

# Volatility Labs

Tuesday, January 14, 2014

## TrueCrypt Master Key Extraction And Volume Identification

One of the [disclosed](#) pitfalls of TrueCrypt disk encryption is that the master keys must remain in RAM in order to provide fully transparent encryption. In other words, if master keys were allowed to be flushed to disk, the design would suffer in terms of security (writing plain-text keys to more permanent storage) and performance. This is a risk that suspects have to live with, and one that law enforcement and government investigators can capitalize on.

The default encryption scheme is AES in XTS mode. In XTS mode, primary and secondary 256-bit keys are concatenated together to form one 512-bit (64 bytes) master key. An advantage you gain right off the bat is that patterns in AES keys can be distinguished from other seemingly random blocks of data. This is how tools like [aeskeyfind](#) and [bulk\\_extractor](#) locate the keys in memory dumps, packet captures, etc. In most cases, extracting the keys from RAM is as easy as this:

```
$ ./aeskeyfind Win8SP0x86.raw
f12bffe602366806d453b3b290f89429
e6f5e6511496b3db550cc4a00a4bdb1b
4d81111573a789169fce790f4f13a7bd
a2cde593dd1023d89851049b8474b9a0
269493cfc103ee4ac7cb4dea937abb9b
4d81111573a789169fce790f4f13a7bd
4d81111573a789169fce790f4f13a7bd
269493cfc103ee4ac7cb4dea937abb9b
4d81111573a789169fce790f4f13a7bd
0f2eb916e673c76b359a932ef2b81a4b
7a9df9a5589f1d85fb2dfc62471764ef47d00f35890f1884d87c3a10d9eb5bf4
e786793c9da3574f63965803a909b8ef40b140b43be062850d5bb95d75273e41
Keyfind progress: 100%
```

Several keys were identified, but only the two final ones in red are 256-bits (the others are 128-bit keys). Thus, you can bet by combining the two 256-bit keys, you'll have your 512-bit master AES key. That's all pretty straightforward and has been documented in quite a few places - one of my favorites being [Michael Weissbacher's blog](#).

The problem is - what if suspects change the default AES encryption scheme? TrueCrypt also supports Twofish, Serpent, and combinations thereof (AES-Twofish, AES-Twofish-Serpent). Furthermore, it supports modes other than XTS, such as LWR, CBC, outer CBC, and Inner CBC (though many of the CBCs are either deprecated or not recommended).

What do you do if a suspect uses non-default encryption schemes or modes? You can't find Twofish or Serpent keys with tools designed to scan for AES keys -- that just doesn't work. As pointed out by one of our Twitter followers ([@brnocrist](#)), a tool by Carsten Maartmann-Moe named [Interrogate](#) could be of use here (as could several commercial implementations from Elcomsoft or Passware).

Another challenge that investigators face, in the case of file-based containers, is figuring out which file on the suspect's hard disk serves as the container. If you don't know that, then having the master keys is only as useful as finding the key to a house but having no idea where the house is.

To address these issues, I wrote several new Volatility plugins. The [truecryptsummary](#) plugin gives you a detailed description of all TrueCrypt related artifacts in a given memory dump. Here's how it appears on a test system running [64-bit Windows 2012](#).

```
$ python vol.py -f WIN-QBTA4959AO9.raw --profile=Win2012SP0x64 truecryptsummary
Volatility Foundation Volatility Framework 2.3.1 (T)
```

```
Process           TrueCrypt.exe at 0xfffffa801af43980 pid 2096
Kernel Module     truecrypt.sys at 0xfffff88009200000 - 0xfffff88009241000
Symbolic Link      Volume{52b24c47-eb79-11e2-93eb-000c29e29398} -> \Device\TrueCryptVolumeZ mounted
2013-10-11 03:51:08 UTC+0000
Symbolic Link      Volume{52b24c50-eb79-11e2-93eb-000c29e29398} -> \Device\TrueCryptVolumeR mounted
2013-10-11 03:55:13 UTC+0000
File Object        \Device\TrueCryptVolumeR\${Directory} at 0x7c2f7070
File Object        \Device\TrueCryptVolumeR\${LogFile} at 0x7c39d750
File Object        \Device\TrueCryptVolumeR\${MftMirr} at 0x7c67cd40
File Object        \Device\TrueCryptVolumeR\${Mft} at 0x7cf05230
File Object        \Device\TrueCryptVolumeR\${Directory} at 0x7cf50330
```

### Volatility Links

[The Volatility Foundation](#)

[The Art of Memory Forensics](#)

[Memory Analysis Site](#)

[2018 Plugin & Analysis Contests](#)

[2017 Plugin Contest](#)

[2016 Plugin Contest](#)

[2015 Plugin Contest](#)

[2014 Plugin Contest](#)

[2013 Plugin Contest](#)

[Code Repository](#)

### Volatility on Twitter

#### Tweets by @volatility

volatility Retweeted



**Volatility**  
@Volatility

WHO: [@stevenadair](#)

WHAT: Digital Surveillance and Cyberespionage at Scale - a look at [#OceanLotus](#)'s ongoing operations including new unreported research

WHEN: Weds, Mar 6 @ 1:30PM

WHERE: [#RSAC](#) | Session Code HT-  
[W10rsaconference.com/events/us19](#)  
[/ag...#DFIR](#) [#infosec](#) [#threatintel](#)

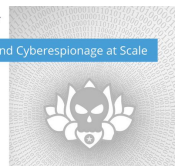
**VOLEXITY**

Digital Surveillance and Cyberespionage at Scale

6 Mar 2019 | 1:30 PM  
RSAC USA 2019



Steven Adair  
President, Volexity



Mar 5, 2019

volatility Retweeted

[Embed](#)

[View on Twitter](#)

### Blog Archive

- [2018](#) (4)
- [2017](#) (3)
- [2016](#) (9)
- [2015](#) (12)
- ▼ [2014](#) (27)
  - [December](#) (1)

```
File Object      \Device\TrueCryptVolumeR\${BitMap at 0x7cfa7a00
File Object      \Device\TrueCryptVolumeR\Chats\Logs\bertha.xml at 0x7cdf4a00
Driver           \Driver\truecrypt at 0x7c9c0530 range 0xffffffff88009200000 - 0xffffffff88009241000
Device           TrueCryptVolumeR at 0xfffffa801b4be080 type FILE_DEVICE_DISK
Container        Path: \Device\Harddisk1\Partition1
Device           TrueCrypt at 0xfffffa801ae3f500 type FILE_DEVICE_UNKNOWN
```

Among other things, you can see that the TrueCrypt volume was mounted on the suspect system on October 11th 2013. Furthermore, the path to the container is \Device\Harddisk1\Partition1, because in this case, the container was an entire partition (a USB thumb drive). If we were dealing with a file-based container as previously mentioned, the output would show the full path on disk to the file.

Perhaps even more exciting than all that is the fact that, despite the partition being fully encrypted, once its mounted, any files accessed on the volume become cached by the [Windows Cache Manager](#) per normal -- which means the [dumpfiles](#) plugin can help you recover them in plain text. Yes, this includes the \$Mft, \$MftMirr, \$Directory, and other NTFS meta-data files, which are decrypted immediately when mounting the volume. In fact, even if values that lead us to the master keys are swapped to disk, or if TrueCrypt (or other disk encryption suites like PGP or BitLocker) begin using algorithms without predictable/detectable keys, you can still recover all or part of any files accessed while the volume was mounted based on the fact that the Windows OS itself will cache the file contents (remember, the encryption is transparent to the OS, so it caches files from encrypted volumes in the same way as it always does).

After running a plugin such as truecryptsummary, you should have no doubts as to whether TrueCrypt was installed and in use, and which files or partitions are your targets. You can then run the truecryptmaster plugin which performs nothing short of magic.

```
$ python vol.py -f WIN-QBTA4.raw --profile=Win2012SP0x64 truecryptmaster -D .
Volatility Foundation Volatility Framework 2.3.1 (T)
```

```
Container: \Device\Harddisk1\Partition1
Hidden Volume: No
Read Only: No
Disk Length: 7743733760 (bytes)
Host Length: 7743995904 (bytes)
Encryption Algorithm: SERPENT
Mode: XTS
Master Key
0xfffffa8018eb71a8 bbe1dc7a8e87e9f1f7eef37e6bb30a25 ...z.....~k..%
0xfffffa8018eb71b8 90b8948fefe425e5105054e3258b1a7 .....B^Q..N2X..
0xfffffa8018eb71c8 a76c5e96d67892335008a8c60d09fb69 ..1^..x.3P.....i
0xfffffa8018eb71d8 efb0b5fc759d44ec8c057fbc94ec3cc9 .....u.D.....<.
Dumped 64 bytes to ./0xfffffa8018eb71a8_master.key
```

You now have a 512-byte Serpent master key, which you can use to decrypt the roughly 8 GB USB drive. It tells you the encryption mode that the suspect used, the full path to the file or container, and some additional properties such as whether the volume is read-only or hidden. As you may suspect, the plugin works regardless of the encryption algorithm, mode, key length, and various other factors which may complicate the procedure of finding keys. This is because it doesn't rely on the key or key schedule patterns -- it finds them in the exact same way the TrueCrypt driver itself finds the keys in RAM before it needs to encrypt or decrypt a block of data.

The truecryptsummary plugin supports all versions of TrueCrypt since 3.1a (released 2005) and truecryptmaster supports 6.3a (2009) and later. In one of the more exciting [hands-on labs in our memory forensics training class](#), students experiment with these plugins and learn how to make suspects wish there was no such thing as Volatility.

**UPDATE 1/15/2014:** In our opinion, what's described here is not a vulnerability in TrueCrypt (that was the reason we linked to their FAQ in the first sentence). We don't intend to cause mass paranoia or discourage readers from using the TrueCrypt software. Our best advice to people seeking to keep data secure and private is to read the TrueCrypt documentation carefully, so you're aware of the risks. As stated in the comments to this post, powering your computer off is probably the best way to clear the master keys from RAM. However, you don't always get that opportunity (the FBI doesn't call in advance before kicking in doors) and there's also the possibility of [cold boot attacks](#) even if you do shut down.

-Michael Ligh (@IMHLV2)

Posted by Michael Hale Ligh at [12:05 PM](#)

Labels: [forensics](#), [training](#), [truecrypt](#), [volatility](#), [win8](#)

## 16 comments:



**Waqar Afridi** January 15, 2014 at 5:36 AM

Very Nice tutorial, something I was looking for. The Plug-in mentioned in this tutorial "truecryptsummary" Will it be upload anytime soon? I looked for it and couldn't find it.

- [October](#) (3)
- [September](#) (5)
- [August](#) (6)
- [July](#) (2)
- [May](#) (1)
- [April](#) (2)
- [February](#) (2)
- ▼ [January](#) (5)

[Malware](#)

[Superlatives: Most Likely to Cry s/Wolf/Cr...](#)

[Comparing the Dexter and BlackPOS \(Target\) RAM Scr...](#)

[TrueCrypt Master Key Extraction And Volume Identif...](#)

[The Secret to 64-bit Windows 8 and 2012 Raw Memory...](#)

[The Art of Memory Forensics](#)

► [2013](#) (39)

► [2012](#) (34)

### Contributors

[AAaron Walters](#)

[Andrew Case](#)

[Dave Lassalle](#)

[Jamie Levy](#)

[Kristel Faris](#)

[Michael Hale Ligh](#)

[Nick](#)

### Blogroll

[JL's stuff](#)

[Enfuse Materials](#) - I'd like to thank everyone for coming to my talks at Enfuse 2016. As promised, here are the materials for the course. I ask that you don't redistribute t...  
*2 years ago*

[Memory Forensics](#)

[Bringing together the DFIR Industry and Academia at DFRWS 2017](#) - I am happy to announce that I have joined the 2017 DFRWS organizing committee. My role for this conference is to bring industry researchers and practitione...  
*2 years ago*

[MNIN Security Blog](#)

Reply

 **Tony Sharp** January 15, 2014 at 12:36 PM

What are you saying, that master keys can be hacked out of TrueCrypt? If so, which alternatives would you recommend?

Reply



**Michael Hale Ligh** January 15, 2014 at 12:43 PM

@TonySharp: Hmm, yes, that was the point, more or less...but we also were very explicit to point out that TrueCrypt discloses this openly (<http://www.truecrypt.org/docs/unencrypted-data-in-ram#Y445>), and products such as PGP and BitLocker are in the same boat.

Reply

Replies



**Tony Sharp** January 15, 2014 at 12:51 PM

Thank you for the information and link. And sorry for the case of TL;DR. I'm at work right now. So, essentially, shutting down the computer to clear the RAM solves this issue.



**Michael Hale Ligh** January 15, 2014 at 1:16 PM

Sure, no worries. shutting the computer down is a good idea to clear out your sensitive data, however you don't always get the opportunity to do that [before being apprehended by an agent who wants to seize your computer]. There's also the possibility of cold boot attacks (<https://citp.princeton.edu/research/memory/>) in those cases.

---

Reply



**Michael Hale Ligh** January 15, 2014 at 12:47 PM

@Waqar: The plugins are planned to be included in the next Volatility release (2.4). Keep an eye on the blog or code repository for an expected date.

Reply



**fgds** January 15, 2014 at 6:21 PM

what about keyfiles?

Reply

Replies



**Aether** January 16, 2014 at 5:15 AM

Key files are processed along with other tokens to form the master key - once you've loaded the key files in, the master key must remain in memory in order to facilitate the encryption / decryption processes. As such, it doesn't matter whether you're using a password, keyfile, or both - if an attacker can get read-only access to your RAM, they can extract the key.

Of course, this is more about forensics than live attacks. The most widely cited attack type is the "FBI boots down your door" attack, whereby your machine is running, has the TC volume mounted, and you don't have time to unmount and power off. Whilst this is unlikely to happen to most of us (I hope) there are other attacks that deserve consideration when thinking about coldboot.

---

Reply



**Quartz** January 16, 2014 at 5:31 AM

Does this have any application for the various ransomware malware that are doing the rounds?

Reply

Replies

**Michael Hale Ligh** January 22, 2014 at 10:49 AM

How to DoS Authenticode Signature Verification and Spoil Live Forensics with Echo  
- A while back I was looking into some internals of Microsoft's Authenticode and found a way to prevent signature verification by creating a specially named ...  
5 years ago



**Push the Red Button**

A couple ideas that went nowhere - I suspect a lot of people in academia end up having a lot of ideas and projects that went nowhere for any number of reasons - maybe there were insurmountable...  
4 months ago

**Volatility**

Ótima opção de advogado trabalhista Porto Alegre, recomendo,... - Ótima opção de advogado trabalhista Porto Alegre, recomendo, ajudaram-me muito e fui super bem atendido!  
1 year ago



Hmm, I'm not quite sure what ransomware malware has to do with this. Can you explain a bit more and then I can possible answer.



**Jigsy** June 7, 2014 at 4:01 PM

Although it's a bit of a late reply, I think what he means is that could a user get the key that. say, Cryptolocker uses to encrypt a users files?

Basically rendering the ransom for the key to decrypt everything moot.

---

[Reply](#)



**G.** January 22, 2014 at 7:23 AM

What about if you mounted the container/partition in say... a Virtual Machine?

[Reply](#)

[Replies](#)



**Michael Hale Ligh** January 22, 2014 at 11:02 AM

In that case, the master keys would be in the VM's memory. This is even less secure, because the VM's memory is fully contained in the host's memory (i.e. the key is in two places now). Furthermore, if the VM was suspended or snapshotted, its RAM is written to a file on the host's machine so that it can properly be resumed...in which case the master keys are written to the host's disk.

---

[Reply](#)



**Chris Taylor** February 27, 2014 at 4:03 PM

So just had a quick flick through, how would someone use the masterkey gain access to a full disk encrypted disk or truecrypt container. Would then need to decrypt the master key?

[Reply](#)

[Replies](#)



**security-geek** March 4, 2014 at 11:03 AM

You can decrypt master key only using your password or key-file. Or obtain UNENCRYPTED master key from memory dumps, RAM etc. as shown in this great article. It's safe enough to use full disk encryption where OS is. Article says more about TC volumes mounted under unencrypted OS - which in fact is obviously much less secure.

---

[Reply](#)



**security-geek** March 4, 2014 at 10:59 AM

As truecrypt site says - use full disk encryption for OS (and other disks too - optional). When there is some memory dump or page file it will remain encrypted on disk, so your master key, even when written to disk, is still safe.

Data stored in RAM is always unencrypted, so do your best to prevent from running forensic tools on your running OS.

Enable auto lock every 5 minutes of inactivity so you have to type in user password if your are away for longer time.

Work on user account with dropped down privileges (do not allow to run unrecognized software).

Disable your firewire ports in bios (to avoid on logon screen memory dump via 1334 attack)

Set memory overwrite at start in bios (its full POST with memory testing which basically wipes out your RAM when they restart your computer, or when you press reset).

Set BIOS both passwords - it will slow them down a bit, and every second is crucial performing cold boot attack. Also disable f11 boot menu device if possible.

When your computer is off for more than 5 minutes - your master keys in RAM will become unusable, unrecognized.

Privacy is not a crime - so use encryption anyway on every storage media. If you want to hide something for sure, use hidden OS.

[Reply](#)

Enter your comment...



Comment as:

4321yoja@gmail ▾

Sign out

Publish

Preview

☐ Notify me

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Powered by [Blogger](#).