CHAPTER 5

Reacting to Security Incidents

Reacting to security incidents can be an overwhelming and difficult task if you are not prepared. This chapter covers several best practices, techniques, and tips for use when reacting to security incidents. In the previous chapters, you learned how to identify, classify, and trace security incidents. Without successful identification, classification, and traceback, you will never be able to effectively react to any security event. Therefore, it is important that you understand the topics covered in previous chapters before reading this one.

Adequate Incident-Handling Policies and Procedures

The steps you take when reacting to security incidents depend on the type of threat you are mitigating. For example, if you are mitigating a distributed denial-of-service (DDoS) attack, you will probably not take the same steps as when reacting to a theft of information where the attacker does not make that much noise on the network. However, when reacting to any security incident, time is one of the most critical factors.

It is extremely important to have well-defined incident handling policies in place. In Chapter 2, "Preparation Phase," you learned that without defined policies and procedures for mitigation, you can put yourself in a difficult position when a security outbreak or event occurs. Following these policies or procedures is important.

These policies may be in the form of standalone documentation, or they may be incorporated into other documentation such as company security policies or disaster recovery plans. You may consider developing different procedures and response mechanisms when responding to a direct DDoS attack versus a worm outbreak, or when information has been stolen. Not all security incidents are the same, and you should make sure that the appropriate response procedures are in place.

You should try to create a security policy and be serious about covering all facets of security. Ideally, you should develop security policies in the preparation phase.

Collaboration between support teams within your organization may be necessary when responding to security incidents. After you have successfully identified a security incident, classified it, and tracked it, you must notify the appropriate personnel. For example, if you are a member of the Information Security (InfoSec) or Security Operations (OpSec) team, you may need to involve administrators from separate parts of your organization. You may

not have access to the affected device or may not be an expert on a specific application. This is why collaboration is so important.

The reason for setting up collaboration between support teams is to establish lines of communication and ensure that personnel understand the areas of responsibility and capability for each partner. In addition, you should provide a detailed description of the incidents technical aspects to your collaborative teams. This will aid in prompt acknowledgment and understanding of the problem. However, great care should be taken, because you do not want to distribute sensitive information unnecessarily.

You should also have adequate emergency procedures in place. In some cases, you may need to discuss issues and tasks within external teams. For example, suppose that you are a member of the OpSec group and you are trying to get information about a specific system that an external team controls. After several attempts, you have received no response. With the correct escalation procedures in place, the task of getting the right people involved becomes easier. Similarly, you should have emergency procedures when other teams try to engage your staff. The main goal of incident response is to restore control of the network and its systems and to limit the impact and damage. Many people say that, in some cases, shutting down affected systems or disconnecting the system from the network may the only practical solution. However, if you have the necessary tools in place, you may be able to quarantine and remediate such systems without unplugging them from the network. For example, you can use routing as a security mechanism and isolate systems within your network. You can use mechanisms such as remotely triggered blackholes (discussed later in this chapter) and in other cases put systems in quarantine segments so that you can patch them accordingly when security outbreaks occur.

Having a systematic approach for patch management is crucial. For instance, if you have a good system in place to provide security operating system and application patches as soon as they become available, your systems are far less likely to fall prey to major attacks. An updated security management system is not a top priority for many companies; however, attackers, worms, and malware do not wait for you to patch every system manually. More importantly, in the case of worm outbreaks, having a distributed patch management system can save you and your staff considerable time thereby saving your organization money.

It is important to create checklists of procedures to be followed during an incident. Documenting events as they happen is important. On most occasions, you may feel as if you do not have time to completely document events in detail during the incident. However, during the identification, classification, and traceback phases, you should gather as much information about the incident as possible. Attempt to answer the following questions:

- What type of incident are you experiencing?
- When did the attack occur (date and time)?
- Where did the attack occur?
- What systems were affected and compromised?

NOTE Chapter 6, "Postmortem and Improvement," includes examples of these checklists and incident response reports.

These are some of the most fundamental questions that need to be answered. You may develop more specific questions on a case-by-case basis.

Another procedure that you must document is when to involve law enforcement. Incident response is probably one of the disciplines most affected by legal considerations because many incidents involve some sort of crime. Consequently, your organization might want to prosecute the attacker, and in this case, it must consider the legal implications of the incident. If legal implications are present, you must assist law enforcement in all aspects of their investigation. Different laws and regulations are covered in the next section.

Laws and Computer Crimes

In most cases, United States and international laws might affect or impact the incident response process. If you want to prosecute an attacker, you might merely have to contact local authorities. In some cases, however, you will need to contact the Federal Bureau of Investigation or equivalent organizations in other countries, especially when dealing with attacks that involve international boundaries. International and inter-jurisdictional cooperation is difficult. What is illegal in one country may not be in another.

Typically, you have three different options. The first option is to mitigate the problem and move on. The second is to prosecute the attacker in his own country (assuming that the security event you experienced is illegal in that country). The third option is to apply for extradition and prosecute the offender in the country where the incident happened. If you opt for the second or third option, you should seek assistance from your local authorities.

NOTE The procedures and circumstances for engaging law enforcement depend on your local laws. International laws may also apply.

The U.S. laws distinguish between crimes *against* computers and crimes *involving* computers. For example, a DDoS or a person gaining unauthorized access to a computer or network is classified as a crime "against a computer." On the other hand, if a person commits an assault against someone else or any other felony in which a computer was only the tool used to commit the crime, this is classified as a crime "involving a computer."

The "Computer Fraud and Abuse Act" is the standard statute covering computer crimes in the United States. This was initially introduced in 1986 and updated ten years later in 1996. Title 18, Section 1030, covers crimes against computers.

NOTE You can access Title 18, Section 1030 at the Cornell University Law School website at http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00001030----000-.html.

The U.S. Department of Justice has a website where you can obtain specific information on who to contact when reporting a security incident. You can access the website at http://www.cybercrime.gov/reporting.htm.

An excellent document titled "Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations" can be accessed at http://www.cybercrime.gov/s&smanual2002.htm.

Another initiative by the U.S. government is the Internet Crime Complaint Center (IC3). IC3 is a partnership between the Federal Bureau of Investigation (FBI) and the National White Collar Crime Center (NW3C). The website is http://www.ic3.gov.

TIP *Infragard* is an organization that is the product of a collaborative effort between the FBI, local enforcement agencies, and private organizations. It has created Special Interest Groups (SIGs), which are resources dedicated to the safeguarding of specific critical infrastructures of both private industry and government through information-sharing networks and a private secure portal of communication. You can obtain more information about Infragard and local chapters at http://www.infragard.net

If you work in the health care industry, you should be aware of several new regulations, such as the Health Industry Portability and Accountability Act (HIPAA). The act requires all persons with access to this information to take reasonable care to protect the integrity and confidentiality of patient data. Not only hospitals and health care facilities, but also insurers are now implementing security safeguards and completing risk assessments to ensure the privacy of patients.

Security Incident Mitigation Tools

This section includes several tools and techniques that you can use when mitigating security incidents, such as DDoS and worm outbreaks.

TIP The mitigation technique and enforcement depends on your network architecture and design. This section covers the most common techniques. As a rule of thumb, you want to base your mitigation operations as close as possible to the source of the attack.

Access Control Lists (ACL)

When you react to a DDoS or to a worm outbreak, one of the most important matters is how fast you can quarantine and isolate the problem. *Quarantining* is the process of identifying all infected machines and blocking them from the network to prevent them from infecting other systems (in case of a worm outbreak). The easiest way to quarantine or block systems is by using router and firewall access control lists (ACL) and VLAN ACLs (or VACL) on Cisco switches. VACLs allow port-level filtering on a VLAN basis. In most cases, VACLs are more feasible when blocking an infected machine. VACLs are applied directly on the switch port, thereby enabling you to do per-host filtering.

It is extremely important that you be familiar with your network topology and understand how all the VLANs are configured. It is a best practice to document the devices (or at least the device types) that reside within each VLAN. This will be extremely helpful to you when you are in the mitigating phase of your reaction to attacks and worm outbreaks.

Another best practice is to prioritize your network resources and critical systems. During the reaction phase, you should protect the most critical systems first.

For more information on tools that can be used for asset management and asset classification, see Chapter 7, “Proactive Security Framework.”

TIP The Cisco Catalyst 6000 series of switches has a switching engine known as a Policy Feature Card (PFC) that contains specialized application-specific integrated circuits (ASIC) that enable the blocking of traffic to occur at close to wire speed on the switch.

One of the major problems with ACLs and VACLs is that you must apply them throughout the network quickly. You can use tools such as the Cisco Security Manager (CSM) to deploy ACLs quickly in your network. You can also use commercial tools such as OpsWare, and SolSoft.

Many security administrators allocate a range of extended ACL numbers that can be dynamically used when mitigating security incidents. For instance, you can assign 190 to 199 for security reaction ACLs, if this range is not in use anywhere else in your network. Some people recommend configuring, on each network, a dummy list device which is well documented with a detailed description so that staff will know that this ACL is reserved

and will know its purpose. If you have NetConfig, you can create templates to ease the deployment.

Private VLANs

Private VLANs can be used to achieve Layer 2 isolation of hosts within a VLAN. Some people use private VLANs in their data center to isolate servers in case they are compromised or infected. However, private VLANs do not provide perfect isolation. For example, you can insert a Layer 3 device to a promiscuous port and hop from one system to another using the destination IP address with the Layer 3 device MAC address. This type of attack and others are explained extensively in the whitepaper at http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns128/networking_solutions_white_paper09186a008014870f.shtml#wp1002364.

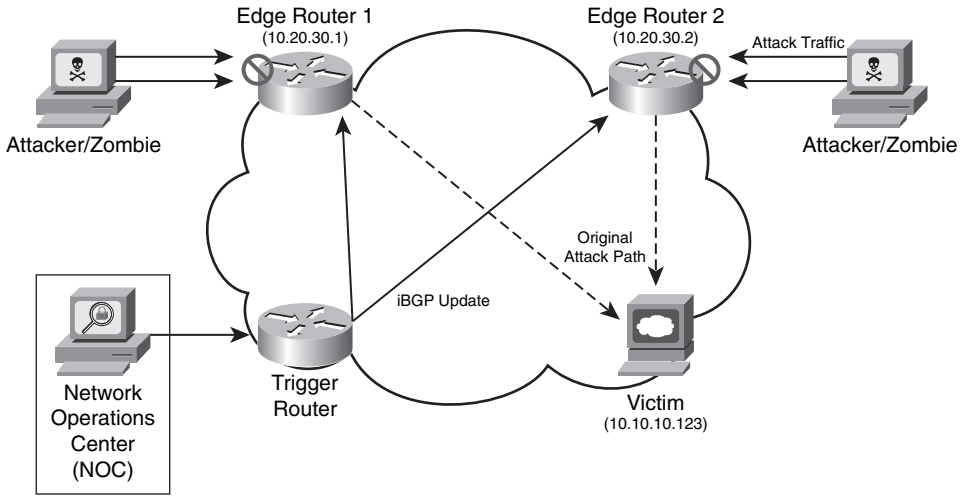
Remotely Triggered Black Hole Routing

Remotely triggered black hole (RTHB) routing is a technique that can be used to drop all attack traffic based on either destination or attack source addresses. Source and destination-based RTBH filter undesirable traffic by forwarding it to the Null0 interface (a pseudointerface that is always up and can never forward or receive traffic). Performance is not a significant challenge with RTBH because it occurs directly in the forwarding path or Cisco Express Forwarding (CEF).

NOTE This section assumes that you have a basic understanding of Border Gateway Protocol (BGP). If you need to review BGP, refer to http://www.cisco.com/en/US/tech/tk365/tk80/tsd_technology_support_sub-protocol_home.html which includes a comprehensive list of BGP-related FAQs, configuration guidelines, and troubleshooting tips.

Destination-based RTBH works by filtering traffic destined to the hosts being attacked or by filtering an infected host (in worm outbreaks) at the boundary closest to the source. The trigger is typically a router that sends a routing update (iBGP in most cases) to other edge routers configured for black hole filtering. The trigger sends an update with the next-hop IP address defined in a static route pointing to Null0. This is illustrated in Figure 5-1.

Figure 5-1 *Destination-Based RTBH*



In Figure 5-1, two zombies are attacking a web server (10.10.10.123). The network administrator in the Network Operations Center (NOC) notices the attack and configures a static route on the trigger router with the destination host address (10.10.10.123), pointing it to Null0. This trigger router then sends an iBGP update to the two other routers causing it to drop the attack traffic. Example 5-1 is the trigger router configuration:

Example 5-1 *Trigger Router Configuration*

```
interface loopback0
 ip address 10.20.30.18 255.255.255.255
!
interface Null0
 no ip unreachables
!
router bgp 64555
 no synchronization
 no bgp client-to-client reflection
 bgp log-neighbor-changes
 redistribute static route-map rtbh-trigger

 neighbor rtbh-group peer-group
 neighbor rtbh-group remote-as 64555
 neighbor rtbh-group update-source loopback0
 neighbor rtbh-group route-reflector-client
 neighbor 10.20.30.1 peer-group rtbh-group
!
route-map rtbh-trigger permit 10
 match tag 666
 set ip next-hop 192.168.20.1
```

continues

Example 5-1 *Trigger Router Configuration (Continued)*

```
 set local-preference 200
 set origin igp
 set community no-export
route-map rtbh-trigger deny 20
! The following is the static route that drops the traffic from the infected machine
ip route 10.10.10.123 255.255.255.255 Null0 tag 666
```

In the previous configuration example, a static route for the IP address (10.10.10.123) of the victim is configured pointing to Null0 and with a tag of 666. A route map called rtbh-trigger is applied prior to redistributing the static route into BGP. This route map is configured to match on a tag value of 666. It also sets the next-hop to 192.168.20.1 which is an unused address space that you must configure to selectively drop the traffic. The trigger router sets the next-hop route for the destination IP address whose traffic will be dropped. Route updates are used to propagate this route to all iBGP peer routers. These routers then set their next-hop to the destination. You must configure a static route for the next-hop address (in this example, 192.168.20.1) pointing to Null0 in all the routers where you want the traffic to be dropped. This enables the edge routers to set their next-hops accordingly and forward all traffic for the black-holed destination IP address to Null0. In this example, the local preference is set to 200, and the origin is set to the remote Interior Gateway Protocol (IGP) system. The community is set to no-export, so these routes will not be advertised to external BGP (eBGP) peers.

NOTE For RTBH to operate successfully, the trigger router must have an iBGP peering relationship with the other two routers. If you use BGP route reflectors, the trigger router must have an iBGP relationship with the route reflectors in every cluster.

If the attacker uses nonspoofed addresses for the attack, you can also do source-based RTBH just by adding a static route to the source or source network, as shown in the following example.

```
ip route 192.168.20.2 255.255.255.255 Null0 tag 666
```

In this example, the attacker is using the IP address 192.168.20.2. However, an attacker could target a legitimate IP address by spoofing it as the source of an attack and counting on you to black-hole the source using sourced-based RTBH filtering. This is why having antispoofing mechanisms in place is crucial for every network in any organization.

Forensics

Many people say that computer forensics is similar to a crime scene investigation, in most cases, the security event you are investigating may be an actual crime. You should determine which computer forensic methodology is most appropriate for your organization.

This investigation can be done by you or your own staff, by law enforcement, or by private sector computer forensic specialists. One of the most critical items to remember is the consequences of mishandling evidence. Forensics is a broad topic, and the laws and handling of evidence vary based on your locality. This chapter is intended to give you only some of the common tools and mechanisms that you can use to perform basic forensics after a security event.

NOTE References to several whitepapers and tools are listed in the sections that follow.

Log Files

After a security incident, you can use log files to obtain clues on what happened. However, logs are useful only if they are actually read. Even in small networks, logs from servers, networking devices, end-host machines, and other systems can be large, and their analysis may be tedious and time consuming. That is why it is important to use event correlation systems and other tools to better analyze and study log entries. You can use robust systems such as CS-MARS or even simple tools and programs such as Swatch. *Swatch* stands for Simple Watcher. It is an open source tool written in Perl that is capable of searching a file for a list of strings and then performing specific actions when such a string is found. Swatch was designed to do real-time monitoring of server log files; however, you can also use it to handle a standalone file. It was also designed to analyze syslog archives, but you can use it on any file.

NOTE The Swatch open source project is maintained on Source Forge at http://swatch. sourceforge.net.

Another excellent tool is Splunk. You can use this tool to conduct real-time searches of different types of event logs from different systems.

NOTE For more information about Splunk, go to http://www.splunk.com. In addition, http://www.loganalysis.org includes information about numerous log parsers that can be used for forensic purposes.

Different systems have different log formats. If it is necessary to compare files, it can be challenging to match up fields. For example, logs from routers are not the same as logs from

firewalls or other networking devices. Similarly, logs from Linux or UNIX servers are not the same as logs from Windows systems. CS-MARS can help you analyze all these different types of logs. Also, some open source tools can help you analyze system logs from UNIX/Linux and Windows machines. The following sections include the most commonly used tools.

Linux Forensics Tools

Two of the most commonly used Linux forensics tools are Autopsy and the Sleuth Kit. These programs are intuitive and are a compilation of the following:

- File system layer tools
- File system journal tools
- Meta data layer tools
- Disk image file tools

Despite the fact that Autopsy and the Sleuth Kit run on Linux, they support the NTFS, FAT, Ext2/3, and UFS1/2 file systems. You can download Autopsy and the Sleuth Kit free from http://www.sleuthkit.org.

Figure 5-2 is a screen shot of Autopsy.

Figure 5-2 *Autopsy Linux Forensics Tool*

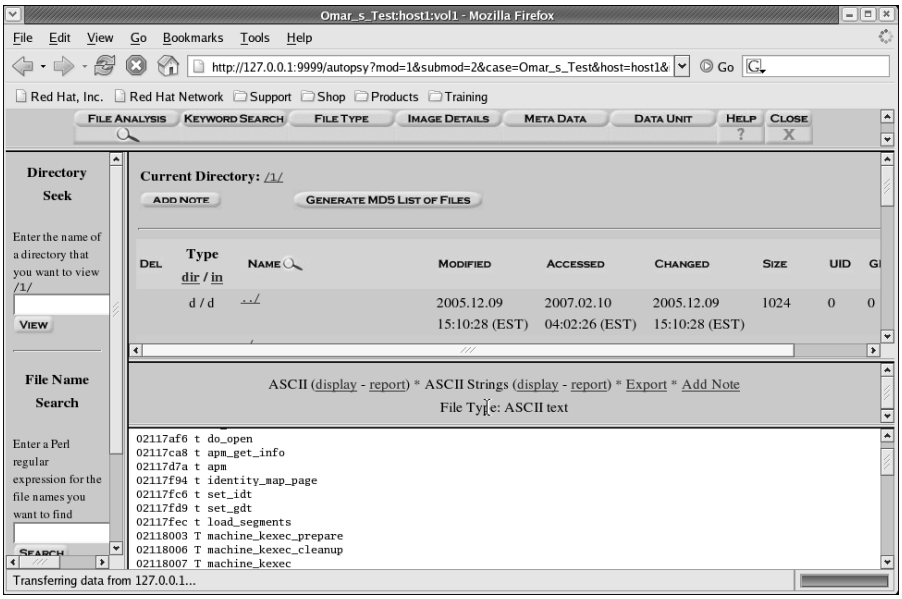
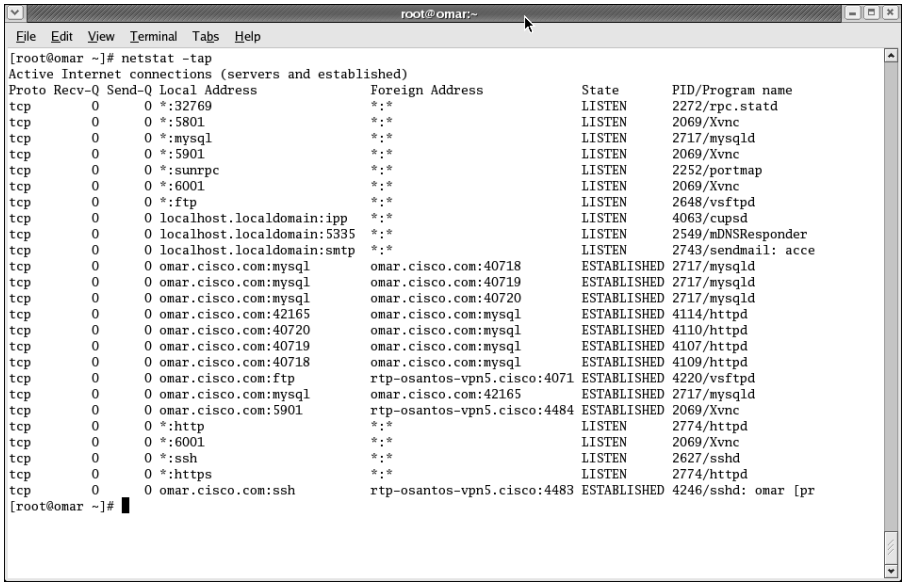


Figure 5-2 shows how you can use Autopsy to analyze the files and directories within a system. You can use this tool to see the names of deleted files. Autopsy can create timelines that contain entries for the "Modified, Access, and Change" times of both allocated and unallocated files. It also allows you to create a "case" to track each security incident.

When collecting information from a Linux or UNIX-based system, you can also use simple tools and commands such as **netstat** and **pstree**. You can use the **netstat -tap** command as shown in Figure 5-3 to obtain information about the active connections in a system.

Figure 5-3 **netstat** *Command Output*

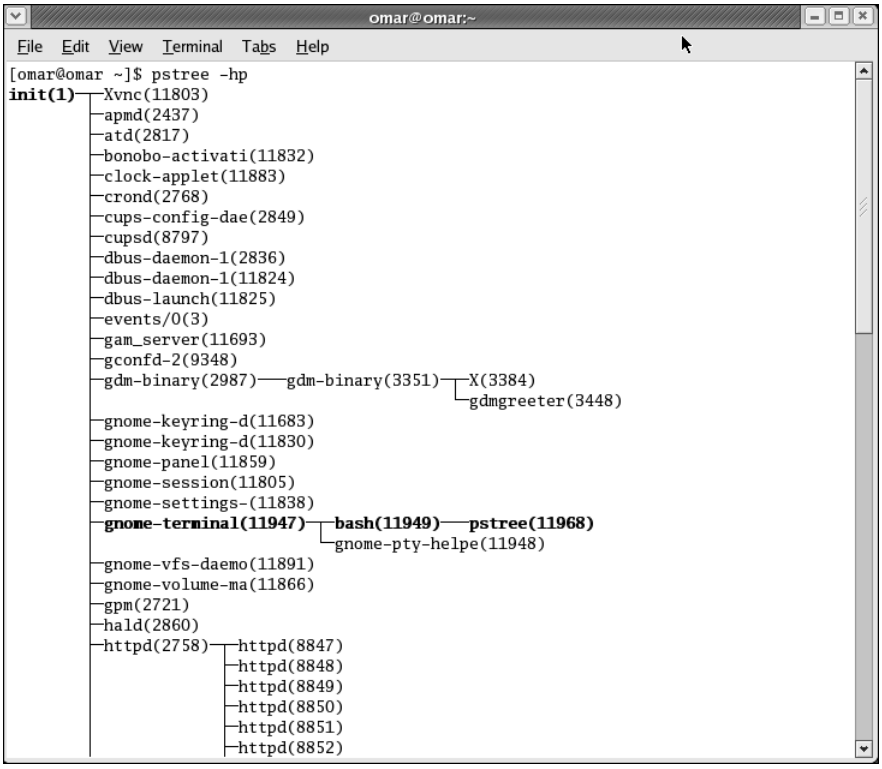


In Figure 5-3, you can see the output showing the different established connections on the system.

NOTE On UNIX- and Linux-based systems (including Mac OS X), use the **man netstat** command to obtain detailed documentation on the available options of the **netstat** command.

You can also use the **pstree** utility on a Linux system to display the processes on the system in the form of a tree diagram. This allows you to have a better view of the processes running on the system that may be part of malicious software. Figure 5-4 includes a screen shot of the output of the **pstree -hp** command. The **-h** option is used to show the current process and its ancestors, and the -**p** option is used to display the process IDs (PID).

Figure 5-4 **pstree** *Command Output*



The detailed whitepaper titled "Checking UNIX/LINUX Systems for Signs of Compromise" supplies insightful information on the forensics of Linux and UNIX systems. You can download the whitepaper from http://www.ucl.ac.uk/cert/nix_intrusion.pdf.

Windows Forensics

The most commonly used toolkit for forensics in Windows-based systems is Systernals. *Systernals* is a compilation of several tools used for analysis, troubleshooting, and forensics of Windows machines. This toolkit was initially created by Mark Russinovich and Bryce Cogswell, and Microsoft acquired it in July 2006. Systernals toolkit includes the following:

- File and disk utilities
- Network statistical and analysis utilities
- Process illustration and analysis utilities
- Security configuration utilities
- System resource usage and configuration tools

NOTE Microsoft has an excellent whitepaper about Windows forensics best practices and methodologies at http://www.microsoft.com/technet/security/guidance/disasterrecovery/computer_investigation/default.mspx.

Guidance Software also develops sophisticated forensics tools. Its EnCase product suite includes different integrated tools that facilitate seamless sharing of evidentiary data and solve the resource drain of encrypted data.

NOTE For more information about the EnCase suite of tools, go to http://www.guidancesoftware.com.

It is important to remember that no matter the vendor, the forensics tool you select must give you flexibility when conducting investigations and should help mask complexity when forensics data is shared with untrained individuals.

Summary

In this chapter, you learned how important it is for any organization to have adequate incident handling policies and procedures. You also learned general information about the different laws and practices involved when you are investigating security incidents and computer crimes. This chapter also included detailed information about different tools you can use to mitigate attacks and other security incidents with your network infrastructure components. This chapter concluded with a discussion of basic computer forensics topics.