inside the parcel and join this with a word reference of pre-produced passwords to influence the assault. On the off chance that the client picked a solid secret word, consolidated with salt of bits, a savage compel assault could set aside a to a great degree long opportunity to finish, if by any means.

Hidden working framework volume header investigation

A similar volume scientific picture utilized as a part of Scenario 3 was utilized for this current segment's investigation. It was watched that the TrueCrypt boot loader was effectively unmistakable in the ace boot record of the hard plate volume as appeared in Figure3. The investigation additionally demonstrated that this piece was hailed not as containing encoded information, but rather the measurable information recommended that it could be compacted information. A record cutting utility was utilized to scan the whole picture for packed document groups, ZIP, RAR and GZIP and concentrate the information taking after a match. Numerous false positive were experienced, yet 2 legitimate files were found at byte counterbalances 2,560 and 17,920 inside the legal picture. The chronicles were coordinated utilizing the GZIP document record mark of hex 1F 8B 08. The documents' substance were effectively extricated and seen in a hexadecimal proofreader. Both documents' substance were indistinguishable and additional proof, which could be ascribed to the TrueCrypt boot loader, was found. The content found can be seen set apart in Figure3.

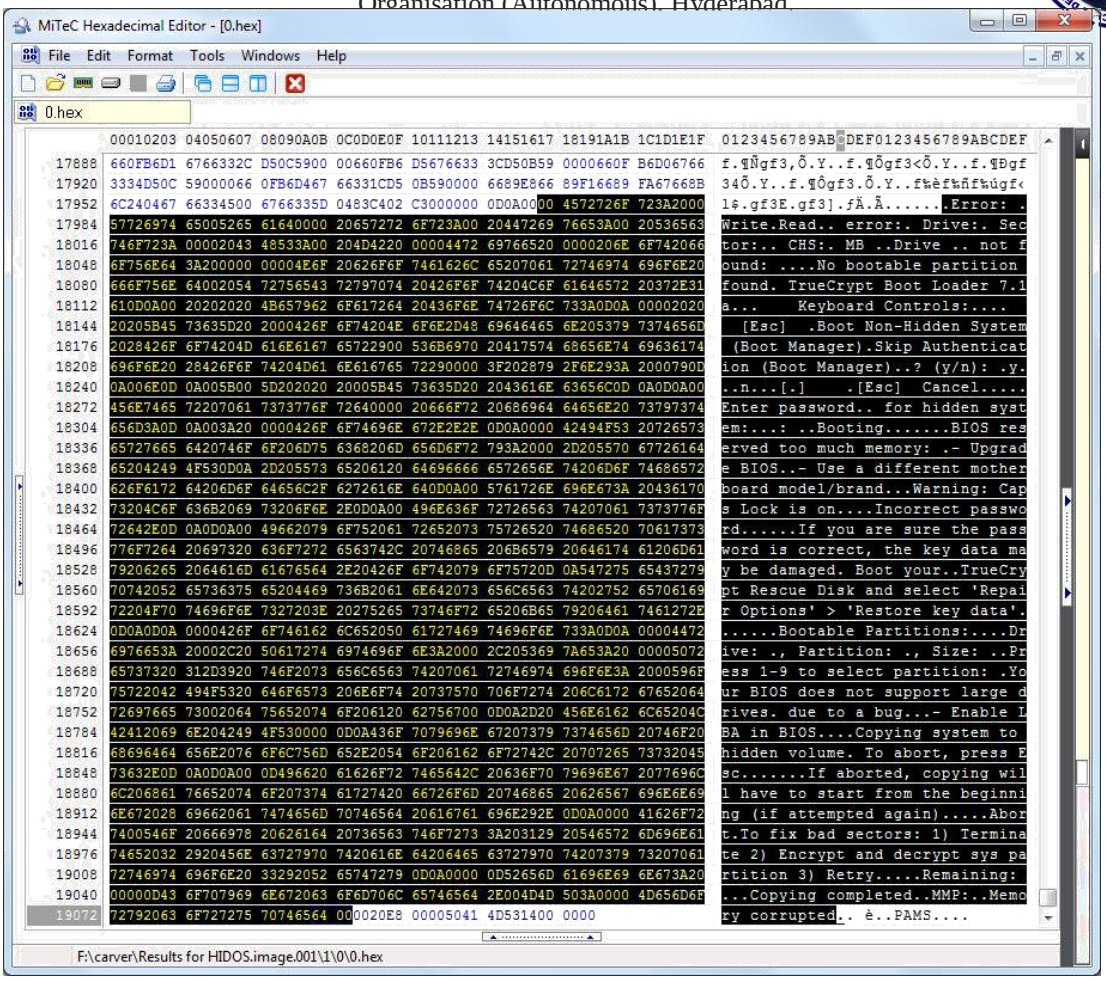

Figure 4: Evidence of the TrueCrypt boot loader found in compressed data

Measurably no distinction was seen at the normal header areas of the imitation working framework and the shrouded working framework allotments. Inside each different parcel no distinction was seen between the header and information ranges subsequently the headers stayed imperceptible in the criminological picture. From examination of the measurable picture, the nearness of the boot loader and the substance of the packed information affirm that TrueCrypt had been utilized with this hard plate eventually already. No additional confirmation of TrueCrypt headers was discovered which could be utilized to decide if the volume contained a shrouded working framework or not. The reasons why the headers couldn't be distinguished were the same as those finished up in area. It was just conceivable to reason that the information was irregular, and that the segment information displayed the same factual properties which were anticipated from TrueCrypt information.[8]

Conclusions

This postulation has exhibited by means of use of the procedure display that various measurable tests can be utilized together and connected to a crude legal picture to arrange information into either being suspected as TrueCrypt or not. The review effectively recognized the typical hard plate parcels from a scrambled TrueCrypt segment. In refining the procedure demonstrate, the span of piece examining was observed to be essential.[9] The square specimen estimate influenced the location of the scrambled information. It was additionally a calculate making the procedure display workable from the viewpoint of information administration. On the off chance that the square specimen size was too little the legal specialist would be defied with a huge number of tests and the procedure would be unworkable. This could likewise cause issues when bringing in the measurable outcomes into some spreadsheet applications for examination, if the outcomes surpassed the most extreme number of columns allowed in the application.

Tending to target one of this theory the standard TrueCrypt volume could be distinguished measurably. Be that as it may, when utilizing just a solitary scientific picture, the shrouded volume which lay inside the standard volume degrees couldn't be recognized by arbitrariness testing. On the off chance that different legal pictures were accessible then a change in factual outcomes could be watched, however the TrueCrypt documentation as of now cautions clients about this potential hazard. It ought to be noticed that a similar change would likewise be obvious if simply contrasting various legal pictures of the hard plate information at the crude byte level and all things considered was not one of a kind to the measurable outcomes. The scientific agent would just sit around idly in the measurable procedure creating these insights.

The procedure show gave a level of certainty that the inspected information contained presumed TrueCrypt information. I utilize the word speculated in light of the fact that the model all the more precisely orders information as either scrambled or not. The legal specialist would need to give extra proof to bolster their view that TrueCrypt was being used. This extra confirmation would rely on upon the mode in which TrueCrypt had been utilized and what data had spilled. Inside segment 5 it was shown utilizing standard measurable process procedures that in a few conditions it was conceivable to distinguish when the TrueCrypt application had been executed in the working framework and when TrueCrypt volumes had been mounted. Notwithstanding when the whole working framework's parcel had been encoded, the boot loader gave prove that TrueCrypt had been utilized. What couldn't be said completely in reply to the second goal was that the spilled data conclusively affirmed the nearness of the concealed volume or shrouded working framework; rather it would just expand doubt of their utilization. It was impractical to recognize the nearness of the TrueCrypt volume headers, either standard or covered up.[10] In noting the last target, as the headers couldn't be identified

they couldn't be utilized to identify a shrouded volume or concealed working framework.

In outline, the TrueCrypt item accomplishes what it embarks to do, in particular keeping up information classification and furnishing the client with the capacity to deny information's presence. On the off chance that the suggested security safeguards are tailed it is to a great degree improbable that the measurable specialist could identify the concealed TrueCrypt volumes or shrouded working framework construct exclusively in light of a criminological picture taken while the presume's PC was turned off.

Difficulties experienced

Toward the begin of this review I visualized that the essential application for the measurable examination would be the NIST Statistical Test Suite.[15] After effective arrangement of the source code on an Apple Mac PC, utilizing the NIST provided test information I took after the guidelines in the going with documentation to affirm the product's right operation. My outcomes did not relate to those in the documentation. The yield delivered by my gathered rendition of STS was additionally unique to that found in the documentation. This drove me to the conclusion that the going with documentation was obsolete.

As I was not able confirm rectify operation of the STS application I depended exclusively upon ENT[41], the factual test suite which I initially conceived utilizing to approve the outcomes from STS. It happened that not at all like STS, ENT could be scripted and the program bolstered a classified information yield design which could without much of a stretch be foreign made into spreadsheets for further investigation. Without the capacity to script the way toward creating the factual information I question whether my examination would have been plausible.

Initially I utilized the Linux order "split" to area my 55 GB scientific picture into the littler records which spoke to my example pieces. While testing the impact of piece sizes on the measurable outcomes I encountered surprising mistakes. Facilitate examination drove me to presume that the mistakes happened in the part procedure when utilizing the "split" order with little pieces sizes of 512 bytes. I couldn't decide when this summon brought about inaccurate yield thus embraced the utilization of the Swiss File Knife application.

I likewise experienced challenges with the document framework on my capacity media utilized when testing the impact of piece size on the investigation procedure. I knew about document registry limits for the FAT record framework, additionally experienced comparative issues when utilizing FAT32 at around 500,000 documents for each catalog.[12] I at that point changed over the capacity gadget to the NTFS record framework which worked without issue.

Future Work

TrueCrypt underpins various conceivable encryption calculations, for subtle elements see Table 1, and the situations could be re-keep running with every calculation to decide if the outcomes stayed predictable over every calculation sort.

Despite the fact that the TrueCrypt source code is uninhibitedly accessible for accumulation, it would not be conceivable to arrange this code to create an executable whose document hash coordinates the same executable accessible from the TrueCrypt site. The reason is that the pre-incorporated executable documents are carefully marked with the TrueCrypt Foundations certificate[7] which is not freely accessible. Despite the fact that the record hashes would not coordinate, it would be worth all the while seeing the self-accumulated and pre-assembled programs execution inside a program debugger to guarantee that every form executes in a steady way.

REFERENCES

1. The National Institute of Standards and Technology, "http://csrc.nist.gov/groups/ST/toolkit/rng/documents/sts-2.1.1.zip," 2.1.1 ed: NIST, August 11, 2010.

2. John Walker, "ENT," ed. http://www.fourmilab.ch/random/random.zip: John Walker, 2008.

3. R. Lyda and J. Hamrock, "Using entropy analysis to find encrypted and packed malware," *Security & Privacy, IEEE,* vol. 5, pp. 40-45, 2007.

4. M. Ponsen, P. Spronck, H. Muñoz-Avila, and D. W. Aha, "Knowledge acquisition for adaptive game AI," *Science of Computer Programming,* vol. 67, pp. 59-75, 6/1/ 2007.

5. International Telecommunications Union, "Measuring the Information Society," Place des Nations CH-1211 Geneva Switzerland2012.

6. P. L'Ecuyer, "Uniform random number generation," *Annals of Operations Research,* vol. 53, pp. 77-120, 1994.

7. P. L'Ecuyer, "Software for uniform random number generation: Distinguishing the good and the bad," in *Simulation Conference, 2001. Proceedings of the Winter*, 2001, pp. 95-105.

8. N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity of 802.11," presented at the Proceedings of the 7th annual international conference on Mobile computing and networking, Rome, Italy, 2001.

9. C. Ellison, "Cryptographic Random Numbers," *Draft P1363 Appendix E.http://www.std.com/~cme/P1363/ranno.html*, 2007.

10. P. Gutmann, "Software generation of practically strong random numbers," in*Proceedings of the Seventh USENIX Security Symposium*, 1998, pp. 243-257.

11. R. Morris and K. Thompson, "Password security: a case history," *Commun. ACM*, vol. 22, pp. 594-597, 1979.

12. C. E. Shannon, "A mathematical theory of communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, pp. 3-55, 2001.

13. M. M. Shannon, "Forensic relative strength scoring: ASCII and entropy scoring,"*International Journal of Digital Evidence*, vol. 2, pp. 151-169, 2004.

14. ASR Data. (2013, Accessed 7 August 2013). *SMARTLinux*.Available: http://www.asrdata.com/forensic-software/smart-linux/

15. StahlWorks Technologies, "Swiss File Knife,"1.6.8ed. http://sourceforge.net/projects/swissfileknife/files/1-swissfileknife/1.6.8/: StahlWorks Technologies, 2013.