Chapter 5

Incident Response: Live Forensics and Investigations

Solutions in this chapter:

- **Postmortem versus Live Forensics**
- **Today's Live Methods**
- **Case Study: Live versus Postmortem**
- **Computer Analysis for the Hacker Defender Program**
- **Network Analysis**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

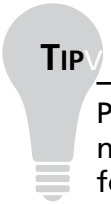
Introduction

To pull or not to pull the plug, that is the question. Today, cyber crime investigators are faced with the grueling task of deciding whether shutting down a computer system is the most efficient and effective method to gather potential electronic evidence. Traditionally, computer forensics experts agreed that shutting the computer system down in order to preserve evidence and eliminate the potential changing of information is best practice prior to examination. I remember having the phrases "shut it down," and "don't change anything" beaten into my brain during the numerous trainings I've attended throughout the years. However, one of the fundamental misconceptions with this philosophy is that computer forensics is the same as physical forensics. I would argue that they are not the same, given that computer forensics technology changes faster than traditional forensics disciplines like ballistics, serology, and fingerprint analysis. The second misconception is that we always collect everything at a physical crime scene. In a physical forensics environment, we commonly photograph the physical crime scene and take "reasonable" precautions to ensure the evidence is not disturbed. The truth is, in many cases, we only collect samples from a physical crime scene.

Nevertheless, we have accepted this methodology as best practice, and have backed ourselves into a litigation corner. The evolution of technology has put us face to face with the harsh reality that it is sometimes more advantageous to perform "Live" analysis than a "Postmortem" one. The problem is that live analysis often changes evidence by writing to the hard drive. File time stamps, Registry keys, swap files, and memory are just some of the items that can be affected when conducting analysis on a live computer system. Often, once the live analyst is done, the resulting MD5 hash will not match the hash collected prior to the live collection.

Postmortmem versus Live Forensics

Why should we even consider conducting live investigations as a valid forensic methodology? The reason is we have to! In the pages that follow, I will discuss the need to move away from traditional methods of computer forensics and toward a live forensics model.



TIP

Postmortem and live forensics are both great evidence gathering techniques. However, in cases where you can only conduct a postmortem forensics, the need to look at other systems within the environment is strengthened. This expansion of your scope to include other systems on the network will give you a better understanding of how the target system acted within its native environment.

Evolution of the Enterprise

Technology has evolved in such a way that conducting live investigations is really the only option you have under certain circumstances. In the days of old, computer networks were simple. In today's world, the evolution of the enterprise network work makes it difficult for system administrators, IT security personal, and the like to be at more than one location. Managing IT resources at a single site can be a daunting task. Now think of the larger corporate network schema. Many companies have multiple computers at a single location. Additionally, those corporations may also have several locations in a city, country, or continent. What would happen to our resources if we had to respond to every site and pull the computer off the network to conduct a forensic analysis for every suspected compliance issue, security breach, or compromised host? This would be even worse if after all the effort, time, and resources, we conclude that none of the aforementioned even occurred. Sound familiar? It should, because it happens every day in the cyber world. Triage is a common practice when diagnosing problems within a network. It is our first reaction, and we don't necessarily assume we are under attack, or that our systems have been compromised. In a live forensic environment, IT security personnel could log on remotely, view running processes, dump physical memory, and make an educated guess as to whether or not the computer should be imaged remotely, or be physically removed from the network for further analysis. In this scenario, the investigator, using live forensics techniques, doesn't have to physically respond to the location to address the issue until they are satisfied with their initial inquiry. This methodology will help conserve resources.

Evolution of Storage

Now back to pulling the pull. Once upon a time there was a server. This server was about 630 terabytes (TB) in size. It was responsible for handling the day-to-day operations of Company X, which traded stocks for its clients 24 hours a day. This server was believed to be compromised because of some unusual traffic detected within the log files of the firewall. This scenario presents us with the following issues. Problem 1: How are we going to fit this 630TB image into our 250GB USB2 external drive? Problem 2: How long would it take to image a drive that size? Problem 3: The machine cannot be shut down because the company would suffer a financial loss. In addition to all these issues, we must remember to make a bit-stream image, which was discussed earlier in Chapter 1. Let's discuss the preceding problems one at a time.

Problem 1: It's not possible. You will need a bigger drive.

Problem 2: The data resides on a substantially large server (630TB). Imaging the entire server is not practical, even though best practices dictate we should. Here is one of the reasons why: 630TB is equal to 6,926,923,254,988,880 bytes. 630 x 1,099,511,627,776 (1 Terabyte) = 6,926,923,254,988,880 bytes. See Table 5.1 to determine the byte sizes used in this scenario.

Table 5.1 Byte Conversion Chart

Drive Size	Numerical Representation	2 to the Following Power
1 kilobyte	1,024	10
1 megabyte	1,048,576	20
1 gigabyte	1,073,741,824	30
1 terabyte	1,099,511,627,776	40
1 petabyte	1,125,899,906,842,624	50
1 exabyte	1,152,921,504,606,840,000	60

Let's assume you use the ICS Image MASSter Solo-3 IT, which states it can duplicate hard drives at a rate of 3 GB a minute.

- Divide 6926923254988880 / 3221225472 (3 gigabytes) = 2150400 total minutes
- Divide 2150400 minutes /60 minutes (1 hour) = 35840 total hours
- Divide 35840 total hours / 24 hours (1 day) = 1493 total days
- Divide 1493 days / 365 day (1 year) = over 4 years to image the entire drive.

As you can see from the preceding bullets, imaging the entire one-to-one drive is not practical. Even if you imaged the data, by utilizing additional resources, the analysis of such a large volume could prove just as prohibitive. The difference in conducting an analysis on such a large volume, as compared to specific data objects and/or smaller storage systems, (using a detective's analogy) would be equivalent to interviewing every person who lives on a block where a homicide has occurred (reasonable), versus interviewing everyone who lives in the city of the homicide victim (not reasonable).

Notes from the Underground…

Using Compression

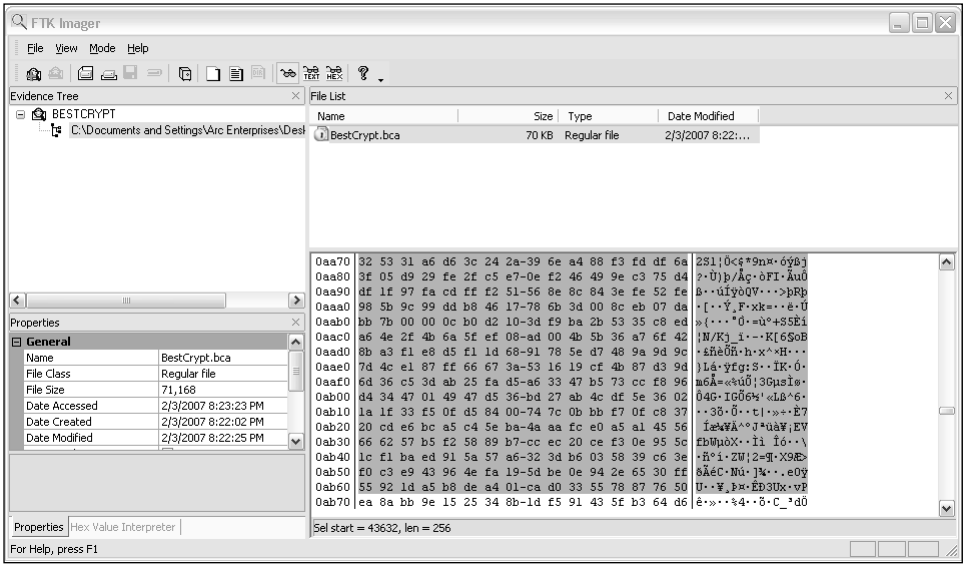
If you're thinking that the use of compression could solve the preceding problems, you would be mistaken. Compression increases the time it takes to image the server's hard drive because the compression algorithm needs to examine and remove the redundant items prior to compressing them. Additionally, it would still be impossible to compress the larger hard drive into the smaller USB external drive.

Problem 3: Shutting down the server is also not an option since the most obvious side effect would be the economic harm Company X would experience as a result. Many systems in existence today are mission critical, such as those supporting health care, transportation, and so on, and they couldn't be shut down without causing detrimental effects.

Encrypted File Systems

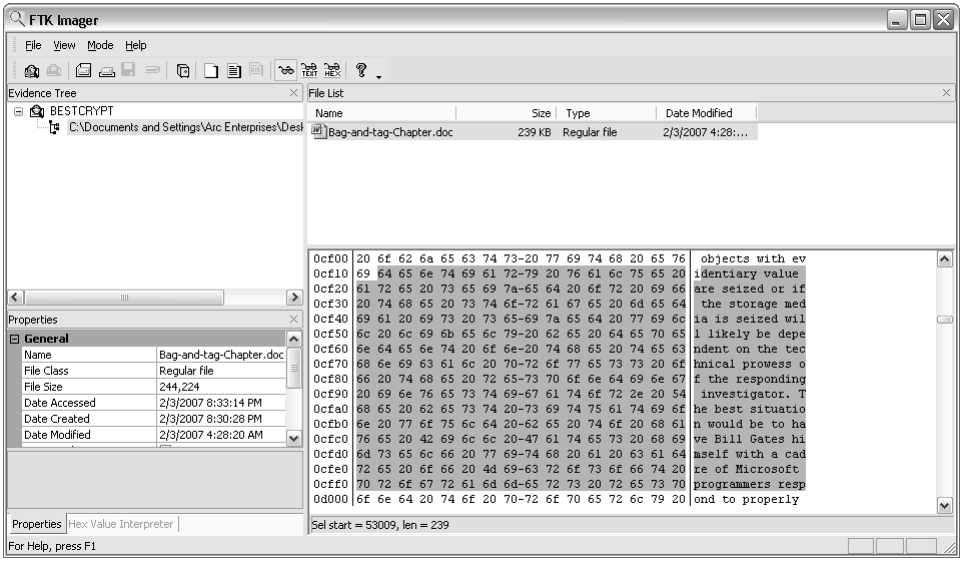
The use of encryption has increased during the last few years. Its increased use presents a unique problem to investigators when conducting postmortem analysis. When encryption is applied to a data object, the contents of that object are illegible. Encryption, by default, is designed to obfuscate, and sometimes compress, the contents of the data object it encrypts. Once encrypted, the object's contents are hidden and are pretty much impossible to interpret. Encryption is applied to these data objects in one of three ways. The first implementation is file level encryption, in which individual files are encrypted. Figure 5.1 shows the contents of an encrypted file.

Figure 5.1 File Contents When the File Is Encrypted Using AccessData's FTK Imager



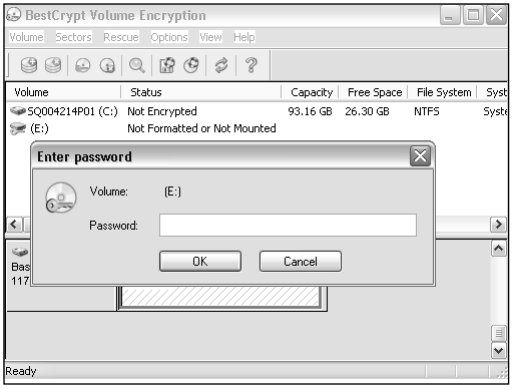
In order for an examiner to perform a postmortem analysis, he must first decrypt the file. Figure 5.2 shows a decrypted file. This could prove extremely difficult if the investigator does not have access to the encrypted file's password. No password may result in having to use a password cracking program. This decrypting process may prove useless if the password is too large, or the file is encrypted with a strong encryption algorithm and implementation.

Figure 5.2 File Contents When the File Is Not Encrypted Using AccessData's FTK Imager



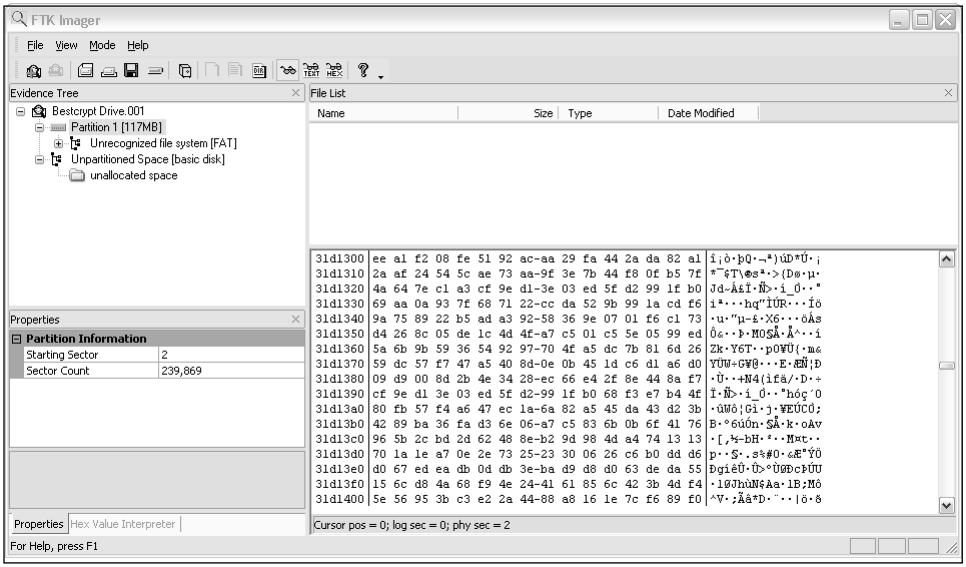
The second method used when applying encryption is volume level encryption. In this case, a volume within the hard disk is encrypted. Figure 5.3 shows an encrypted volume.

Figure 5.3 A BestCrypt Encrypted Volume



The third method used when encrypting a data object is whole disk encryption. This is when the entire hard drive is in encrypted. Figure 5.4 offers a forensic image of a fully encrypted disk. As you can see, its contents are illegible, and are of little value to a forensic examiner.

Figure 5.4 A Forensic Image of an Encrypted Hard Drive Using AccessData's FTK Imager



When conducting postmortem forensic analysis against the first two methods, investigators often hope to find artifacts of an encrypted file in its decrypted state that may be left in allocated or unallocated space. These artifacts are sometimes created once the document has been opened, or when the plug has been pulled while the file is still displayed on the screen. While this is a valid premise, recovery of these artifacts may not always be successful. Moreover, performing a proper shutdown may further decrease your chances of finding such evidence. In Figure 5.5, you will notice that the program BestCrypt offers to open the file in a temporary folder, and then securely delete the file when the program is closed.

When you use live forensics, the chances are significantly greater to view the contents of the encrypted file. If the document is open, it will most likely be loaded into physical memory. In a live forensic environment, the investigator could image the physical memory of the computer system and glean useful information about what files and programs the suspect may be currently using. So, before pulling the plug, it may be worth our while to examine the contents of the physical memory. Figure 5.6 shows one example of how we could image physical memory by using a network forensics tool.

Figure 5.5 A File-Cleaning Operation Offered by BestCrypt

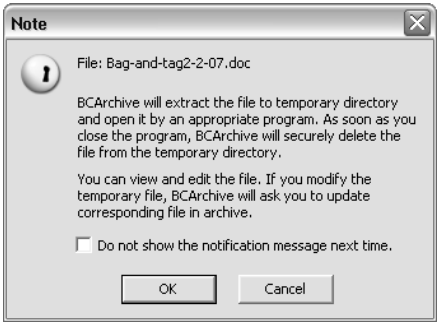
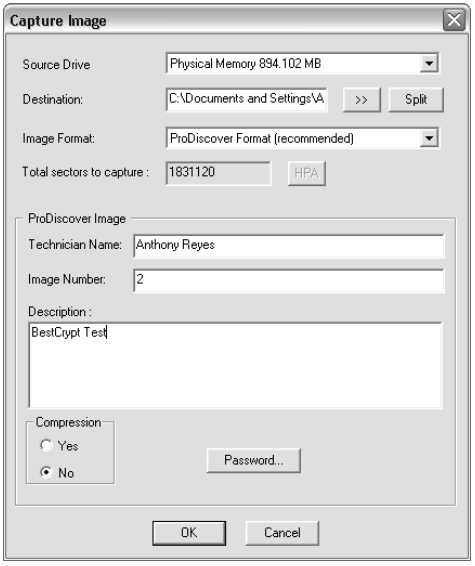


Figure 5.6 The Technologies Pathways' ProDiscover IR Imaging Screen



Once the image has been created, we can examine its contents. In Figure 5.7, you will notice the contents of the encrypted file are displayed in a readable format in the lower right-hand pane. Recovery of this information is because the file has been unencrypted by the user who is currently working with the document. Additionally, in Figure 5.8 you can see the BestCrypt program is running in physical memory. This information is also displayed in the lower right-hand pane.

Figure 5.7 An Unencrypted Document in Memory Using Technologies Pathways' ProDiscover IR

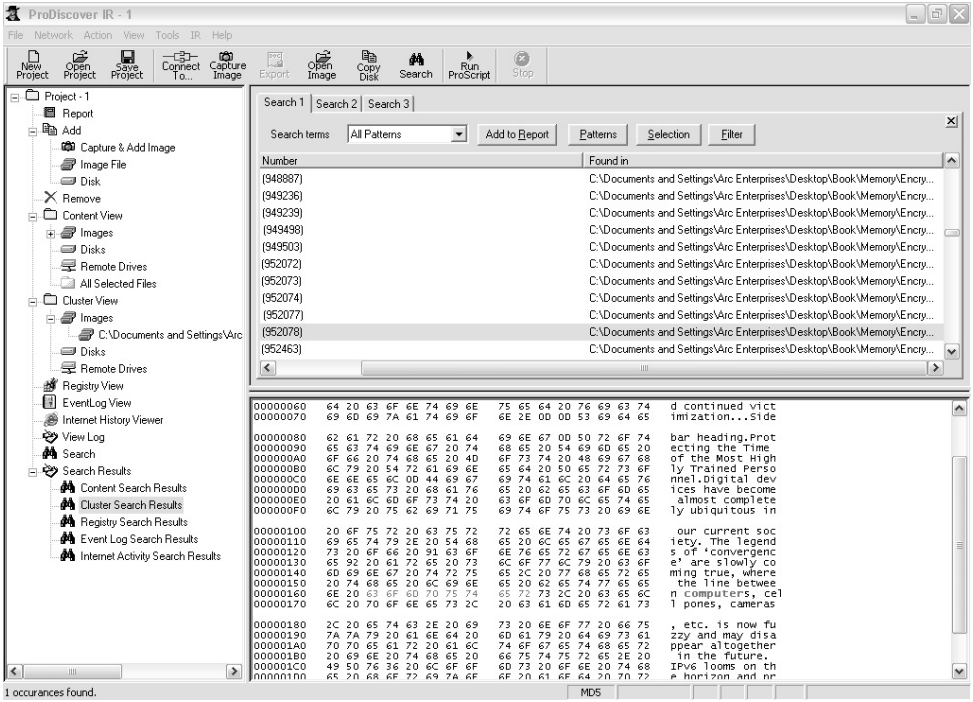
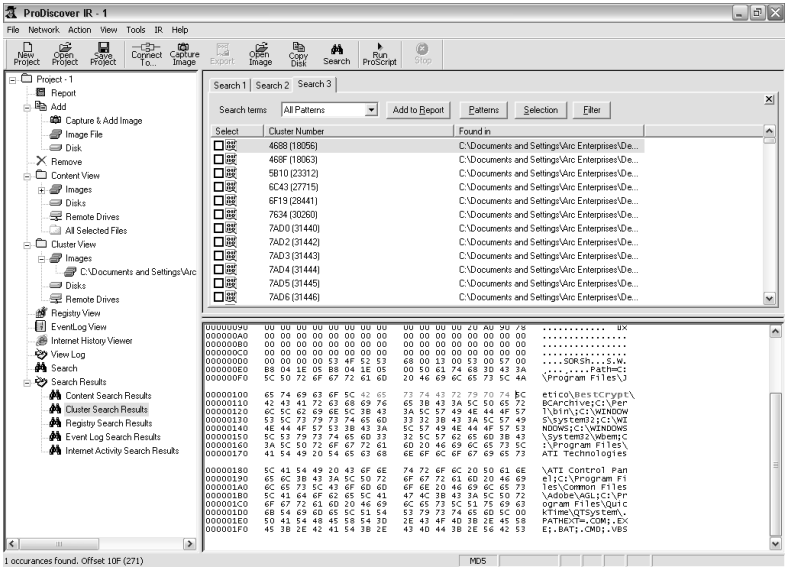
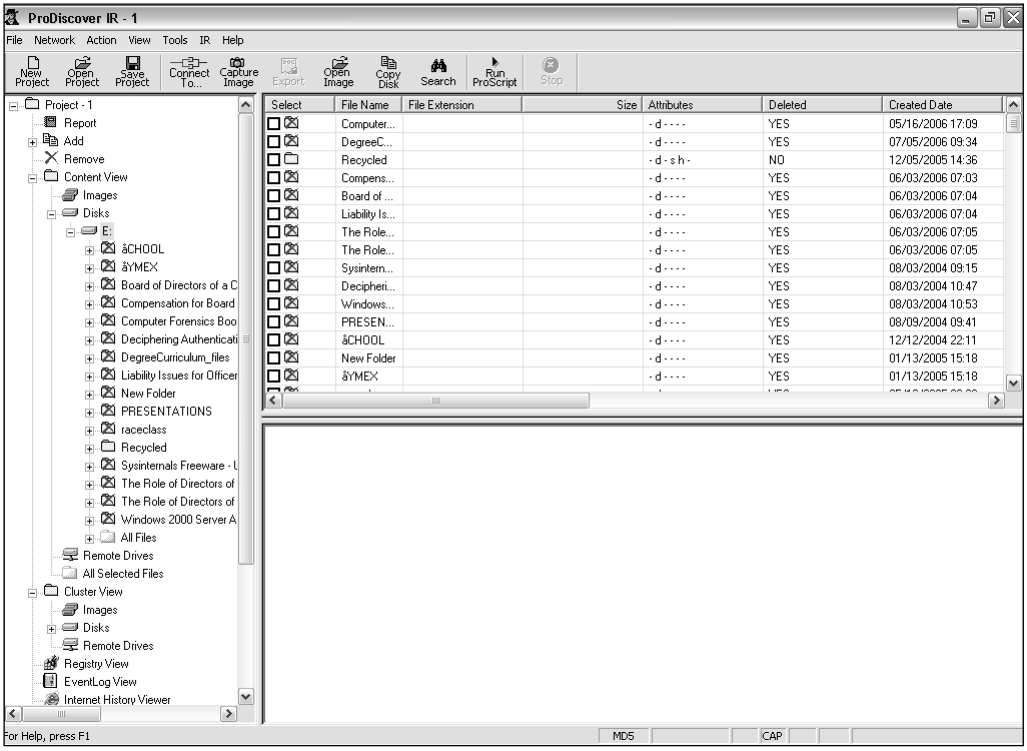


Figure 5.8 A View of Physical Memory Contents Using Technologies Pathways' ProDiscover IR. Note that the BestCrypt Process Is Running.



In the case of whole disk encryption, a forensic examiner using live forensics techniques would be able to view the content of the drive when it is mounted by the suspect. Simply put, because the drive is presently being used, it is unencrypted. Figure 5.9 demonstrates our ability to view the mounted drive's contents in its unencrypted state.

Figure 5.9 An Encrypted Hard Drive's Contents When Mounted Live with a Forensics Tool Like Technologies Pathways' ProDiscover



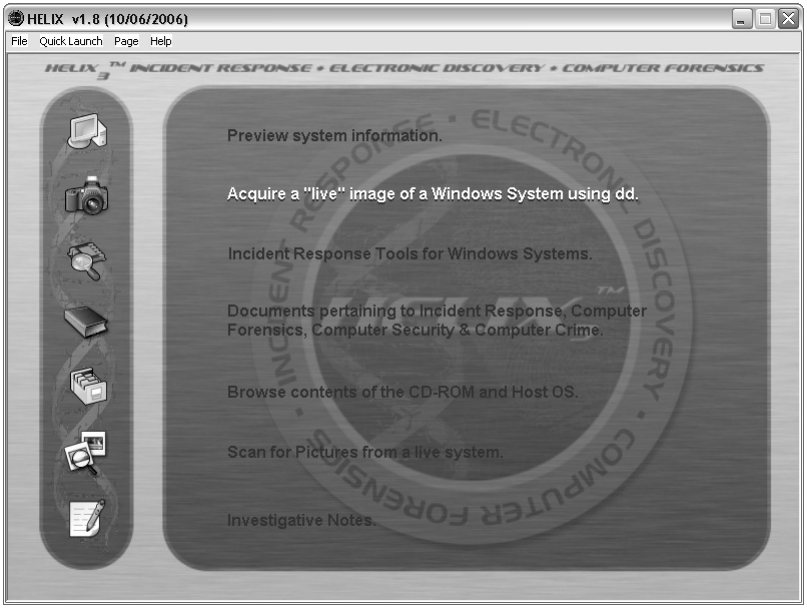
As you can see from the preceding examples, encryption presents a variety of problems for the traditional forensics examiner. With live investigative techniques, however, we can overcome these problems and obstacles.

Today's Live Methods

Several software companies presently manufacture network forensic and investigative software. Guidance Software, Technologies Pathways, Wetstone Technologies, ASR Data, E-fense, and E Trust by CA are just some of the

companies that produce this forensic and incident response software. These manufacturers use a variety of methods to conduct live investigations. The first method employed is the Pre-Deployed Agent model, where special software is pre-installed on a computer system prior to an incident. It is usually hidden from the end user and is invoked once it is connected to remotely. The second method currently in use is the Direct Connect model. In this model, the target computer is directly connected to by a remote machine and the software is pushed into memory. The connection remains active until the remote machine is disconnected. A third method is the On Demand Connection model, where the computer connects to the target machine and pushes the software into memory for a specific task. Once the task issued by the remote machine is completed, the connection is immediately torn down. Finally, some software developers use a boot disk or an investigative CD-ROM. During a live analysis, a disk is loaded to the live machine and a virtual session is initiated with a set of examination tools. Figure 5.10 shows a boot disk that allows you to conduct live forensics, as well as investigations.

Figure 5.10 The E-fense's HELIX Incident Response, Electronic Discovery, and Computer Forensics Boot Disk



Case Study: Live versus Postmortem

Live investigations allow investigators to capture volatile information that would not normally be present in a postmortem investigation. This information can consist of running processes, event logs, network information, registered drivers, and registered services. Why is this important to us, you ask? Let's take a look at the case of running services and how this could be extremely important us.

Running services tell us the types of services that may be running on a computer. These services run at a much higher priority than processes, and many users are unaware that these services actually exist. Given their high priority and lack of attention by the typical end user, they are a common target for hackers. By conducting a live investigation, we are able to see the state of these services, which could prove crucial to our investigation. For example, a hacker could turn off the service for McShield, which is a McAfee Antivirus service, and then later come back and infest the machine with malicious software.

You might argue in the case of registered drivers that you could get a list of the drivers in a postmortem investigation. This is true; however, if you are at a crime scene and you conduct a live investigation, you might be able to see a driver for a digital camera. So you know to look for that camera in your surrounding area. But if you left the location, and then returned later to find that camera driver, you could only hope that the camera is still there when you make it back. As shown in the previous example, seeing registered drivers gives investigators knowledge of the peripherals of a suspect machine. Figure 5.11 illustrates some of the volatile information you can obtain about a systems state.

Viewing running processes with the associated open network ports is one of the most important features of analyzing the system state. To peek into a system and correctly assess what processes are running and what ports they may be using is critical when trying to perform an investigative triage. Figure 5.12 offers a detailed look at the running processes of a target machine under investigation.

Figure 5.11 An Example of Live System Information You Can Obtain Using Wetstone's LiveWire

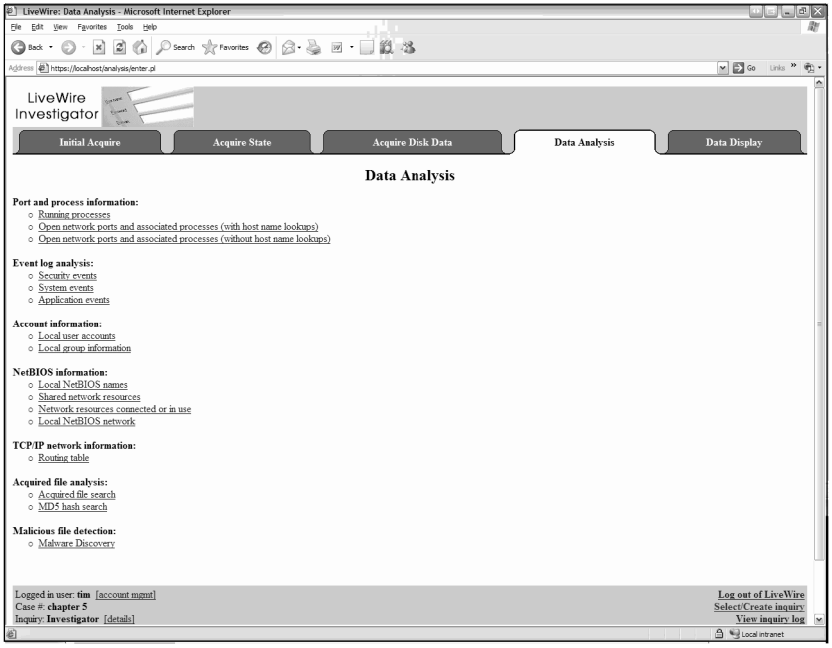
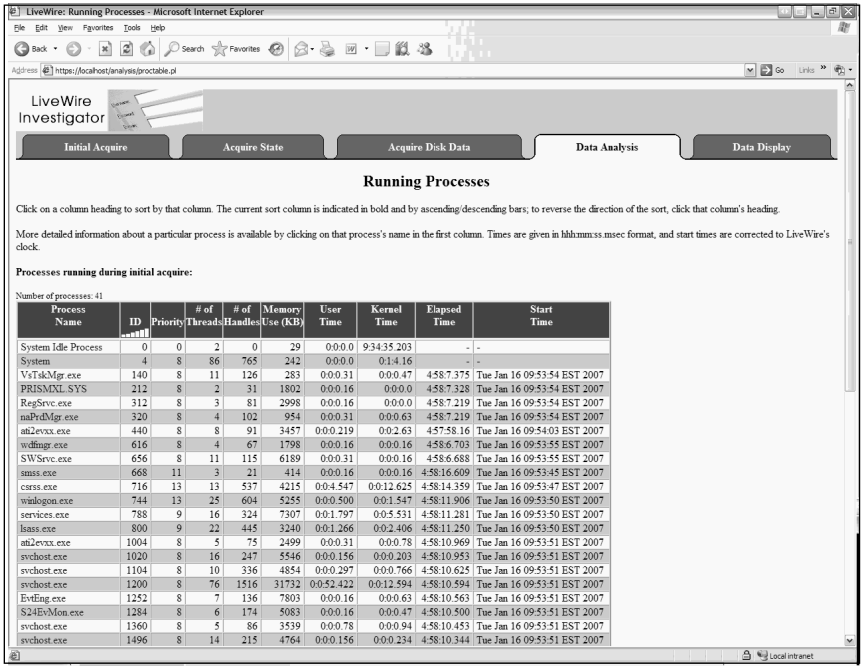


Figure 5.12 A View of Running Processes Using Wetstone's LiveWire

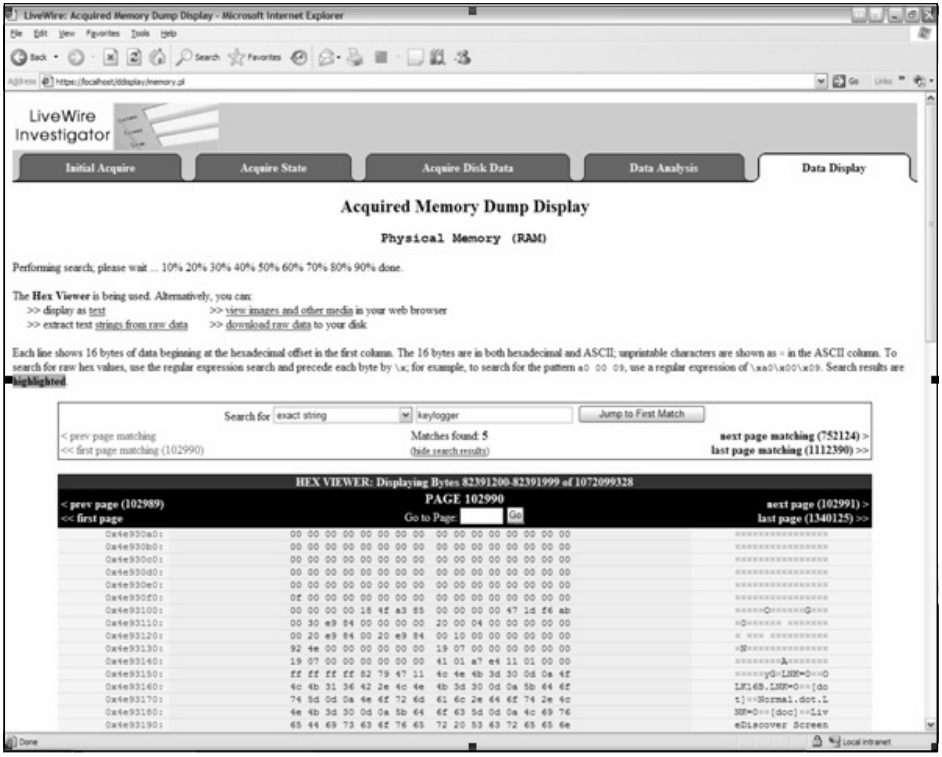


Process Name	ID	Priority	# of Threads	# of Handles	Memory Use (KB)	User Time	Kernel Time	Elapsed Time	Start Time
System Idle Process	0	0	2	0	29	0:0:0.0	9:34:35.203	-	-
System	4	8	86	765	242	0:0:0.0	0:1:4.16	-	-
VsTskMgr.exe	140	8	11	126	283	0:0:0.31	0:0:0.47	4:58:7.375	Tue Jan 16 09:53:54 EST 2007
PRISMXL.SYS	212	8	2	31	1802	0:0:0.16	0:0:0.0	4:58:7.328	Tue Jan 16 09:53:54 EST 2007
RegSrvc.exe	312	8	3	81	2998	0:0:0.16	0:0:0.0	4:58:7.219	Tue Jan 16 09:53:54 EST 2007
naPrdMgr.exe	320	8	4	102	954	0:0:0.31	0:0:0.63	4:58:7.219	Tue Jan 16 09:53:54 EST 2007
ati2evxx.exe	440	8	8	91	3457	0:0:0.219	0:0:2.63	4:57:58.16	Tue Jan 16 09:54:03 EST 2007
wdfmgr.exe	616	8	4	67	1798	0:0:0.16	0:0:0.16	4:58:6.703	Tue Jan 16 09:53:55 EST 2007
SWSrvc.exe	656	8	11	115	6189	0:0:0.31	0:0:0.16	4:58:6.688	Tue Jan 16 09:53:55 EST 2007
smss.exe	668	11	3	21	414	0:0:0.16	0:0:0.16	4:58:16.609	Tue Jan 16 09:53:45 EST 2007
csrss.exe	716	13	13	537	4215	0:0:4.547	0:0:12.625	4:58:14.359	Tue Jan 16 09:53:47 EST 2007
winlogon.exe	744	13	25	604	5255	0:0:0.500	0:0:1.547	4:58:11.906	Tue Jan 16 09:53:50 EST 2007
services.exe	788	9	16	324	7307	0:0:1.797	0:0:5.531	4:58:11.281	Tue Jan 16 09:53:50 EST 2007
lsass.exe	800	9	22	445	3240	0:0:1.266	0:0:2.406	4:58:11.250	Tue Jan 16 09:53:50 EST 2007
ati2evxx.exe	1004	8	5	75	2499	0:0:0.31	0:0:0.78	4:58:10.969	Tue Jan 16 09:53:51 EST 2007
svchost.exe	1020	8	16	247	5546	0:0:0.156	0:0:0.203	4:58:10.953	Tue Jan 16 09:53:51 EST 2007
svchost.exe	1104	8	10	336	4854	0:0:0.297	0:0:0.766	4:58:10.625	Tue Jan 16 09:53:51 EST 2007
svchost.exe	1200	8	76	1516	31732	0:0:52.422	0:0:12.594	4:58:10.594	Tue Jan 16 09:53:51 EST 2007
EvtEng.exe	1252	8	7	136	7803	0:0:0.16	0:0:0.63	4:58:10.563	Tue Jan 16 09:53:51 EST 2007
S24EvMon.exe	1284	8	6	174	5083	0:0:0.16	0:0:0.47	4:58:10.500	Tue Jan 16 09:53:51 EST 2007
svchost.exe	1360	8	5	86	3539	0:0:0.78	0:0:0.94	4:58:10.453	Tue Jan 16 09:53:51 EST 2007
svchost.exe	1496	8	14	215	4764	0:0:0.156	0:0:0.234	4:58:10.344	Tue Jan 16 09:53:51 EST 2007

Notice how we can see not only the process's name in Figure 5.12 but also the priority, the number of threads, number of handles, memory usage, and uptime. Again, you might ask why all of this is important. Well, if you are trying to assess what someone is currently doing, or even what they have done in the past, this information is critical. In addition, in the world of memory resident executables, analyzing the current process list is vital.

In a postmortem investigation, physical memory (RAM) is potentially the most important piece of evidence that is lost. However, this crucial piece of evidence is easily captured using live forensic and investigative tools, allowing the entire contents of RAM to be captured locally and even remotely. In Figure 5.13, we can see the contents of a memory dump and can conduct a search for the word keylogger in memory.

Figure 5.13 A Keyword Search for the Term Keylogger in a Memory Dump Using Wetstone's LiveWire



The raw data contents of the memory provide a vast amount of information that could have been lost if the machine was powered down for a postmortem investigation. Memory contains evidence ranging from user accounts, passwords, unsaved document content, and malicious software.

Terminology Alert…

Malicious Software

Malicious software is a term describing a broad range of tools. However, memory-resident malicious software generally is seen with rootkits, Trojan horses, worms, and keyloggers. The following example contains a detailed explanation on how some memory-resident malicious software work.

Computer Analysis for the Hacker Defender Program

Hacker Defender is a popular rootkit that is capable of hiding processes, files, and even open ports. By default, when Hacker Defender is executed, it hides every file containing the prefix "hxdef." As a result, the file "hxdef100.ini," which is part of Hacker Defender, is hidden as soon as Hacker Defender executes. This file is then hidden from all users and even Windows Explorer itself. However, the file still exists in physical memory. Using live investigation techniques, you can take a memory snapshot and identify the file "hxdef100.ini" stored in RAM (see Figure 5.14). This same method can be used to reveal any file or process that Hacker Defender hides (see Figure 5.15). During a postmortem investigation, any files or processes hidden by Hacker Defender may not be accessible to the investigator. Figures 5.14 and 5.15 show evidence of the Hacker Defender program in the physical memory of a computer.

Figure 5.14 Hacker Defender in Psychical Memory Using Wetstone's LiveWire

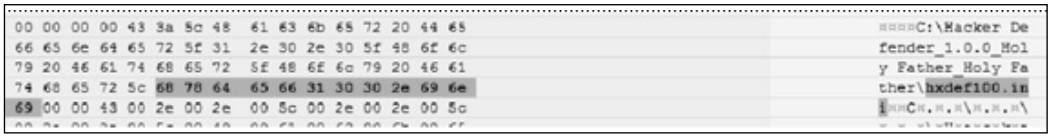


Figure 5.15 Another View of Hacker Defender in Psychical Memory Using Wetstone's LiveWire

As stated earlier, investigating a computer's system state is an important part of any investigation. It could help glean valuable information in a case and reduce the risk of missing data that could prove critical to your investigation.

Network Analysis

Often overlooked in live investigations is the environment in which the target computer resides. Data obtained from firewall laws, routers, intrusion detection systems, and so on are equally important to an examiner in obtaining the big picture. In the Hacker Defender case presented earlier, a defense attorney may argue that his client's machine was compromised and could not have committed the crime. A review of the firewall logs may show that the Hacker Defender activity from this computer was blocked, making this argument about the rootkit a moot point. As a live investigator, you should try to gain as much information about the network activity as possible. You might want to install a packet sniffer—with the appropriate permission, of course—and conduct a packet analysis of the traffic. Using this technique, you could determine if someone is connected to the box before conducting an analysis on the target machine. So remember, you may find additional evidence beyond the computer you are examining. Look for it.

Summary

As we move forward, computer forensics as we now know it will change dramatically. The release of Microsoft's Vista will enable users to fully encrypt their hard drives. The use of virtual machines and virtual server farms are becoming more commonplace. Internet-based application servers will be harder for forensic examiners to physically collect. Additionally, Internet-based applications may generate diskless workstations, leaving the only evidence in physical memory. Finally, software vendors are starting to deploy a larger amount of software that securely deletes data because of identity-theft concerns. Because of these changes, and as I have pointed out in the examples in this chapter, I surmise that traditional forensics will become more impractical, and live investigations will become a necessity rather than a luxury. Traditional methodologies are becoming somewhat obsolete. The need to adopt a new way of conducting these types of investigations is essential. While we have shied away from touching the computer in order to prevent any changes, it is now obvious that there are times when an examiner must interact with a live computer in order to retrieve vital data. Under the circumstances described earlier, you should be able to provide a reasonable explanation to any judge or jury as to why live forensics was used in place of traditional methods. However, should none of these circumstances exist, it may be best just to pull the plug.

Special Thanks

I would like to give thanks to my colleagues Christopher L. T. Brown and Chet Hosmer for their help with this chapter. Their wisdom and insight into incident response and network forensic issues were invaluable.

References

Brown, Christopher L.T. *Computer Evidence Collection & Preservation*. Massachusetts: Charles River Media, Inc., 2006.

Chirillo, John. *Hack Attacks Revealed*. New York, John Wiley & Sons, Inc., 2001.

Mandia, Kevin et al. *Incident Response: Investigating Computer Crime.* California: Osborne/McGraw-Hill, 2001.

McClure, Stuart *et al. Hacking Exposed: Network Security Secrets & Solutions.* California: Osborne/McGraw-Hill, 2001.

Szor, Peter. *The Art of Computer Virus Research and Defense.* New Jersey: Addison-Wesley, 2005.

Solutions Fast Track

Postmortem versus Live Forensics

- ☑ In a live investigation, a system administrator can conduct an analysis remotely.
- ☑ Imaging large volumes can be a daunting task.
- ☑ Live forensics can be used to obtain data when encryption is in use.
- ☑ Capturing the contents of memory may provide you with the "missing link."

Today's Live Methods

- ☑ A Pre-Deployed Agent is software that is installed onto the computer prior to an incident.
- ☑ A boot disk can be used to contact live investigations.

Case Study: Live versus Postmortem

- ☑ Live investigations allow investigators to capture volatile information that would not normally be present in a postmortem investigation. This information can consist of running processes, event logs, network information, registered drivers, and registered services.
- ☑ Running services tell us the types of services that may be running on a computer. These services run at a much higher priority than

processes, and many users are unaware that these services actually exist.

- ☑ Viewing running processes with the associated open network ports is one of the most important features of analyzing the system state. To peek into a system and correctly assess what processes are running and what ports they may be using is critical when trying to perform an investigative triage.

Computer Analysis for the Hacker Defender Program

- ☑ Hacker Defender hides files from the user.
- ☑ Rootkit artifacts can sometimes be found in physical memory.

Network Analysis

- ☑ You should look for evidence beyond the target computer.
- ☑ Understanding the network where the system resides can help you when conducting a live investigation.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Can I view encrypted data in a live environment without having the password?

A: The answer is yes, provided that the drive or file is unencrypted on the suspect's machine.

Q: Can I view hidden processes like rootkits on a live computer?

A: Using special software, you can view hidden processes and files on a live computer.

Q: If I cannot image the entire drive, can I just copy the files I need?

A: Yes, you can copy the files you need using live forensic software to ensure you have the entire copy. Also, take notes when doing this since you may have to testify later about why you chose this method and what, if anything, you changed.