Chapter 2

The Complex World of Corporate CyberForensics Investigations

Gregory Leibolt

Gregory Leibolt, GCFA, GCIH, GREM, GPEN, C|EH, GCCF, CISSP is a Senior Technical Director at AT&T where he leads both the Digital Forensics Team and the Ethical Hacking Team. Leibolt’s extensive experience in incident response and digital forensics includes large and small cases involving intrusions, intellectual property theft, financial embezzlement, investigating employee misuse of assets, malware infestations, and others. He has spent considerable effort developing tools and techniques in support of the Linux and UNIX environments. He regularly evaluates new forensics tools and techniques to find the best solutions for this challenging field. On the ethical hacking side, Leibolt’s penetration testing experience encompasses ethically exploiting websites, databases, computer systems, internal LANs, perimeter defenses, and wireless networks. He has evaluated and tested numerous network components, various critical architectures such as Communications Assistance for Law Enforcement Act (CALEA) implementations and many other unique environments. Leibolt’s extensive IT experience also includes UNIX and Windows system administration, software development, project management, evaluating security tools, and advanced security research. Leibolt holds multiple GIAC certifications, including Certified Forensics Analyst (GCFA), and is a member of the GIAC Advisory Board. His other certifications include Certified Incident Handler (GCIH), Reverse Engineer Malware (GREM), Certified Penetration Tester (GPEN), Certified Ethical Hacker (C|EH), and Certified Information System Security Professional (CISSP). Leibolt has a BA from Barry University, an MA in Computer Resources and Information Management from Webster University, and a Graduate Certificate in Computer Forensics (GCCF) from the University of Central Florida.

2.1 Investigation Characteristics

This chapter addresses key points of corporate cyberforensics that are required to explain the many challenges investigators face within the corporate environment. The information technology (IT) environment of a large corporation can be daunting to characterize. Large corporations are often composed of a mixture of IT components ranging from legacy production systems to bleeding-edge laboratory systems. Additionally, business acquisitions, partnering, and nonstandard IT implementations usually bring additional variety to the mix. As a result, many corporations go for many years without ever knowing the full extent of their corporation’s networks and systems. Basically, it is safe to say that, from a cyberforensics perspective, just about any type of IT environment can be encountered.

A more granular view would show that a wide variety of computer operating systems and hardware is usually deployed. The ever-popular Microsoft® Windows operating system permeates most corporate IT environments. A corporation that is predominately a Windows shop can make life easier for investigators because the forensic skill sets are similar across the different Windows platforms. In large companies one can expect to see virtually every type and version of
Windows desktop and server operating systems. In the
author’s corporation, similar to many others, there is
a mixed bag of operating systems (OS) to address.
The investigators routinely run into Windows, Linux,
Solaris, HP-UX, and AIX, plus different versions of
each OS.

Not only are there different operating systems, but
also they are deployed in a wide array of implementa-
tions. Some are running relatively small, self-contained
servers and desktops that use native file system types.
Others use large RAID, SAN, or NAS storage or
employ a mixture of various file systems. The envi-
ronment might also be heavily integrated with many
support systems, such as database, LDAP, or other
component servers.

Administration also comes in a variety of flavors.
Systems could be accessed by either direct or virtual
consoles. A single system could have one or more
administrators. Authentication could be local or via
domain administrative access. Two-factor authentica-
tion that is based on “something you know” such as
a password and “something you have” such as an
RSA SecurID® token might be required. What this
means is that investigators can often find it difficult to
gain administrative access because of technical issues,
complex paperwork, hard-to-locate administrators, or
complex access control lists (ACL). To perform inci-
dent response or remote forensics, computers often
need to be accessed via the network, but sometimes this
is a problem for reasons of network routing, ACLs, or
firewalls. Even the environment containing the system
of forensic interest is an important factor. Production,
test, laboratory, and desktop environments are vastly
different from each other and greatly affect the forensic
approach.

The discussion so far has centered on the differ-
et types of computers and operating systems. There
are also many elements supporting a network such as
routers, switches, firewalls, and intrusion detection
systems that are likely to be part of the investiga-
tion. Computers and network equipment can be located
in very diverse physical locations. VLANs can sup-
port multiple servers on one subnet, even though the
actual hardware is distributed in different buildings
miles from each other. Last but not least, investiga-
tors need to have some understanding of the vari-
ous applications that could be related to a compro-
mise. Aside from the technical challenges, there are
legal, political, financial, geographical, cultural, time
zone, resource, and skill set issues. In a word, cyber-
forensics in large corporate IT environments can be
complex.

The most important aspect of successfully man-
gaging cyberforensics in a corporate environment is
cooperation. Upper management must support a desig-
nated corporate forensics team working all such cases,
and the corporation as a whole must be aware of
the proper policies and procedures relating to inci-
dent response and digital forensics. Take, for example,
intrusion cases, which have life cycles with the follow-
ing phases: preparation, identification, containment,
eradication, recovery, and follow-up. The complexity
of the corporate IT environments means that many par-
ties must work together to handle cases such as these.
For example, legal departments, application owners,
application developers, system administrators, network
engineers, firewall administrators, and many others
must cooperate. Early communications between mem-
bers of this conglomerate team would be based around
topics such as understanding the details of the incident,
analyzing business risk, defining and providing appro-
priate access methods and controls to the investigators,
coordination of legal matters with corporate attorneys,
and reporting to upper management and government
entities as needed. The lead investigator must provide
the leadership to orchestrate the multi-team procedures
and activities to ensure the integrity and preservation
of the evidence. In addition, each case must be managed
in specific ways to ensure that evidence is protected,
properly collected, and correctly transferred, which
includes documenting the chain of custody. All parties
handling any evidence should take detailed notes of all
activities involving that evidence. Clearly, cyberforen-
sics professionals need a broad range of skills, both
technical and managerial, especially when handling
large corporate incidents.

2.2 The Investigative Approach

Cyberforensics investigators appreciate the simple
cases. An example might be a straightforward desk-
top computer system that, because of the nature of the
case, can simply be unplugged and taken to the lab.
These cases provide the opportunity to follow the tra-
ditional forensics procedures that can be performed on
systems that do not require live forensics: they only require imaging the drive and analyzing a static system. However, times are changing, and in the corporate world even the simplest of cases can be complex. Many user computer systems, particularly laptops, are now using full disk encryption, which requires application of appropriate decryption methods, before acquiring the necessary data. Hard drive sizes are growing in capacity each year, which also adds time to the acquisition and analysis of data. Access to user computer systems is often difficult because corporate users can be widely dispersed in remote offices or they may telecommute from their home offices. Older traditional methods are constantly being supplemented with tools and techniques that expand investigators’ capabilities and help to work around many of today’s complexities.

There is a growing shift in the corporate environment toward performing cyberforensics remotely. For example, the author’s corporation frequently uses this approach. It has proven to be a great asset in performing cyberforensics and in the e-Discovery of user systems. It also circumvents the need to decrypt encrypted hard drives and provides easy access to systems. Four often-mentioned forensics software vendors—Access Data®, Guidance Software (R), MANDIANT, and Technology Pathways—provide tools that enable remote forensics. Their software has the ability to connect to a small applet on a target computer system and perform the forensic acquisition and analysis of data. The old standard forensics practice of unplugging a system and analyzing a static image is becoming less and less common as new tools and practices are emerging to support evolving technology.

Investigators will tell you that each case is different. Even though standard policies and guidelines may be in place, the investigative approach taken in each case can vary. The approach is primarily determined by the characteristics of the incident in relation to the affected environment. In other words, the type of incident (for example, an insider threat, a hacking incident, a malware infestation) is considered when the investigators think about the best way to identify, preserve, and acquire potential evidence. Naturally, many approaches are considered. A sniffer on the network might be important, capturing system memory may be critical, disconnecting from the network might be important, or monitoring activity on the system may be the best approach. In addition to the type of incident, knowledge of the attacked environment (for example, an external website, a customer database server, an employee desktop, a lab test system) is critical and may severely limit the ability to implement some of the approaches originally considered. Finally, in addition to the incident type and the environment, the final approach must take into account issues such as physical access to the system(s), network access to the system(s), login access, skill sets, software tools, and political, legal, and business/network risk factors.

The union of all these many factors ultimately determines which approach is taken by the investigators. Because of these many complexities, the ultimate goals of an investigation sometimes lean toward identifying, containing, and remediating the problem rather than obtaining useful evidence for legal action. As such, investigators can find themselves in situations in which they can only apply “best forensics practices” in certain areas, but not all; these are usually cases where the company is not likely to go to court. Regardless of the type of case, the investigators should be sure to document why certain investigative steps took place. This ensures that all the actions can be explained should the need arise.

To provide multiple examples of investigative approach, the following discussion raises issues relating to a theoretical intrusion incident of a complex environment, such as a web portal. Whether the investigator is dealing with a small or large case, the investigative approach always starts with obtaining information. Especially in larger cases, it is not unusual to go back and ask for the same information several times. The first time that investigators receive information, it is often incomplete, and it can be inaccurate. This problem happens for many different reasons. One reason is that it is human nature for people to think that their network or system has not really been compromised. Investigators are often told that the problem is probably something a system administrator did, or an application programming issue, or maybe just a minor script kiddie attack. Additionally, people may be afraid that they could be in trouble and are not likely to provide any self-incriminating information. Another reason is that the needed information is not always at someone’s fingertips. It may take some time for people to find the most recent network diagram or application description document. Or, it may not exist. And, of course, key people who must be consulted may not be available. It is necessary for investigators to start the
investigation with what they have, but keep going back for more accurate and detailed information.

Using all the information related to the incident, investigators must try to identify, preserve, and acquire all possible sources of potential evidence. The following steps are important:

- Ensure that system administrators do not start their own investigation. Many times, administrators have accidentally corrupted evidence themselves or, in looking for evidence, were noticed by the hackers, who then deleted log files and their hacker tools before logging off the system.
- Ask the network architects for the network diagrams and descriptions of the affected network.
- Identify critical business components related to the affected network (for example, financial, contractual, legal).
- Determine if disaster recovery plans exist and can permit certain systems or even the whole portal to be restored in a clean state without interfering with the ability for the investigators to complete forensic analysis of the affected systems.
- Ask the security organization for any recent security audit reports of the affected network, systems, and applications.
- Establish the initial goals of the investigation. Goals are not always the same; however, they would generally include these:
  - Identify and preserve evidence.
  - Determine, as accurately as possible, the method, time frame, and the scope of the compromise.
  - Provide feedback about containment, remediation, and security enhancement.
  - Perform the investigation with as little disruption to the corporation as possible.
  - Identify, immediately, the firewall logs, centralized syslogs, router logs, IDS/IPS logs, sniffer data, application logs, and any other data that could help in the investigation.\(^7\)
  - Implement, immediately, a plan for the preservation of logs so that historical evidence is not deleted.
  - Identify, immediately, the available data storage to handle copies of all the logs. Note that this is often a challenge because corporations do not always have available and accessible storage handy. Sometimes, the easy answer is to ship out large-capacity USB external hard drives.
  - Implement proper controls: that is, large corporations are likely to have different teams and organizations related to different areas of potential evidence. Knowledge of cyber incidents in corporations is usually on a “need-to-know” basis. The appropriate controls must be implemented as various teams become involved.
  - Decide which preliminary tools would be helpful from a forensics perspective. A sniffer on the subnet can be useful in identifying unusual network traffic between systems or into internal network space. An intrusion detection system (IDS) can help by providing warnings of continued exploitation attempts, although in many cases, further exploitation is through credentials or trusts obtained via the currently exploited system(s).

Investigators need to be careful to not jump to any conclusions or make assumptions. There may be some pressure to take the “just re-image the system” approach to return to a known clean operating state or, perhaps, pressure to limit the time to perform proper analysis. Anything that limits the ability for investigators to properly perform their jobs can leave unidentified compromises in corporate assets and preclude any understanding of what information the hacker(s) may have obtained. If the investigation is thorough, the investigators can usually determine the method and scope of the attack and provide proper input about containment, eradication, and recovery.

To estimate the amount of time needed to resolve a compromise, there are many factors to consider. Time is needed for all the following steps: the work to identify, collect, and analyze logs, to perform forensic analysis of the servers, to analyze malware, to correlate data from different sources, and to manage the various corporate responsibilities such as reporting the status and coordinating meetings. Overall, it can take weeks or months to complete a large enterprise-level case. Naturally, the sooner an attack is recognized, the better. It is common to hear how the first 24 or 48 h are critical in physical law enforcement cases. The same is true for most cyberforensics investigations. Hackers will usually work as quickly as possible to get their rootkits, backdoors, and various tools
fully implemented. They usually have scripts that very quickly download tools and install them, which permit them to hop from machine to machine, installing as they go. A speedy cyberforensics response, very early in the hacker’s attack phase, will contain and mitigate much of the damage. Corporate forensics also has what could be called the “golden weeks,” usually a period of about three weeks. During this time period, the cyberforensics investigation team has the full attention and cooperation of the key people in the various business units. Then, sometime during the third week, this starts to “dry up” because people get pulled back into their daily routines. Also, the investigators naturally start to slow down physically and mentally as a result of the many long hours they have already put into the case. The lead investigator has to contend with these issues by making sure that the critical path of the investigation is followed and that certain critical steps (perhaps different in each case) are completed as soon as possible.

In addition to working hard to maximize their progress during the valuable “golden weeks,” the investigators will need to provide information about the case to upper management and the other key people. In this case, similar to most cases in the corporate world, everyone is waiting to find out what was compromised, how the compromise occurred, what needs to be done to contain the problem, how to eradicate the problem, and how to get back to normal business. The investigators, as soon as possible, should collect enough evidence and perform enough analysis to start providing feedback regarding the containment, eradication, and recovery steps. If the case is well managed, then the business application owners, network engineers, firewall administrators, systems administrators, and others can start implementing containment, cleanup, and recovery steps at appropriate times without impeding the additional analysis that the investigators must perform to finalize their assessment of the incident.

An important part of the investigative approach, in any potential hacking intrusion case, is for the investigators to think in a way similar to hackers. The ability to think similar to hackers comes basically from a combination of cyberforensics experience and penetration testing expertise. Hackers often take advantage of the “low-hanging fruit” (the easiest exploitable vulnerabilities). Identifying such “low-hanging fruit” may allow investigators to more quickly identify areas likely to contain potential evidence. Investigators should analyze any recent security audit reports that relate to the environment. Any vulnerability noted in the report(s) may be the avenue used by the hackers. As a parallel effort to the forensics work, a penetration tester could perform a pen-test of a test environment (an internal test or a development environment containing the same applications) to find vulnerabilities. Proper security protocol would also dictate a full penetration test of the affected environment after the investigation was complete and all fixes and patches were put in place. Before any kind of penetration testing is performed, however, it is important to clearly understand the implications. Any testing of the compromised environment could destroy evidence, and all penetration tests must align with legal and corporate policies. The key point to remember is that it is critical for investigators to have a clear and accurate understanding of the environment.

Because of the number of servers and network elements that may be affected in most enterprise incidents, many are classified as large-scale incidents. Such incidents generate so much data to be identified, acquired, and analyzed that investigators need an efficient way to manage it all. One way to do this is to have a team of experienced cyberforensics analysts and others working on various tasks at the same time. Thus, a lot of work is performed in parallel and the approach works well to save time. However, more effort is needed to ensure the quality of the work and to combine the distributed knowledge into a proper understanding of the incident. As soon as possible, investigators should start analyzing any known or suspected compromised systems and their related logs. Some cases are somewhat benign, such as web defacement or retrieval of non-critical information. Even though an attack may look relatively benign, it may not be. It is important to be thorough, to pursue all leads, and to look further, beyond initial findings, to understand the scope of the compromise. For example, a compromise could be serious if there are signs of escalated privileges, rootkit installation, or remnants of hacker tools. Also, investigators may identify artifacts that can be used to determine if other systems have been compromised. Search engines, such as Google® and other tools should be used to look for potential comments about the compromised systems on blogs, hacker websites, chat rooms, etc.
If the investigators are lucky, only one or two systems have been compromised. However, the investigators have to consider several possible scenarios.

- The same hackers may have compromised other systems in the portal using the same vulnerability.
- The same hackers may have compromised other systems in the portal using a different vulnerability.
- Different hackers may have compromised other systems in the portal using the same vulnerability.
- Different hackers may have compromised other systems in the portal using a different vulnerability.
- Multiple hackers may have compromised the same systems, each with their own agendas.

Many investigations involve load balanced web farms, or sets of computers that all host the same web server platforms for use by multiple user communities. A key problem in web farms is that the vulnerability that the hackers used on a given system is likely to exist on all the other web servers. Ideally, in a compromised web farm, every single web server would be analyzed and then be completely rebuilt, whether or not it was found to be compromised. This decision, however, is based on an understanding of the compromise, level of risk, time, and cost.

If the known scope and complexity of the compromise grow as the investigation progresses, then naturally the required cleanup and mitigation efforts will expand. In addition, the realized level of risk to the company changes and the investigative approach might have to change as well. In extreme cases, the whole service or web portal might have to be shut down to protect corporate assets and allow containment, eradication, and recovery.

As already mentioned, investigators must determine which tools would be best to use to analyze systems in a given case. The author is a firm believer in the principle supported by the great Japanese sword master, Miyamoto Musashi, who said, “You should not have a favorite weapon. To become over-familiar with one weapon is as much a fault as not knowing it sufficiently well.” Having a favorite tool could impair the investigators’ performance during forensics analysis because they might ignore other tools and options. Investigators should know how to use a wide range of digital forensics tools appropriately, and know which tools are the most effective in various types of circumstances.

That being said, there are types of cases that benefit from extensive use of “live” analysis tools, which gather and process data such as running processes, lists of open files, and network activity on running computers. Live tools are distinguished from “static” analysis tools, which only work with data stored on disk drives. The typical “static” analysis example is that of analyzing data from a shut down computer. Investigation techniques to retrieve both volatile and static data are more thoroughly described in Chapter 8. The enterprise investigator must thoroughly understand the pros and cons of each type of tool when making choices for the upcoming steps of the investigation.

A web farm is a good example of an environment where forensic analysis would benefit by using live rather than static analysis because:

1. Volatile data such as network activity, running processes, and open files may be critical.
2. Memory dumps could be very useful.
3. The large number of systems could take weeks to image, potentially causing costly downtime.
4. Analysis needs to begin as soon as possible.
5. Remote access (a network connection to the computer) saves travel costs.
6. The investigators need to perform incident response on many systems at the same time.
7. The needs of the business require that the systems remain running and performing as usual.

Many of the live analysis tools require investigators to be on site with physical access to the systems. Enterprise-wide investigations requiring live data analysis require, very conservatively, a team of at least three investigators to stay for a minimum of two to three weeks. This is quite costly. So, using any tools that can perform the required tasks remotely would be of considerable value. The remote forensics analysis tools, such as Guidance Software’s (R) Encase® Enterprise Edition, MANDIANT Intelligent Response™, ProDiscover™ Incident Response, and AccessData® Enterprise use a light footprint and are carefully designed to preserve evidence. Even though they do not rely on any system programs, they will, however, modify memory. Here are some advantages of using remote vendor forensics tools:
Their usage may be harder to detect by hackers.

Volatile evidence (processes, network activity, open files) and file system analysis can be performed online in ways that are less likely to tip off hackers.

They can allow multiple systems to be analyzed and compared at the same time.

They are generally reliable with the preservation and acquisition of evidence.

They can capture evidence files and store them in special evidence containers.

They contain a broad spectrum of features.

Some support snapshot comparisons of volatile data; this can be very useful in identifying malicious programs and processes if comparisons are made against a known clean snapshot.

A pre-incident installation of a remote forensics tool applet would offer very quick incident response and forensic access to the systems.

The courts are often familiar with these tools.

Some of the disadvantages of these vendor tools are these:

They do not support all OS/hardware configurations and file systems.

They can take a very long time to parse and display Linux/UNIX directories and files.

They are quite expensive.

They tend to work better analyzing Windows environments rather than Linux and UNIX.

They are sometimes harder to use in certain investigations.

They need to be well tested in different scenarios to develop appropriate operating guidelines.

Although the major vendor tools just mentioned are popular, they are not the only choice for investigators. Other forensics tools containing scripts for live analysis and acquisition are also effective, especially if known binaries and libraries are used to ensure trusted results and avoid changing the “modify, access, and create” (MAC) times of system programs. Microsoft’s older Windows OnLine Forensics (WOLF), and their tools for law enforcement, Computer Online Forensic Evidence Extractor (COFEE) and Windows FE, are examples of useful script-based tools. Other scripts to acquire volatile data can be found in the older free versions of Helix as well as e-fense’s Helix3 Pro and Live Response tools. Most of these scripts are designed to run from a CD or USB drive mounted on the target system. Scripts can then be run from the media to acquire memory and other volatile data. Additionally, the CD or USB drive is often bootable to permit a complete reboot of the target system without using data from the hard drive, and thus is ideal for the task of performing hard drive imaging or static analysis. Even though, in general, these scripts are not specifically designed to be run remotely, it might be possible to turn them into remotely run scripts such that an administrative login account or an administrative remote execution capability such as WMI or psexec could be used to run them. Because the major forensics software vendors offer limited support for Linux and UNIX, homegrown scripts are common to support these operating systems. Unfortunately, many Windows and Linux/UNIX scripts, by default, will modify evidence as they peruse through operating system files. It is, therefore, important for the investigator to understand the functionality of script tools and use them appropriately. Some of the advantages of the script utilities are these:

They often provide a better view of system activity, running processes, and network activity.

They can provide a more real time monitoring environment compared to the snapshot approach used in the vendor tools.

They can provide more detail and a wider range of volatile data information.

They can be tailored for different kinds of investigations.

They are sometimes the only option for the incident/environment.

Some of the disadvantages of scripts are these:

They might be more noticeable to hackers.

Depending on many factors, they are more likely to modify MAC times.

They might not work as expected due to differences in OSs and hardware.

They need to be developed and tested on a variety of platforms.

They may have to be defended in court.
Sometimes a combination of vendor software and script utilities might be needed. Investigators should be prepared for occasions where the expensive vendor tool cannot quite get the job done, as happens in a variety of cases. Probably the most frequent problems occur in the Linux/UNIX environments. Vendor tools can have problems with certain file system types because most of these tools rely only on their own code to read and analyze the file system structure. Because native operating system binaries are not used or trusted to display directories, files, unallocated space, or volatile data in memory, the integrity of the results can be supported. The tools also do not modify file access times when opening directories or files. This is the very best forensics approach, but it limits the tool’s functionality to only the files systems that are supported. For example, SAN storage, the Solaris ZFS file system, and Linux LVM14 can be problematic for some tools. Additionally, applets running in virtualized environments such as Solaris 10 zones may not permit the forensics tool access to the physical devices. Even certain RAID hardware implementations can cause problems for these tools. One possible solution for some of these problems can be found in an interesting tool called F-Response®, which connects to a small applet on the target computer. The tool makes the drive on the target computer read-only and accessible to the investigators’ computers. It appears as a local, raw, read-only, physical storage device that can be examined with local forensics or other specialized software.

Sometimes the vendor tool just will not work as the investigators would like. For example, one tool will capture volatile data but cannot display the data unless it is connected to the target system. In this case, after the incident is over, there is no way to view the volatile evidence. Additionally, investigators may find operating or performance issues with any given tool. A tool might take too long to open and parse certain system log files or perhaps it cannot sufficiently identify rootkits or malware. The point is that many tools are usually needed to fully process an incident.

It is important to note that hackers constantly watch for administrative activity and, if they see anything of which they are suspicious, they will run cleanup scripts that greatly hamper forensic analysis. Stealthy investigative techniques should be considered when performing any type of live analysis of systems still connected to a network. By “stealthy” the author means that the investigators should consider exactly which names are used when they deploy their tools and how these tools look and behave when run on the live system. When performing incident response, running a program called “dbindex” will draw less attention from hackers than a program called “forensic-script.” The goal is to have hackers think that the incident response script is a normal, everyday process.

Before running the selected tools, the investigators need to carefully consider the order in which to collect evidence. Naturally, the most volatile data should be gathered first, as it will be the first to disappear.15 Volatility is the first concern, but there are other concerns when performing live forensic analysis through a remote connection. Some tasks take a long time to run and keep the investigators from performing any other work. This can be a real issue when using some of the vendor remote forensics tools to remotely view data on Linux or UNIX data storage drives. In these cases, the investigators should find out the mount points of the drives and the size of all the slices before starting to attach to them for preview. The reason this is important is that, as already mentioned, the investigators will have to wait a long time before they can view the drive data. Therefore, investigators must prioritize which slices to look at. It can be highly frustrating to be waiting on the system to preview a terabyte slice that holds user data, when the much smaller /, /var, /etc, /tmp, and other operating system directories that may be on individual slices could have already been analyzed. System directories are usually the first place to start, because frequently hacking artifacts will be found there. Other tasks, such as imaging a drive/partition or performing a string search, or parsing a registry, or indexing can force investigators to wait many hours for completion. Depending on the tool being used, investigators may just have to wait until it completes. Developing and using recommended guidelines that are tool specific can help alleviate some of these issues. For example, a good procedure would be to immediately forensically extract copies of key system files and obvious artifacts (hacker tools and malware) that other investigators can start analyzing. As an initial starting point, certain key files should be obtained and additional files extracted as the need arises. In the Windows environment, some of these key operating system files include registry files, event logs, web logs, database logs, various configuration files, and files wherein operating systems
may store information related to the activity of a given user, such as user index.dat files. In the Linux/UNIX environments, files such as /proc files, hidden files and directories, /etc/passwd, /etc/security/passwd (AIX), /etc/shadow, /etc/group, /etc/hosts, various other config files, plus the logs in /var/log and application logs would be of initial interest. From the user space, root and possibly other user history files should be retrieved. Additionally, at a minimum, a report of running processes, open files, and network status should be obtained using forensically sound methods to gather volatile information. This approach enables parallel work flow, which circumvents the single investigator and forensics tool bottleneck issues.

There are occasions on which a vendor’s remote forensics tools and other tools cannot be used for network access reasons. One issue that appears in corporate environments is that some networks and systems can only be reached remotely via a jump server. Such restricted access can limit the ability to extract data from the systems. It is key to remember that the remote forensics vendor tools need to have direct network access to the target applet or vice versa. There is often difficulty in finding tools to use to quickly respond in “jump server” Linux/UNIX environments. Firewall and router ACL rules could be changed to permit other tools to effectively gain access, but this can take a long time. Incident response teams need to quickly determine that a compromise occurred because a great deal of decision making, planning, and organization must take place right away before proceeding with a full forensics investigation. Hence, it is advisable for an enterprise incident response team to become familiar with potential access paths available within the enterprise and work with operations teams on access plans in advance of the occurrence of an incident.

Whenever critical to the case and whenever possible, obtaining images of memory and disks is beneficial. The process of imaging usually demands most of the resources of the system, network, and/or a forensics tool. As a result, sometimes other forensics activities can only be performed on evidence that has already been extracted for analysis. The question of when to start the imaging depends on the case. It is often important to obtain a copy of memory before any other work is performed. Both physical memory and “paged out” (swap) memory should be obtained if possible. Note that the process of acquiring memory actually modifies the memory, so this needs to be taken into account. Imaging the drive(s) could occur later in the investigation, for example, once evidence of intrusion has been validated. Numerous factors are involved in this decision, such as legal requirements, the type of incident, the forensics tools in use, disk storage, network capacity, and access issues.

Once the investigators find key evidence, some of the remote forensics tools can be used to concurrently search all the other systems for the same evidence. After the forensics tool is connected to all the systems, searches can be as simple as sorting by file name across all the systems to see similar names grouped together, using advanced string search capabilities, or by identifying similar hash values. As anyone might assume, finding the evidence is just part of the process. Understanding what the evidence means will fill in the blanks. Hackers use all different kinds of tools in their trade. To understand the scope of the compromise, analysis of these tools, that hackers installed, needs to be performed. Some of the common tools hackers implement are keyloggers, IRC bot software, distributed ssh dictionary attack software, web command shell programs, rootkits, trojans, stored cross-site scripting attacks, sniffers, local and remote exploits, phishing sites, malware distribution websites, and malware command and control code.

To better understand exactly what the hacker tools really do, investigators will need to perform “malware analysis” of any suspicious programs found on the system. Malware can be submitted to various free Internet services, which run the code in a sandbox environment to determine what the program does. This analysis works pretty well for code extracted from the Windows environments but not so well for Linux and UNIX environments. Commercial sandbox analysis software is available for in-house analysis, such as, Truman, Norman Sandbox®, and CWsandbox. Additionally, investigators can imitate the commercial sandboxes by manually performing behavioral analysis of binaries in self-created sandbox environments using debuggers such as Ollydbg or gdb. Disassemblers such as IDA pro can be used to statically analyze the malicious programs as well. Much of this analysis is technically demanding, requiring a good knowledge of assembler language as well as knowledge of specifically how various functions look when compiled
by different compilers. In many cases, the malware code is made intentionally difficult to analyze by using packing techniques and run-time tricks to deter reverse engineering. Although the effort to analyze malware can be costly, time consuming, and technically challenging, the intent of the hacker(s) and potential scope of the compromise usually begin to come into focus. Additionally, when the tools themselves are run, they may create artifacts that could contain important information, such as keylogger output files, imported registry files, or activity logs. Understanding how malware and hacker tools work can help investigators determine if and how the tools were used. Malware investigations are more thoroughly described in Chapter 6.

Applications and operating system programs provide all kinds of status in the form of log output. Many levels of error messages, authentication status, activity logging, debugging, and other information are continually reported to log files. The example web portal environment used in this chapter is likely to have significant amounts of historical log data in addition to the logs residing on the servers. The cyberforensics investigation team could easily have terabytes of data from application logs, windows event logs, Linux/UNIX syslogs, firewall logs, and sniffer logs. When appropriate levels of logging are used, logs will usually provide the best resource for determining the time frame, attack vectors, and scope of a compromised system. There can be many challenges involved in preserving these logs, fingerprinting the files with MD5 or SHA1 hashes, collecting and/or disseminating log evidence for analysis, analyzing the logs, understanding time zones of time stamp data, and correlating log activity with system or other activity. There are simply no easy answers as to how to determine the best approach to handling large volumes of log data. Also, no single tool can easily incorporate any and all log types. There are tools that can import many common log formats, but it is very likely some logs will require manual intervention. Often, unique skills are required to be able to recognize different attack profiles in logs. Someone familiar with web attacks should be looking at web logs, someone else familiar with Linux/UNIX system attacks should view those logs, and so on. Then, items of interest are extracted, as well as compared and correlated with other data. Depending on the amount of data, key data points from different sources could be loaded into log parsing software or even spreadsheets for sorting and filtering. Some of the freeware log parsing tools, such as the popular Microsoft® LogParser.exe, are great for providing query capabilities and log correlation. Of course, Specialized Log Manager or Security Incident and Event Manager (SIEM) appliances may have already been deployed. If they already contain the logs that the investigators need, they could probably be used to perform most of the log analysis. Either a database or some SIEM/log appliance is the answer for large volumes of data that need indexing and query capabilities. The goal is to gain insight from the data, while preserving evidence. Advanced data analytic techniques are more thoroughly addressed in Chapter 9.

Sometimes, anomalies or suspicious activity in network traffic leads investigators to such things as bot software communications, spamming software, or even live hacking efforts. Therefore, at some point, there might be real value in deploying sniffer technology in the compromised network space on switch spanning ports or taps; this can be done at any time, but early on in the investigation is preferable. Investigators might not know what to look for in the early stages of the investigation. For example, they might not know how to identify malicious traffic from valid traffic. Once they become more familiar with the environment and the nature of the compromise, their ability to target “suspicious” traffic improves. The tools used can be a great help as well. An AT&T-developed tool called GS Tool can handle huge network volumes. Another tool called NetWitness® Investigator presents data in a unique aggregated form that can help find unusual traffic, and a tool called SilentRunnerTM is worth noting. Wireshark and tcpdump are free tools and very useful for smaller-volume situations. It also is important to remember the intrusion detection systems (IDS) and intrusion prevention systems (IPS). They sniff network traffic looking for known attack signatures. An often-used IDS/IPS called snort developed by SOURCEfire® is fairly easily deployed. IDS/IPS and sniffers can also assist in monitoring a “cleaned” compromised environment to see if any unusual traffic still exists. There are many choices available in the sniffer and IDS/IPS arenas. The trick is finding what works best for the situation at hand. The utility of network forensics tools is more thoroughly covered in Chapter 7.
There are many ways for hackers to gain access to systems. Some of the most common are through web application vulnerabilities, but also, a variety of vulnerable services can be exploited with remote buffer overflow or format string attacks. Unfortunately, access to systems is often much easier than expected. Hackers are frequently successful at guessing account names and passwords to systems and applications; this is usually done by using tools that attempt multiple logins using lists of common IDs and passwords. Investigators sometimes need to determine if the passwords to known compromised accounts were easily guessed. For example, the investigators may know that the hackers logged in using the “jsmith” account. It might not be possible to find out and verify the password of the jsmith account because jsmith could have forgotten the password, or left the company, or is a customer, or maybe the account was put in by previous hackers. The only choice in this case is to try to crack the password. After cracking the password, if it turned out to be an “easy-to-guess” password, that is, a commonly used password found in password dictionaries, then it is more likely that access to the account was gained through guessing the password.

Usually passwords are stored in some encrypted form called a hash. Commonly used encryption algorithms are LANMAN, NT hash, MD5, SHA1, crypt, and XOR; there are many others as well. Some algorithms can also be “salted,” which adds more complexity.

There are basically three approaches to “cracking” passwords. The first method is to use Rainbow tables, which works well with the “unsalted” LANMAN and NT hashes used in Windows systems. In simple terms, Rainbow tables use a special lookup algorithm that scans through precomputed hashes stored in chains that are optimized lengths of hash sequences. The algorithm can very quickly search through chains to look up lists of reference hashes (hashes retrieved from the compromised system), which then return the clear-text passwords.

The second method is to use a dictionary attack (an extensive list of common passwords) with tools such as John the Ripper. Using the reference hashes, the program takes every word in the dictionary, encrypts it with the same salt (if used) and encryption algorithm used on the reference, then compares the resulting hash to the reference hashes. If the two hashes are the same, then the password is the word in the dictionary that created the matching hash.

The third method is to use a brute force attack, which tries every possible combination of a set of characters. A set of characters is made up by choosing some combination of the following: all lowercase letters, all uppercase letters, numbers, or special characters. The cracking process will take longer if more complex character sets are used and/or longer passwords are used. The investigators should find out if password complexity rules were enforced by the application or operating system that used the reference hashes. For example, user policy could enforce the practice of requiring a minimum 6-character-long password that contains lowercase letters and numbers. This knowledge is valuable in being able to apply the correct character set and password length settings to the password cracking software. Once the appropriate settings are applied, the password cracking software tries every combination of the character set to create a hash and compares it with the original reference hash sample(s). Simple passwords are usually cracked within a day, if not sooner. Complex passwords can take weeks, months, and even years. The investigators have to identify the type of encryption to be able to crack it: this can be simple, hard, or even impossible. In most cases, the hashes are in recognizable formats and salts are usually delimited for easy identification, such as this UNIX /etc/shadow MD5 hash: “$1$YUf57J75$fs4wB5AW6t188vTu6F.guM.” The “$” is the delimiter, so “1” denotes the hash type as MD5, “YUf57J75” is the 8-character salt, and “fs4wB5AW6t188vTu6F.guM” is the actual hash. However, sometimes investigators find themselves working with hashes that are just raw data and information about the hash type or salting is not available. For example, this hex representation of a 28-byte cipher: “09e5e70a554d71fb9d2bd4ab5552f7850de1878913b17091377374d” turned out to be a SHA1 20-byte cipher appended to an 8-byte salt. Once the encryption method is identified, a tool that can be used to crack the password(s) needs to be identified. Sometimes a special tool or patch to an existing program such as John the Ripper needs to be written. In addition, some applications use proprietary hashing algorithms that can make it just about impossible to figure out.

For a long time investigators have been using images of memory to extract “strings” data that could
contain passwords, program usage/comment information, and communication session content [for example, Internet Relay Chat (IRC), instant messaging (IM), and other useful information]. In the past, most investigators did not have the skill set and/or time to manually parse through memory structures to extract the breadth of information contained in memory. In a white paper written by Greg Hoglund called *The Value of Physical Memory for Incident Response*, Greg mentions various challenges of Windows physical memory analysis. Below is an abridged summary of some of these challenges.

- To properly understand the memory structure and potential data structure differences, the operating system, the version, and the service pack level need to be determined.
- Undocumented operating system data structures need to be understood to effectively extract useful data.
- Both physical and “swapped” or “paged” memory needs to be properly organized and structured to rebuild a complete memory model.
- Binary EXE or DLL code extracted from memory needs to be modified to add a portable executable (PE) header to make the code an executable file.
- MD5 or SHA1 hashes of files are not useful in identifying those files loaded in memory because the hashes will not match.

Tools are becoming more and more sophisticated in their ability to parse a memory image (live or static) and extract key information about the contents in memory, such as running processes and drivers (even those hidden by rootkits), open files and registry keys, as well as network connections, etc. These tools are predominantly supporting the Windows operating systems, and they are effective enough in this arena that investigators should seriously consider grabbing a memory image as the first step in a live forensics investigation.

There are commercial tools such as HB Gary’s Responder, Encase® Enterprise Edition, and Forensic Toolkit® 3.0 that image and analyze memory. Also, there are free tools available such as MANDIANT’s Memoryze, Volatile Systems’ Volatility Framework, and Andreas Schuster’s PTfinder.

In large enterprises that house Personally Identifiable Information (PII) about consumers, protecting data privacy is always a concern. Laws such as the California Database Security Breach Notification Act (SB 1386) have been initiated as has similar legislation in other states. Therefore, if possible, the investigators should consider performing database forensics, especially if they contain credit card or other personal data or there is reason to suspect unlawful database access. This is, however, a new field of investigation for investigators. Few tools exist, and they would most likely have to be database specific. Oracle®, Sybase®, Informix®, MySQL, and Microsoft SQL Server® are just some of the many types of databases that exist. Each one would have specific methods of obtaining database information. Areas that may contain potential evidence are audit trails that would have information about database changes such as updates, inserts, and deletes. These audit records would generally record the date and time of the transactions and information on the user who performed these activities. Unfortunately, many databases are so large and so heavily used that it can often be impractical to perform forensic analysis.

David Litchfield, a renowned database security expert, has explored developing a database forensics tool called the “Forensic Examiners Database Scalpel.” Various sources, primarily metadata from transaction logs, web logs, and deleted data are used for analysis. Litchfield says, “It will be able to do comparisons between backup files and the metadata of the database to look at differences between the two and work out who did what when.” Even in cases where database auditing is disabled, it may still be possible to find evidence of SQL queries, and Litchfield provides an example of this in his article, “Oracle Forensics Part 5: Finding Evidence of Data Theft in the Absence of Auditing.” He explains that in Oracle10g Release 2 queries are compiled into an execution plan that leaves behind potentially useful evidence in special tables used to optimize database resources. The contents of these tables can be dumped and carefully analyzed to provide information pertaining to database usage.

For the most part, databases are targeted to extract valuable data, but sometimes databases are used to store persistent cross-site scripting (XSS) code used to further exploit anyone viewing that content with a browser. The hacking community will always be creative and innovative in how compromised assets are used. Because databases are often so vital and heavily integrated with corporate products and services, the reasons to perform database forensics will undoubtedly keep growing.
2.3 Case Study

The enterprise environment provides numerous opportunities for a discussion of cyberforensics cases. Standards of professional ethics prevent the author from a detailed description of a specific case. However, there are enough examples of cases wherein a hacker compromises a web portal to be able to use this scenario as a case study without unnecessary disclosure. Web portal investigations provide excellent case studies because these environments can be enormously complex in many dimensions:

- Technically: because they often employ a complex mixture of networking, computing and application technologies
- Functionally: because web services can use complex authentication mechanisms, large databases can be involved, and multiple applications may be designed to interact with each other and other support utilities
- Legally: because the Payment Card Industry Data Security Standard (PCI DSS), the Sarbanes-Oxley Act (SOX) of 2002, the Health Insurance Portability and Accountability Act (HIPAA), and numerous other federal and state government regulations or contractual requirements may entail certain actions
- Economically: because these environments can be a source of invaluable income
- Politically: because various parties such as corporate clients, internal organizations, the press, and other interests may be involved

A breakdown of cyberforensics issues in such an environment permits the author to synthesize known cases into an example of the issues and challenges related to cyber investigations in a corporate IT world. From this discussion, the reader will be able to construe how other types of cases would be investigated.

2.3.1 The Incident

A company can become informed of an incident in many ways. Perhaps someone sees disturbing data in a log file or an IDS issues an alarm. Companies can even be informed about an incident from an outside party. In this case, an administrator notices TCP traffic on a strange port and immediately calls for incident response and forensics support.

2.3.2 The Environment

Web portals are often implemented in load-balanced web farms. It is common to find 20, 40, 60, or more systems supporting the corporation’s website content. In today’s environments, the web servers are probably virtualized and are likely to be supporting multiple sites on virtualized IP addresses. There could also be a variety of other servers such as back-end database servers, domain controllers, LDAP servers, DNS servers, mail servers, firewalls, management jump servers, and the list goes on. The first task of the investigator is to obtain or create documentation that clearly identifies the infrastructure and application architecture of the supported components, including the focus of the investigation.

2.3.3 Initial Investigation

After initial information gathering, interviews and meetings with various parties such as the application team, network administrators, and the legal department, the investigators realize that incident response and forensics have to be performed on live systems. In this case, remote forensics vendor tools would have to be supplemented with additional tools to evaluate all the systems in the affected environment; this is caused by the very restricted network ACLs and a broad mixture of operating systems including Windows, Linux, and various UNIX systems. For the Linux/UNIX systems, the investigators have to work through a jump server and need to use their own UNIX shell script incident response tool. Although many of these kinds of scripts exist, they need something that is portable across Linux, Solaris, AIX, and HP-UX. Also, all the freely available scripts use netcat to shovel evidence back to the investigators, which does not always work because of firewall/router ACLs. The investigators’ script automates collection of volatile data,
common log and configuration files through a terminal connection. The tool uses known safe binaries that are statically compiled when possible. Additionally, this script opens files using Dan Farmer’s and Wietse Venema’s ica27 program to preserve MAC times. When ica does not work (for example, with an incompatible file system), MAC times of these files are recorded before copies are hashed, encrypted,
encoded, and sent back on the same channel as the ssh or telnet session. Using the same ssh or telnet session to transport the data avoids the problems associated with using netcat in port-restricted environments. On the investigators’ workstations, all activity on the systems is recorded for validation and auditing. All received evidence and the activity log data are hashed and stored in a zipped archive. Investigators use workstation-based scripts to copy evidence from the zipped archives, then decode and decrypt them for purposes of analysis.

Memory images are obtained on all systems that support the capability. This is done because:

1. Processes hidden by kernel level or Trojan rootkits are visible.
2. Much of the volatile information is captured in one shot, just by capturing memory.
3. The investigator, as far as possible, goes unnoticed by an intruder that is still logged into the system because dumping memory can be a more stealthy process than running a whole suite of tools to otherwise capture volatile data.

The initial analysis of the collected evidence provides some insight into the method of compromise and enables input to support preliminary containment efforts. Web servers are usually compromised by misconfigurations or code without proper input validation, providing an opening for a hacker. Recent security audit reports were useful in identifying potential vulnerabilities to focus on. In this example, it turns out that a local file inclusion (LFI) vulnerability in a particular page would allow reading some files on the web server. If the hacker can inject code into a known file, perhaps a log file, they can once again use the LFI to read and execute the code. Once this occurs, further exploitation is likely. Ultimately, administrative level access of the compromised systems could be possible. In this web portal scenario, evidence of an LFI attack is found in web logs. Because a user account is required to gain access to the vulnerable web page, it is obvious that either the actual user identified in the logs is the hacker or, more likely, the account is compromised. Analysis of log files shows that code was injected to upload and run an IRC bot with the permissions of the web user. File system analysis also finds IRC bot executable and configuration files with MAC times that correlate the log file data.

2.3.4 Extended Analysis

After initial investigation of the target web servers, investigators must determine if the hacker managed to elevate privileges and whether hacker activity extends to other systems on the network. Management of all these systems requires various types of login accounts and the administration of them can be very challenging. In an effort to make this as easy as possible, administrative accounts (used by system administrators) and application accounts (used by application administrators) might have the same or similar types of passwords. They are also likely to be used across multiple systems. In a Windows environment, administrators may authenticate locally or through a domain controller that can permit access to a wide range of systems. In a Linux/UNIX environment, it is possible that ssh keys have been set up to permit “password-free” access between systems. Unfortunately, the compromise of a key administrative ID could permit access to many systems. Hackers are not usually content with compromising just one system. They often identify other targets through various techniques while on the first system. If the compromised system ID can access other servers without being challenged to provide a password, or if the passwords are the same, hackers can access other systems easily. A common practice that makes life easier for hackers is that many systems are on the same subnet or broadcast domain. Even when systems are on different subnets, and not visible at the data link layer, they may be listed in configuration files or identified by viewing network activity with the “netstat” command. Basically, it does not take long for hackers to glean knowledge of other potential targets. If they cannot elevate privileges from an application-level ID to an administrative ID or accomplish their goals on the currently compromised system, then jumping to another computer system may provide that opportunity. As hackers are likely to compromise multiple systems, investigators need to look for signs of this activity. System administrators are interviewed and assist in identifying any unusual login log file entries. Systems are searched for unusual artifacts that
may be remnants of hacker activity. Systems are evaluated for rootkits. Network activity logs are reviewed for unusual network traffic.

As the investigation progresses, different tools are used as environments and investigative needs change. Many gigabytes of historical web logs and other logs of correlation value that date back to the creation date of the vulnerable web page are collected and analyzed to look for earlier attempts of exploiting the LFI vulnerability. The remote forensics vendor tools extend the search and correlation functions. Key account passwords, in particular, the compromised user account, are cracked to determine password complexity, and the results clearly show that the user account had a simple guessable password, which supports the likelihood that the account was compromised. Without question, hackers can obtain even complex passwords from phishing sites and keyloggers, so this also must be taken into account.

Unallocated space is evaluated for potential deleted evidence. Timelines are created by correlating evidence from a variety of network, system, security and application logs. Evidence-gathering tools report details that are not known to those who reported the incident nor to those who called in the investigators. Information received from both parties conflict with data retrieved from the system itself, and interviews are conducted to sort out misconceptions and misunderstandings. These steps are conducted in an atmosphere of pressure as investigators try to expand knowledge from how the compromise started, to understanding the scope of the compromise, and how best to contain it, all while fielding constant queries from the legal department, management, application owners, administrative teams, etc.

This web portal scenario has databases associated with the application architecture. Forensic analysis is done with the help of a database administrator who understands the database structure and content. Results from the analysis and interviews are used to narrow the scope of database logs and tables of interest; these are likely to be application-related logs as opposed to error or transaction logs created by the database management system itself. For example, the user account revealed in the web server log is used to focus on queries of the application commands executed by that user. The actual data that the user accesses are inferred from the commands. Queries are carefully chosen in consultation with the database administrator to minimize impact on the production environment while at the same time revealing the maximum amount of information that may be useful to the investigation.

### 2.3.5 Investigation Conclusions

Investigation and analysis can, in theory, go on forever. A good lead investigator needs to keep the work effort focused and productive with reference to the goals in the case. Because the web portal environment is so large and complex, a good question is, exactly when does the cyberforensics investigation end? No hard-and-fast rule can apply. The investigators’ primary goals in this case are to do the following:

1. Identify and preserve evidence.
2. Determine, as accurately as possible, the method, time frame, and scope of the compromise.
3. Provide feedback about containment, remediation, and security enhancement.
4. Perform the investigation with as little impact to the corporation as possible.

Cases can, of course, be closed by corporate management or the legal department. But, normally, the lead investigator determines when to close the case by reviewing the goals and balancing the needs of the business with the overall analysis of case evidence. The investigation of every case arrives at a point when new evidence of probative value starts to dry up and the corporation’s financial/business needs prevail. This point sometimes occurs before the goals of the investigation can be reached.

In this web portal example, the owner(s) of any compromised user IDs are informed if the hacker managed to obtain PII database information. Any required legal or regulatory procedures are initiated. Appropriate cleanup measures are performed while continued monitoring of the environment takes place. Network and application security scans are scheduled and are followed by a thorough penetration test. These types of compromises are costly to a company. However, there are benefits. Security awareness is raised and at least one area of the corporate IT infrastructure becomes more secure.
2.4 Issues and Trends

2.4.1 CyberForensics in the Corporate Environment

Based on experiences from many internal and external investigations, some key points may be useful to the reader.

Even the smallest problem on a computer system can lead to identifying a huge compromise. Casually ignoring unusual system behavior or signs of compromise can be devastating. If the unusual behavior was noticed at the beginning of a compromise that no one investigated, the opportunity to nip the attack “in the bud” would be missed and the hackers would have plenty of time to explore, to probe deeper, and to enlist the help of other hackers. On the other hand, if the compromises were already accomplished, no one would be remedying the situation. Here are some examples:

- An administrator notices that files in the /tmp directory are getting deleted. Everyone assumes that it was someone else on their team who is doing it. It turns out that it was poorly written hacker code that is causing the problem, and the investigation identifies that rootkits are now installed on several systems.
- A client’s web page is defaced. The investigation determines that the client’s ID and password are compromised. The first thought is that the password was easy to guess. However, that is not the problem. The hackers have figured out how to bypass the authentication mechanisms.

Both these cases turn out to be complex investigations.

- Setting up phishing sites (for example, a cloned bank website designed to entice users to enter in their ID and passwords)
- Setting up a malware/hacker tool distribution site
- Installing covert communication channels
- Using the system to attack other sites (for example, distributed ssh ID/password guessing attacks)
- Installing sniffers and keyloggers

Any network space that is exposed to the Internet needs to be well secured. Everyone already knows this, but in reality, things do not always go as planned. Unsecured routes to internal network space can exist for a variety of reasons:

- Minor errors in firewall rules
- Errors in router ACLs
- Systems with IP forwarding enabled
- Temporary test systems being casually deployed
- Incorrectly installing or patching systems
- Weak passwords and administrative mistakes

Additionally, often unmonitored corporate entities inadvertently create vulnerabilities and permit access to internal networks: these would include small labs, satellite offices, business partner networks, outsourced computer administration services, consulting services, etc. Cases involving Internet facing systems need to include careful evaluation of all these possibilities.

It is a common practice for development and administration groups to keep documentation on a corporate server: these often include application documents, network diagrams, procedures, and contacts. Additionally, administrative tools (for example, simple database administrative scripts, UNIX shell scripts, Windows batch files) as well as source code can often be found. Unfortunately, this information is sometimes found on production servers. Security scanning tools are not likely to flag these files as potential vulnerabilities, but hackers look for this information. There was a case in which hackers found the source code that detailed how an encryption was deployed to store passwords in a database. This knowledge was later used to circumvent the encryption relating to an application on a completely different network than the one originally compromised.

Hackers will use compromised systems in every way imaginable, including these approaches:

- Running IRC reflectors and servers
- Setting up a DVD movie distribution site
- Setting up malware infecting sites
- Setting up hacker training sites
- Installing rootkits and backdoors
24

- Setting up porn distribution sites
- Setting up botnet command and control (software used to communicate and issue commands to many thousands of bot-infected computers)
- Installing spamming software
- Launching wars on other hacking groups
- Probing into internal networks
- Many areas. In reality, the majority of cyberforensics cases focus on employees and contractors. The following list contains examples of the kinds of cases frequently investigated:

- Intellectual property
- Corporate espionage
- Corporate violence
- Cyber harassment
- Unauthorized use of the Internet
- Sexual harassment
- Threats
- Pornographic material
- Unauthorized access
- Fraud (for example, medical benefit fraud)
- Off-duty employee conduct
- Bribes
- Burglary
- Using work time inappropriately

Aside from hacking and employee cases, investigators are sometimes asked to analyze systems for potential malware. One such case involved an executive who had spent time in China and wanted to be sure their laptop was not compromised while there. Also, corporations are often involved in legal issues that require the use of e-Discovery.

E-Discovery tools are usually modules of remote forensics tools and, as such, investigators are often involved in these efforts.

2.4.2 Considerations for the Future

The great explosion of smart phone and mobile device usage means corporations now need to provide suitable forensics support. The ever-growing and wide variety of devices forces investigators to purchase and use different hardware connectors and software. The procedures used in handling these devices are quite different than in computer systems. When a need to perform forensics arises and the device is still turned on, it is usually kept on; however, that can cause problems.

Another layer of complexity is applied if there is a long transport time in taking the device back to the lab or if the equipment needs to be shipped out of state for analysis. In these cases, power needs to be maintained on the device. If the device uses a password-protected auto-lock feature, the investigators will need the password or keep the device from entering the protected lock-up mode. Also, communication with the device may have to be prohibited using a Faraday cage.30

Aside from these complexities, keeping up with the constant change in the cell phone/mobility industry is daunting.

Cloud computing and virtualization have a very bright future for corporations.31 From a forensics perspective, this is a new territory that comes with its own complexities and issues. There could easily be some real benefits to digital forensics in these environments:

- Nearly instant backups of cloud environments would be possible.
- Storage for backing up log files and other evidence would be available.
• The speed of the cloud in making backups or images would be useful.
• Special forensics systems with a suite of tools could be part of the cloud and called up as needed.

However, there are many questions about data integrity. Cloud Service Delivery Models, such as Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS), all have their various controls. In that light, the following questions arise:

• When data is deleted under each of these models, is it really gone? Could there be copies of the data under one of the other service layers?
• Can data cross-contaminate other clouds or trusted boundaries, particularly in multi-tenant operating environments?
• Could data identified through forensic examination be a contaminant or comingle data from other sources?
• How would data encryption methods affect the ability to perform forensics?

Answers to these questions will have to be answered over time and will also vary depending on different cloud architectures and deployment.

It is likely that forensic approaches will need to be modified to support cloud environments. For example, sniffers may need to be installed under a virtual machine to permit the capturing of traffic between virtual systems that are communicating across the hardware backplane. Cloud-provided forensic workstations may need to be used to gain quick access to both static images and live systems. There may be a need to rely more on cloud administrators to perform many of the backing-up and imaging tasks.

Memory forensics is rapidly becoming more mainstream. As this tendency becomes more prevalent, the expectation for investigators to include memory acquisition and analysis as a standard practice will grow. This expectation may cause difficulties when dealing with certain UNIX systems that do not support easy access to memory.

The insidious use of bot malware and the expanding use of client-side attacks drastically change the attack vectors that most threaten corporations. These attack mechanisms place hackers directly inside corporate networks. Ethical and dependable employees can have their systems taken over by hackers and then used to further attack corporate assets. Companies need to focus more and more attention on identifying internal attacks stemming from the use of these methods.

There is a growing need for cyberforensics investigators to rely on various corporate individuals to assist in handling or storing evidence. In most cases these individuals are not properly trained and need to be carefully managed by the investigators. The author expects this trend to continue to grow as the IT world expands in complexity.

Currently, there is a lot of talk in the industry regarding various states in the United States regulating cyberforensics investigations. Many states require that cyberforensics investigators be private investigators. While this is not an issue for investigators performing internal corporate investigations it may affect corporate investigators in other ways. Today discussions exist about requiring standard certifications for all cyberforensics investigators. There are many sides to these issues. It is true that unqualified individuals are working as cyberforensics experts and perhaps some kind of qualifications or controls should be used to separate the qualified from those unqualified. That being said, there are plenty of highly experienced, talented cyberforensics investigators who have no formal training in the field. It will be interesting to see how all this pans out.

In summary, cyberforensics investigations will continue to become much more complex. The knowledge needed to perform investigations is rapidly becoming more diverse, and as a result, there will be a need for investigators with different specializations. Not everyone is cut out for this field. Cyberforensics investigators need to be self-motivated, detailed, critical thinkers with a willingness to keep up with the ever-changing technology. Here are a few other traits that are important:

• Interviewing skills
• Analysis skills
• The ability to focus
• Integrity
• Tenacity
• Report-writing skills

The work pressure can be enormous. However, it is deeply rewarding to solve a complex puzzle and provide a valuable service to the company.
Notes

1. RAID (Redundant Array of Independent Disks) provides a way to store data across different disks; this can improve performance, increase the mean time between failures, and provide fault tolerance. The operating system sees a RAID as a single logical hard disk.

   SAN (Storage Area Network) provides a channel-attached centralized pool of disk storage to servers.

   NAS (Network-Attached Storage) is a server on the network that is dedicated and specialized to handle file reads and writes.

2. LDAP (Lightweight Directory Access Protocol) is a protocol for accessing information directories, often used in authentication mechanisms.

3. Virtual LAN, commonly known as a VLAN, is a group of hosts with a common set of requirements that communicate as if they were attached to the Broadcast domain, regardless of their physical location. See: http://en.wikipedia.org/wiki/Virtual_LAN

4. Electronic discovery (or e-discovery) refers to discovery in civil litigation that deals with information in electronic format, also referred to as electronically stored information (ESI). See: http://en.wikipedia.org/wiki/Electronic_discovery

5. An applet is a generic name for a small program that, in this case, listens on a network port for commands.

6. A “script kiddie” is a hacker culture term used to describe a less-experienced hacker who uses hacking tools without the technical understanding of how the tools work.

7. Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are network monitoring devices that identify attack patterns. The prevention system will also terminate a network connection once an attack pattern is identified.

8. An artifact would be files, processes running in memory, database entries, or log data that represent signs of compromise.


10. Footprint, in this case, means that the forensics program running on the compromised system takes little storage space and is minimally invasive to the system.

11. Files usually have associated modify, access, and create time stamps. These times can be critical evidence in a forensics investigation. They help an investigator develop a timeline of activity.

12. Windows Management Instrumentation (WMI) is the primary management technology for Microsoft® Windows® operating systems. It enables consistent and uniform management, control, and monitoring of systems throughout your enterprise. See: http://www.microsoft.com/technet/scriptcenter/guide/sas_wmi_overview.mspx?mfr=true

13. PsExec is a lightweight telnet-replacement that lets you execute processes on other systems, complete with full interactivity for console applications, without having to manually install client software. PsExec’s most powerful uses include launching interactive command-prompts on remote systems and remote-enabling tools such as IpConfig that otherwise do not have the ability to show information about remote systems. See: http://technet.microsoft.com/en-us/sysinternals/bb897553.aspx

14. LVM is a logical volume manager for the Linux kernel; it manages disk drives and similar mass-storage devices, in particular large ones. See: http://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux)

15. The most volatile data, from most to least, are memory, network status and connections, running processes, and data on hard drives or other media.

16. Jump servers are access gateway servers. They are usually used to permit an administrator to securely access network isolated systems by first logging into a jump server using ssh or telnet, then connecting to the target system.

17. An IRC bot is a set of programs that connects to Internet Relay Chat as a client. These connections appear similar to any other IRC user, but the purpose of the bot is to perform automated functions.

18. A sandbox environment is an isolated computer network that can be used to analyze running malware without the risk of infecting anything outside the sandbox.

19. A salt is a set of random characters that are used as part of the encryption key. Both the hash value and the seed are needed to crack a password hash.


23. Persistent cross-site scripting (XSS) is the storing of malicious scripts that control browser activity. Internet message boards are frequently targeted because user browsers would be affected whenever a XSS stored posting is read.

24. Web portals are networks supporting web servers, usually for a large audience such as the Internet.

25. A web farm is a collection of web servers running the same web application in a manner that distributes the load evenly across the servers.

26. A single computer can run multiple instances of “virtual” operating systems, which saves money on hardware and electricity. Also, a single computer can be referenced by multiple “virtual” IP addresses often used to host different websites on the same computer.

27. Dan Farmer and Wietse Venema developed The Coroner’s Toolkit, which included the icat program that opens files based on the inode number without modifying the MAC times. Icat is currently part of the Sleuthkit and Autopsy tools supported by Brian Carrier.
28. Hackers can use sniffer software to capture IDs and passwords or hash credentials. This approach works best when many systems are on the same broadcast domain.

29. A broadcast domain is a logical division of a computer network in which all nodes can reach each other by broadcast at the data link layer. See: http://en.wikipedia.org/wiki/Broadcast_domain

30. A Faraday cage is a metallic enclosure that prevents the entry or escape of an electromagnetic (EM) field. See: http://searchsecurity.techtarget.com/sDefinition/0,sid14,gcis42282,00.html

31. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. See: http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc

A virtual machine (VM) is a software implementation of a machine (that is, a computer) that executes programs such as a real machine. See: http://en.wikipedia.org/wiki/Virtual_machine

32. A typical example of a client-side attack is a malicious web page targeting a specific browser vulnerability that, if the attack is successful, would give the malicious server complete control of the client system. See: http://www.honeynet.org/node/157