

Re: Guidance Software response to iSEC report on EnCase (fwd)

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2007-07/msg00314.html>

- *From:* jf <jf@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 27 Jul 2007 05:03:18 +0000 (UTC)
-

they were able to identify six test scenarios, out of “tens of thousands” of test scenarios run

It only takes one. Plus this statement implies that its relatively hard to crash EnCase; anyone that used it on any regular basis knows that crashing EnCase is on par with kicking retarded children out of their wheelchairs. I will admit that I don't have any examples off-hand though—Cory Altheide, please stand up.

, that apparently revealed
minor bugs

By minor, you mean things like (1) where a disk image cannot be acquired or (2) that appears to cause an out-of-bounds memory operation or (3) which most likely has one hell of a race condition?

All of the testing involved
intentionally corrupted target data that highlighted a few relatively
minor bugs.

Yes, and pretty much every exploit on the planet involves intentionally corrupted data.

The issues raised do not identify errors affecting the integrity of the evidence collection or authentication process, or the EnCase Enterprise process (i.e., the operation of the servlet code or the operation of the SAFE server). Moreover, the issues raised have nothing to do with the security of the product.

Re: Guidance Software response to iSEC report on EnCase (fwd)

Yes, the md5 hash of the lack of an image is 0, so no data was corrupted (or acquired). Furthermore, are you will to step out and say that none of these bugs will get an examiners workstation owned?

But hey, inability to acquire an image and dangling pointers are not a security condition.

Therefore, we strongly dispute any media reports or commentary that imply that there are any “vulnerabilities” or “denials of service” exposed by this report.

Of course you do, I can't blame you or your company. But let's be serious here for a moment, wishing that you're the queen of England doesn't make it so.

Forensic examiners will inevitably come across corrupted data on target systems from time to time; and in standard computer forensics training, including classes offered by Guidance Software, examiners are trained to account for such issues. In addition, while Guidance Software maintains a robust in-house quality assurance process and strives to make our software as stable as possible, no software is completely crash-proof and there will always be anomalies, particularly involving extreme scenarios of corrupted target data.

Did you really just turn the shoddiness of your application into a training opportunity?

1. [Logical] Disk Image Cannot be Acquired With Certain Corrupted MBR Partition Table.

Response: It should be no surprise to any computer forensic examiner that a logical copy of a volume may not be possible if that volume has a corrupted MBR Partition table. EnCase features an option to acquire the target media physically, rather than logically, to specifically account for this type of scenario. The authors ignored the option of acquiring the data physically. Also, by corrupting the MBR Partition table, the perpetrator would likely render his computer inoperable, which calls into question both the likelihood and feasibility of such a tactic.

So if I give you a computer with a corrupted MBR partition table that still boots, what are you going to give me? You realize that the corrupted MBR partition table really doesn't break anything, it just requires that the boot code in the first sector of the disk to be rewritten. Do you actually understand how a PC boots?

Re: Guidance Software response to iSEC report on EnCase (fwd)

Here's a good idea. corrupt the partition table, store the real one at some distant offset, rewrite bytes 0–446 in the first sector to load the sector with the 'real' partition table, read it and boot correctly. Furthermore, I know some BIOSs will still boot without a valid MBR partition table in the first place.

2. Corrupted NTFS file system crashed EnCase during acquisition.

Response: The authors state that “this issue appears to be caused by an attempt to read past the end of the buffer.” However, EnCase features an option to de-select the automatic reading of the file system during the acquisition process. Thus, there is an easy work-around. Also, by corrupting the NTFS partitions, the perpetrator would likely render his file system dysfunctional, which calls into question both the likelihood and feasibility of such a tactic. Thus, the chances of this specific scenario occurring in the field are extremely remote; however, Guidance Software will test and, if verified, place this anomaly in its development queue to address the crashing problem in the future.

So really all I need to do is wrap my partition/file-system in a corrupted NTFS (btw NTFS file system is redundant), and poof I potentially owned the forensics workstation and with a little lucky just kept you from accessing my data.

Furthermore, okay, seeing as I control the disk, I potentially control whats on both sides of the buffer you read past, so I can potentially corrupt the integrity of the image you made, and thats even I can't grab control of eip through you're bad read.

3. Corrupted Microsoft Exchange database crashes EnCase during multi-threaded search/analysis concurrent to acquisition

Response: The report discloses that this particular anomaly occurred only when every single check box was selected in the search dialogue box, including the search, hash value calculation and verify file signatures features. This means that EnCase was directed to acquire an Exchange database and perform a detailed multi-threaded search and analysis of the data at the same time. This procedure is extremely inconsistent with best practices and akin to opening several hundred files in a word processing program, which of course would cause a memory overload.

So, you have options that you don't expect customers to select? If this is such a problem, why do you allow all of the options to be selected at the same time?

5. EnCase Had Difficulty Reading Intentionally Corrupted NTFS File System Directory.

Response: This issue involves the authors intentionally corrupting an NTFS file system to create a “loop” by, “replacing a directory entry for a file with a reference to the directory’s parent directory.”

Experienced forensic examiners are trained to identify such instances of data cloaking. The purposeful hiding of data by the subject of an investigation is in itself important evidence and there are many scenarios where intentional data cloaking provides incriminating evidence, even if the perpetrator is successful in cloaking the data itself. The chances of this specific scenario occurring in the field are extremely remote, but Guidance Software will test and, if verified, place this anomaly in its development queue to be addressed in the future.

Your argument is that they should be able to detect this manually? Isn't that why they bought your overpriced software in the first place?

why are the odds of this remote? Because you said so, or because of reasons like the corrupted MBR partition table?

So here's some truth, you're a forensics software company, one that deals by and large with LEO, which means you have an incredibly high bar to fill, if you're application is not pristine, all sorts of 'bad people with good lawyers (tm)' walk the streets, essentially invalidating the entire purpose of your tool. Stating that there are work-arounds or attempting to sweep bugs under the carpet just exasperates the situation.