Special Publication 800-61

Computer Security Incident Handling Guide

Special Publication 800-86

Guide to Integrating Forensic Techniques into Incident Response

Recommendations of the
National Institute of Standards and Technology

Abridged by Guidance Software, Inc.

Guidance™ Software

Special Publication 800-61

Computer Security Incident Handling Guide

Special Publication 800-86

Guide to Integrating Forensic Techniques into Incident Response

Abridged by Guidance Software, Inc.

Recommendations of the National Institute of Standards and Technology

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), "Securing Agency Information Systems."

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority, nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

This guideline should not be held as binding to law enforcement personnel relative to the investigation of criminal activity.

Forward

The Federal Information Security Management Act (FISMA) of 2002 mandates that federal agencies must establish incident response capabilities consistent with the guidelines and standards established by the National Institute of Standards and Technology (NIST). Pursuant to this mandate, NIST issued Special Publication 800-61 *Computer Security Incident Handling Guide*, which sets forth detailed technical, procedural and policy guidelines for federal agencies to implement a comprehensive incident response program, however the document did little to define the exact steps involved in the actual investigation and resolution of a given incident.

In response to the high-level language of 800-61, in August of 2006, NIST published SP800-86 *Guide to Integrating Forensic Techniques into Incident Response*. Here, NIST defines in a much more precise and specific way the procedures, issues and technologies required to move an incident from the point of discovery all the way through to resolution.

Together, these documents are now a fact for federal civilian agencies and with it so is the requirement that organizations augment their incident response capabilities by developing a complete forensics program. In addition to acquiring appropriate forensic tools, FISMA compliance mandates extensive and ongoing training and the development of clear policies and procedures.

As the leading provider of forensic software, investigative solutions, services and training, Guidance Software is ideally positioned to help guide organizations through this process to ensure efficient and intelligent implementation of a system that not only enables compliance, but also allows you to radically reduce costs and optimize network security (see figure 1).

Its EnCase® software delivers unmatched forensic investigation, incident response and auditing capabilities. This technology coupled with Guidance Software's extensive investment in human capital allows Guidance Software to provide comprehensive services in all of the key areas necessary for compliance with NIST 800-61 and 800-86.

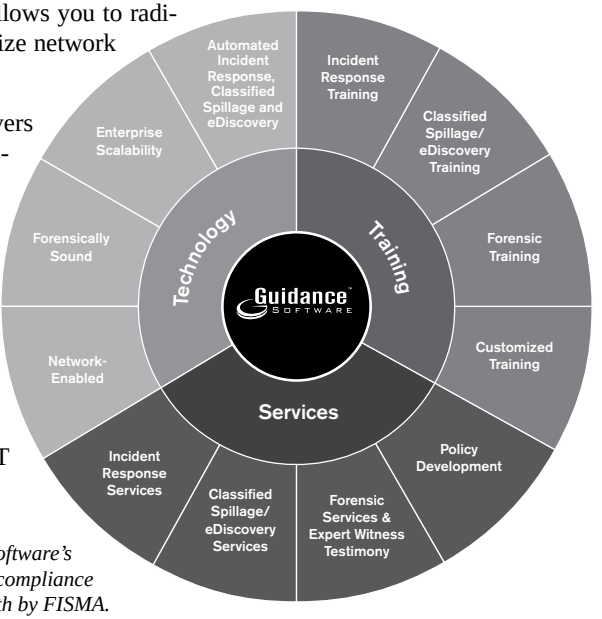

Figure 1. Together, Guidance Software's technology and services ensure compliance with the requirements set forth by FISMA.

Table of Contents

Section 1 – Special Publication 800-61

Computer Security Incident Handling Guide

Recommendations of the
National Institute of Standards and Technology

Tim Grance
Karen Kent
Brian Kim

January 2004
Abridged by Guidance Software, Inc.

This publication seeks to help both established and newly formed incident response teams. This document assists organizations in establishing computer security incident response capabilities and handling incidents efficiently and effectively.

Implementing the following requirements and recommendations should facilitate efficient and effective incident response for Federal departments and agencies.

Organizations must create, provision, and operate a formal incident response capability. Federal law requires Federal agencies to report incidents to the Federal Computer Incident Response Center (FedCIRC) office within the Department of Homeland Security.

The Federal Information Security Management Act (FISMA) of 2002 requires Federal agencies to establish incident response capabilities. Each Federal civilian agency must designate a primary and secondary point of contact (POC) with FedCIRC, report all incidents, and internally document corrective actions and their impact. Each agency is responsible for determining specific ways in which these requirements are to be fulfilled.

Establishing an incident response capability should include the following actions:

- Creating an incident response policy
- Developing procedures for performing incident handling and reporting, based on the incident response policy
- Setting guidelines for communicating with outside parties regarding incidents
- Selecting a team structure and staffing model
- Establishing relationships between the incident response team and other groups, both internal (e.g., legal department) and external (e.g., law enforcement agencies)
- Determining what services the incident response team should provide
- Staffing and training the incident response team.

Organizations should reduce the frequency of incidents by effectively securing networks, systems, and applications.

Organizations should document their guidelines for interactions with other organizations regarding incidents.

Organizations should emphasize the importance of incident detection and analysis throughout the organization.

Organizations should create written guidelines for prioritizing incidents.

Organizations should use the lessons learned process to gain value from incidents.

Organizations should strive to maintain situational awareness during large-scale incidents.

1. Introduction

1.1 Authority

(Refer to the prefix of this guide)

1.2 Purpose and Scope

This document presents general incident response guidelines that are independent of particular hardware platforms, operating systems, and applications. Specifically, it includes guidance on establishing an effective incident response program, but the primary focus of the document is detecting, analyzing, prioritizing, and handling incidents.

1.3 Audience

This document has been created for computer security incident response teams (CSIRTs), system and network administrators, security staff, technical support staff, chief information officers (CIOs), and computer security program managers who are responsible for preparing for, or responding to, security incidents.

2. Organizing A Computer Security Incident Response Capability

Organizing an effective computer security incident response capability (CSIRC) involves several major decisions and actions. One of the first considerations should be to create an organization-specific definition of the term "incident" so that the scope of the term is clear. The organization should decide what services the incident response team should provide, consider which team structures and models can provide those services, and select and implement one or more incident response teams. Incident response policy and procedure creation is an important part of establishing a team, so that incident response is performed effectively, efficiently, and consistently.

2.1 Events and Incidents

An event is any observable occurrence in a system or network. Events include a user connecting to a file share, a server receiving a request for a Web page, a user sending electronic mail (e-mail), and a firewall blocking a connection attempt. Adverse events are events with a negative consequence, such as system crashes, network packet floods, unauthorized use of system privileges, defacement of a Web page, and execution of malicious code that destroys data. This guide addresses only adverse events that are computer security-related and excludes adverse events caused by sources such as natural disasters and power failures.

2.2 Need for Incident Response

Incident response has become necessary because attacks frequently cause the compromise of personal and business data. Malicious code incidents such as the SQL Slammer worm, the Blaster worm, and the Love Letter worm have disrupted or damaged millions of systems and networks around the world. Heightened national security concerns are also raising awareness of the possible effects of computer-based attacks.

These events—and many more—make the case daily for responding quickly and efficiently when computer security defenses are breached. To address these threats, the concept of computer security incident response has become widely accepted and implemented in the Federal Government, private sector, and academia.

The following are benefits of having an incident response capability:

- Responding to incidents systematically so that the appropriate steps are taken
- Helping personnel to recover quickly and efficiently from security incidents, minimizing loss or theft of information, and disruption of services
- Using information gained during incident handling to better prepare for handling future incidents and to provide stronger protection for systems and data
- Dealing properly with legal issues that may arise during incidents.

Besides the business reasons to establish an incident response capability, Federal departments and agencies must comply with law, regulations, and policy directing a coordinated, effective defense against information security threats. Chief among these are the following:

- OMB's Circular No. A-130, Appendix III, which directs Federal agencies to "ensure that there is a capability to provide help to users when a security incident occurs in the system and to share information concerning common vulnerabilities and threats. This capability shall share information with other organizations … and should assist the agency in pursuing appropriate legal action, consistent with Department of Justice guidance."
- FISMA, which requires agencies to have "procedures for detecting, reporting, and responding to security incidents" and establishes a centralized Federal information security incident center, in part to:
  - "Provide timely technical assistance to operators of agency information systems … including guidance on detecting and handling information security incidents …
  - Compile and analyze information about incidents that threaten information security …
  - Inform operators of agency information systems about current and potential information security threats, and vulnerabilities … ."

2.3 Incident Response Policy and Procedure Creation

This section discusses policies and procedures related to incident response, with an emphasis on interactions with outside parties, such as the media, law enforcement agencies, and incident reporting organizations.

2.3.1 Policy and Procedure Elements

Policy governing incident response is highly individualized to the organization. However, most policies include the same key elements, regardless of whether the organization's incident response capability is indigenous or outsourced:

- Statement of management commitment
- Purpose and objectives of the policy
- Scope of the policy (to whom and what it applies and under what circumstances)
- Definition of computer security incidents and their consequences within the context of the organization
- Organizational structure and delineation of roles, responsibilities, and levels of authority; should include the authority of the incident response team to confiscate or disconnect equipment and to monitor suspicious activity, and the requirements for reporting certain types of incidents
- Prioritization or severity ratings of incidents
- Performance measures (as discussed in Section 3.4.2)
- Reporting and contact forms.

Procedures should be based on the incident response policy. Standard operating procedures (SOPs) are a delineation of the specific technical processes, techniques, checklists, and forms used by the incident response team. SOPs should be comprehensive and detailed to ensure that the priorities of the organization are reflected in response operations. In addition, following standardized responses should minimize errors, particularly those that might be caused by incident handling tempo and stress. SOPs should be tested to validate their accuracy and usefulness, then distributed to all team members. Training should be provided for SOP users; the SOP documents can be used as an instructional tool. Suggested SOP elements are presented throughout Sections 3 through 8.

2.3.2 Sharing Information With Outside Parties

An organization may want to—or be required to—communicate incident details with an outside organization for numerous reasons. The incident response team should discuss this at length with the organization's public affairs office, legal department, and management before an incident occurs to establish policies and procedures regarding information sharing. The team should document all contacts and communications with outside parties for liability and evidentiary purposes.

Figure 2-1. Incident-Related Communications With Outside Parties

2.3.2.1 The Media

Dealing with the media is an important part of incident response. The following actions should be considered for preparing those who may be communicating with the media:

- Conduct training sessions on interacting with the media regarding incidents, which should include:
  - The importance of not revealing sensitive information, such as technical details of countermeasures (e.g., which protocols the firewall permits), which could assist other would-be attackers
  - The positive aspects of communicating important information to the public fully and effectively.
- Establish procedures to brief media contacts on the issues and sensitivities regarding a particular incident before discussing it with the media.
- Hold mock interviews and press conferences during incident handling exercises. The following are examples of questions to ask the media contact:
  - Who attacked you?
  - When did it happen?
  - How did they do the attack?
  - How widespread is this incident?
  - Did this happen because you have poor security practices?
  - What steps are you taking to determine what happened?
  - What is the impact of this incident?
  - What is the estimated monetary cost of this incident?

2.3.2.2 Law Enforcement

One reason that many security-related incidents do not result in convictions is that organizations do not properly contact law enforcement. Several levels of law enforcement are available to investigate incidents: The incident response team should become acquainted with its various law enforcement representatives before an incident occurs to discuss conditions under which incidents should be reported to them, how the reporting should be performed, what evidence should be collected, and how it should be collected.

Note that the organization typically should not contact multiple agencies because doing so might result in jurisdictional conflicts. The incident response team should understand what the potential jurisdictional issues are (e.g., physical location-an organization based in one state has a server located in a second state attacked from a system in a third state, being used remotely by an attacker in a fourth state).

2.3.2.3 Incident Reporting Organizations

FISMA requires Federal agencies to report incidents to FedCIRC,12 which is a governmentwide incident response capability that assists Federal civilian agencies in their incident handling efforts. FedCIRC does not replace existing agency response teams; rather, it augments the efforts of Federal civilian agencies by serving as a focal point for dealing with incidents. FedCIRC analyzes the information provided by all agencies to identify trends and precursors of attacks; these are easier to discern when reviewing data from many organizations than when reviewing the data of a single organization.

Each agency must designate a primary and secondary POC with FedCIRC, report all incidents, and internally document corrective actions and their impact. All Federal agencies must ensure that their incident response procedures adhere to FedCIRC's reporting requirements and that the procedures are followed properly. This is not only mandatory for Federal agencies, but also beneficial for them because FedCIRC can provide better information to agencies if they receive better incident data from them.

Organizations other than Federal civilian agencies should not report incidents to FedCIRC, unless the incidents affect Federal agencies. If an organization does not have its own CSIRT to contact, it can report incidents to other organizations, including:

- Information Analysis Infrastructure Protection (IAIP).
- CERT® Coordination Center (CERT®/CC).
- Information Sharing and Analysis Centers (ISAC).

2.3.2.4 Other Outside Parties

As previously mentioned, an organization may want to discuss incidents with several other groups, including:

- The Organization's ISP.
- Owners of Attacking Addresses. Handlers should be cautious if they are unfamiliar with the external organization because the owner of the address space could be the attacker or an associate of the attacker.
- Software Vendors.
- Other Incident Response Teams.
- Affected External Parties.

2.4 Incident Response Team Structure

An incident response team should be available for contact by anyone who discovers or suspects that an incident involving the organization has occurred. One or more team members, depending on the magnitude of the incident and availability of personnel, will then handle the incident. The incident handlers analyze the incident data, determine the impact of the incident, and act appropriately to limit the damage

to the organization and restore normal services. Although the incident response team may have only a few members, the team's success depends on the participation and cooperation of individuals throughout the organization. This section identifies such individuals, discusses incident response team models, and provides guidance for selecting an appropriate model.

2.4.1 Team Models

Incident response team structure models fall into one of three categories:

- **Central Incident Response Team**. A single incident response team handles incidents throughout the organization. This model is effective for small organizations and for large organizations with minimal geographic diversity in terms of computing resources.
- **Distributed Incident Response Teams**. The organization has multiple incident response teams, each responsible for handling incidents for a particular logical or physical segment of the organization. This model is effective for large organizations (e.g., one team per division) and for organizations with major computing resources at distant locations (e.g., one team per geographic region, one team per major facility). However, the teams should be part of a single centralized entity so that the incident response process is consistent across the organization and information is shared among teams.
- **Coordinating Team**. An incident response team provides guidance and advice to other teams without having authority over those teams

Incident response teams can also use any of three staffing models:

- **Employees**. The organization performs all of its incident response work, with limited technical and administrative support from contractors.
- **Partially Outsourced**. The organization outsources portions of its incident response work.
- **Fully Outsourced**. The organization completely outsources its incident response work, typically to an onsite contractor.

2.4.2 Team Model Selection

When selecting appropriate structure and staffing models for an incident response team, organizations should consider the following factors:

- **The Need for 24/7 Availability.** Real-time availability is the best for incident response because the longer an incident lasts, the more potential there is for damage and loss. Real-time contact is often needed when working with other agencies and organizations—for example, tracing spoofed traffic back to its source through router hops. An incident response team that can react quickly to investigate, contain, and mitigate incidents should be genuinely useful to the organization.
- **Full-Time Versus Part-Time Team Members.** Organizations with limited funding, staffing, or incident response needs may have only part-time incident

response team members. Organizations with part-time team members should ensure that they maintain their incident response skills and knowledge.

- **Employee Morale.** Incident response work is very stressful, as are the on-call responsibilities of most team members. This combination makes it easy for incident response team members to become overly stressed. Many organizations will also struggle to find willing, available, experienced, and properly skilled people to participate, particularly in 24-hour support.
- **Cost.** Cost is a major factor, especially if employees are required to be onsite 24/7. Organizations may fail to include incident response-specific costs in budgets.
- **Staff Expertise.** Incident handling requires specialized knowledge and experience in several technical areas; the breadth and depth of knowledge required varies based on the severity of the organization's risks. Outsourcers may possess deeper knowledge of intrusion detection, vulnerabilities, exploits, and other aspects of security than employees of the organization. However, technical staff members within the organization usually have much better knowledge of the organization's environment than an outsourcer would, which can be beneficial in identifying false positives associated with organization-specific behavior and the criticality of targets.
- **Organizational Structures.** If an organization has three departments that function independently, incident response may be more effective if each department has its own incident response team. The main organization can host a centralized incident response entity that facilitates standard practices and communications among the teams.

When considering outsourcing, organizations should keep these issues in mind:

- Current and Future Quality of Work.
- Division of Responsibilities.
- Sensitive Information Revealed to the Contractor.
- Lack of Organization-Specific Knowledge.
- Lack of Correlation.
- Handling Incidents at Multiple Locations.
- Maintaining Incident Response Skills In House.

2.4.3 Incident Response Personnel

Regardless of which incident response model an organization chooses, a single employee should be in charge of incident response. In a fully outsourced model, this person is responsible for overseeing and evaluating the outsourcer's work.

Members of the incident response team should have excellent technical skills because they are critical to the team's success.

It is important to counteract staff burnout by providing opportunities for learning and growth.

2.4.4 Dependencies Within Organizations

It is important to identify other groups within the organization that may be needed to participate in incident handling so that their cooperation can be solicited before it is needed. Every incident response team relies on the expertise, judgment, and abilities of others, including:

- Management.
- Information Security.
- Telecommunications.
- IT Support.
- Legal Department.
- Public Affairs and Media Relations.
- Human Resources.
- Business Continuity Planning.
- Physical Security and Facilities Management.

2.5 Incident Response Team Services

The main focus of an incident response team is performing incident response; however, it is fairly rare for a team to perform incident response only. The following are examples of additional services that an incident response team might offer:

- Vulnerability Assessment.
- Intrusion Detection.
- Education and Awareness.
- Technology Watch.
- Patch Management.

2.6 Recommendations

The key recommendations presented in this section for organizing a computer security incident handling capability are summarized below.

- **Establish a formal incident response capability**. Organizations should be prepared to respond quickly and effectively when computer security defenses are breached. FISMA requires Federal agencies to establish incident response capabilities.
- **Create an incident response policy and use it as the basis for incident response procedures.** The incident response policy is the foundation of the incident response program. It defines which events are considered incidents, establishes the organizational structure for incident response, defines roles and responsibilities, and lists the requirements for reporting incidents, among other items.

- **Establish policies and procedures regarding incident-related information sharing.** The organization will want or be required to communicate incident details with outside parties, such as the media, law enforcement agencies, and incident reporting organizations. The incident response team should discuss this requirement at length with the organization's public affairs office, legal department, and management to establish policies and procedures regarding information sharing. The team should comply with existing organization policy on interacting with the media and other outside parties.
- **Provide pertinent information on incidents to the appropriate incident reporting organization.** Federal civilian agencies are required to report incidents to FedCIRC; other organizations can contact other incident reporting organizations. Reporting is beneficial because the incident reporting organizations use the reported data to provide information to the reporting parties regarding new threats and incident trends.
- **Consider the relevant factors when selecting an incident response team model.** Organizations should carefully weigh the advantages and disadvantages of each possible team structure model and staffing model in the context of the organization's needs and available resources.
- **Select people with appropriate skills for the incident response team.** The credibility and proficiency of the team depend to a large extent on the technical skills of its members. Poor technical judgment can undermine the team's credibility and cause incidents to worsen. Critical technical skills include system administration, network administration, programming, technical support, and intrusion detection. Teamwork and communications skills are also needed for effective incident handling.
- **Identify other groups within the organization that may need to participate in incident handling.** Every incident response team relies on the expertise, judgment, and abilities of other teams, including management, information security, IT support, legal, public affairs, and facilities management.
- **Determine which services the team should offer.** Although the main focus of the team is incident response, most teams perform additional functions. Examples include distributing security advisories, performing vulnerability assessments, educating users on security, and monitoring intrusion detection sensors.

3. Handling an Incident

The incident response process has several phases, from initial preparation through post-incident analysis.

Figure 3-1. Incident Response Life Cycle

3.1 Preparation

Incident response methodologies typically emphasize preparation—not only establishing an incident response capability so that the organization is ready to respond to incidents, but also preventing incidents by ensuring that systems, networks, and applications are sufficiently secure. Although the incident response team is not typically responsible for incident prevention, it is so important that it is now considered a fundamental component of incident response programs.

3.1.1 Preparing to Handle Incidents

Table 3-1. Tools and Resources for Incident Handlers

Acquired	Tools/Resource
Incident Handler Communications and Facilities	
	Contact information for team members and others within and outside the organization (primary and backup contacts), such as law enforcement and other incident response teams; information may include phone numbers, e-mail addresses, public encryption keys (in accordance with the encryption software described below), and instructions for verifying the contact's identity
	On-call information for other teams within the organization, including escalation information (see Section 3.2.6 for more information about escalation)
	Incident reporting mechanisms, such as phone numbers, e-mail addresses, and online forms that users can use to report suspected incidents; at least one mechanism should permit people to report incidents anonymously
	Pagers or cell phones to be carried by team members for off-hour support, onsite communications
	Encryption software to be used for communications among team members, within the organization and with external parties; software must use a Federal Information Processing Standards (FIPS) 140-2 validated encryption algorithm
	War room for central communication and coordination; if a permanent war room is not necessary, the team should create a procedure for procuring a temporary war room when needed
	Secure storage facility for securing evidence and other sensitive materials

Acquired	Tools/Resource
	Incident Analysis Hardware and Software
	Computer forensic workstations and/or backup devices to create disk images, preserve log files, and save other relevant incident data
	Laptops, which provide easily portable workstations for activities such as analyzing data, sniffing packets, and writing reports
	Spare workstations, servers, and networking equipment, which may be used for many purposes, such as restoring backups and trying out malicious code; if the team cannot justify the expense of additional equipment, perhaps equipment in an existing test lab could be used, or a virtual lab could be established using operating system (OS) emulation software
	Blank media, such as floppy diskettes, CD-Rs, and DVD-Rs
	Easily portable printer to print copies of log files and other evidence from non-networked systems
	Packet sniffers and protocol analyzers to capture and analyze network traffic that may contain evidence of an incident
	Computer forensic software to analyze disk images for evidence of an incident
	Floppies and CDs with trusted versions of programs to be used to gather evidence from systems
	Evidence gathering accessories, including hard-bound notebooks, digital cameras, audio recorders, chain of custody forms, evidence storage bags and tags, and evidence tape, to preserve evidence for possible legal actions
	Incident Analysis Resources
	Port lists, including commonly used ports and Trojan horse ports
	Documentation for OSs, applications, protocols, and intrusion detection and antivirus signatures
	Network diagrams and lists of critical assets, such as Web, e-mail, and File Transfer Protocol (FTP) servers
	Baselines of expected network, system and application activity
	Cryptographic hashes of critical files33 to speed the analysis, verification, and eradication of incidents
	Incident Mitigation Software
	Media, including OS boot disks and CD-ROMs, OS media, and application media
	Security patches from OS and application vendors
	Backup images of OS, applications, and data stored on secondary media

Many incident response teams create a jump kit, which is a portable bag or case that contains materials that an incident handler may likely need during an offsite investigation. For example, each jump kit typically includes a laptop, loaded with appropriate software (e.g., packet sniffers, computer forensics). Other important materials include backup devices, blank media, basic networking equipment and cables, and operating system and application media and patches.

3.1.2 Preventing Incidents

It is outside the scope of this document to provide specific advice on securing networks, systems, and applications.

3.2 Detection and Analysis

Figure 3-2. Incident Response Life Cycle (Detection and Analysis)

3.2.1 Incident Categories

Incidents can occur in countless ways, so it is impractical to develop comprehensive procedures with step-by-step instructions for handling every incident. The best that the organization can do is to prepare generally to handle any type of incident and more specifically to handle common incident types.

3.2.2 Signs of an Incident

For many organizations, the most challenging part of the incident response process is accurately detecting and assessing possible incidents—determining whether an incident has occurred and, if so, the type, extent, and magnitude of the problem. What makes this so challenging is a combination of three factors:

- Incidents may be detected through many different means, with varying levels of detail and fidelity.
- The volume of potential signs of incidents is typically high; for example, it is not uncommon for an organization to receive thousands or even millions of intrusion detection sensor alerts per day.
- Deep, specialized technical knowledge and extensive experience are necessary for proper and efficient analysis of incident-related data.

Signs of an incident fall into one of two categories: indications and precursors. A precursor is a sign that an incident may occur in the future. An indication is a sign that an incident may have occurred or may be occurring now. Too many types of indications exist to exhaustively list them, but some examples are listed below:

- The network intrusion detection sensor alerts when a buffer overflow attempt occurs against an FTP server.
- The antivirus software alerts when it detects that a host is infected with a worm.
- The Web server crashes.
- Users complain of slow access to hosts on the Internet.
- The system administrator sees a filename with unusual characters.

- The user calls the help desk to report a threatening e-mail message.
- The host records an auditing configuration change in its log.
- The application logs multiple failed login attempts from an unfamiliar remote system.
- The e-mail administrator sees a large number of bounced e-mails with suspicious content.
- The network administrator notices an unusual deviation from typical network traffic flows.

One should not think of incident detection as being strictly reactive. In some cases, the organization can detect activities that are likely to precede an incident. Examples of precursors are as follows:

- Web server log entries that show the usage of a Web vulnerability scanner
- An announcement of a new exploit that targets a vulnerability of the organization's mail server
- A threat from a hacktivist group stating that the group will attack the organization.

3.2.3 Sources of Precursors and Indications

Precursors and indications are identified using many different sources, with the most common being computer security software alerts, logs, publicly available information, and people.

Table 3-2. Common Sources of Presursors and Indications

Precursor or Indication Source	Description
Network and host-based IDS	IDS products are designed to identify suspicious events and record pertinent data regarding them, including the date and time the attack was detected, the type of attack, the source and destination IP addresses, and the username (if applicable and known). Most IDS products use a set of attack signatures to identify malicious activity; the signatures must be kept up to date so that the newest attacks can be detected. IDS software often produces false positives–alerts that indicate malicious activity is occurring, when in fact there has been none. Analysts should manually validate IDS alerts either by closely reviewing the recorded supporting data or by getting related data from other sources.
Antivirus software	When it detects malicious code, antivirus software typically sends alerts to the affected host and a centralized antivirus console. Current antivirus products are very effective at detecting and eradicating or isolating malicious code if their signatures are kept up to date. This updating task can be overwhelming in large organizations. One way of addressing it is to configure centralized antivirus software to push signature updates to individual hosts, rather than rely on hosts to be configured to pull updates. Because detection varies among antivirus products, some organizations use products from multiple vendors to provide better coverage and higher accuracy. Antivirus software should be deployed in at least two levels: at the network perimeter (e.g., firewalls, e-mail servers43) and at the host level (e.g., workstations, file servers, client software).

Precursor or Indication Source	Description
File Integrity checking software	Incidents may cause changes to important files; file integrity checking software can detect such changes. It works by using a hashing algorithm to obtain a cryptographic checksum for each designated file. If the file is altered and the checksum is recalculated, an extremely high probability exists that the new checksum will not match the old checksum. By regularly recalculating checksums and comparing them with previous values, changes to files can be detected.
Third-party monitoring services	Some organizations pay a third party to monitor their publicly accessible services. The third party automatically attempts to access each service every x minutes. If the service cannot be accessed, the third party alerts the organization using the methods specified by the organization. Some monitoring services can also detect and alert on changes in certain resources–for example, a Web page. Although a monitoring service is mainly useful from an operational standpoint, it can also provide an indication of a DoS attack or server compromise.
	Logs
Operating system, service and application logs	Logs from operating systems, services, and applications (particularly audit-related data) are frequently of great value when an incident occurs. To facilitate effective incident handling, organizations should require a baseline level of logging on all systems, and a higher baseline level of logging on critical systems. All systems should have auditing turned on and should log audit events, particularly administrative-level activity. All systems should be checked periodically to verify that logging is functioning properly and adheres to the logging standards.
Network device logs	Logs from network devices such as firewalls and routers are not typically used as a primary source of precursors or indications. Although these devices are usually configured to log blocked connection attempts, they provide little information about the nature of the activity. Still, they can be valuable in identifying trends (e.g., a significantly increased number of attempts to access a particular port) and in correlating events detected by other devices.
Information on incidents at other organizations	There are Web sites and mailing lists where incident response teams and security professionals can share information regarding reconnaissance and attacks that they have seen. In addition, some organizations acquire, consolidate, and analyze logs and intrusion detection alerts from many other organizations.
	People
People from within the organization	Users, system administrators, network administrators, security staff, and others from within the organization may report signs of incidents. It is important to validate all such reports. Not only do users generally lack the knowledge to determine if an incident is occurring, but also even the best-trained technical experts make mistakes. One approach is to ask people who provide such information how confident they are of the accuracy of the information. Recording this estimate along with the information provided can help considerably during incident analysis, particularly when conflicting data is discovered.
People from other organizations	Although few reports of incidents will originate from people at other organizations, they should be taken very seriously. Other incident response teams also may report incidents. It is important to have mechanisms in place for external parties to report indications and for trained staff to monitor those mechanisms carefully; this may be as simple as setting up a phone number and e-mail address, configured to forward messages to the help desk.

3.2.4 Incident Analysis

Incident detection and analysis would be easy if every precursor or indication were guaranteed to be accurate; unfortunately, this is not the case. Making matters worse, the total number of indications from human and automated sources may be thousands or millions a day. Finding the few real security incidents that occurred out of all the indications can be a daunting task.

Some incidents are easy to detect, such as an obviously defaced Web page. However, many incidents are not associated with such clear symptoms. Skilled attackers are careful to cover their tracks, and even unskilled attackers are becoming more difficult to detect because the tools that they use are more sophisticated and stealthy. Small signs such as one change in one system configuration file may be the only indications that an incident has occurred. In incident handling, detection may be the most difficult task. Incident handlers are responsible for analyzing ambiguous, contradictory, and incomplete symptoms to determine what has happened. Although technical solutions exist that can make detection somewhat easier, the best remedy is to build a team of highly experienced and proficient staff members who can analyze the precursors and indications effectively and efficiently and take appropriate actions. Without a well-trained and capable staff, incident detection and analysis will be conducted inefficiently, and costly mistakes will be made.

Performing the initial analysis and validation is challenging. The following are recommendations for making incident analysis easier and more effective:

- Profile Networks and Systems.
- Understand Normal Behaviors.
- Use Centralized Logging and Create a Log Retention Policy.
- Perform Event Correlation.
- Keep All Host Clocks Synchronized.
- Maintain and Use a Knowledge Base of Information.
- Use Internet Search Engines for Research.
- Run Packet Sniffers to Collect Additional Data.
- Consider Filtering the Data.
- Consider Experience as Being Irreplaceable.
- Create a Diagnosis Matrix for Less Experienced Staff.
- Seek Assistance From Others.

Symptom	Denial or Service	Malicious code	Unauthorized Access	Inappropriate Usage
Files, critical, access attempts	Low	Medium	High	Low
Files, Inappropriate content	Low	Medium	Low	High
Host crashes	Medium	Medium	Medium	Low
Port scans, Incoming, unusual	High	Low	Medium	Low
Port scans, outgoing, unusual	Low	High	Medium	Low
Utilization, bandwidth, high	High	Medium	Low	Medium
Utilization, e-mail, high	Medium	High	Medium	Medium

Table 3-3. Excerpt of a Sample Diagnosis Matrix

3.2.5 Incident Documentation

As soon as an incident response team suspects that an incident is occurring or has occurred, it is important to immediately start recording all facts regarding the incident. Every step taken from the time the incident was detected to its final resolution should be documented and timestamped. Every document regarding the incident should be dated and signed by the incident handler. Information of this nature can also be used as evidence in a court of law if legal prosecution is pursued.

The incident response team should take care to safeguard data related to incidents because it often contains sensitive information.

3.2.6 Incident Prioritization

Prioritizing the handling of the incident is perhaps the most critical decision point in the incident handling process. Incidents should not be handled on a first-come, first-served basis as a result of resource limitations. Instead, handling should be prioritized based on two factors:

- Current and Potential Technical Effect of the Incident.
- Criticality of the Affected Resources.

The team should prioritize the response to each incident based on its estimate of the business impact caused by the incident.

Organizations should document prioritization guidelines in a format such as the sample matrix shown in Table 3-4.

Table 3-4. Sample Incident Response SLA Matrix

Current Impact or Likely Future Impact of the Incident	Criticality of Resources Currently Impacted or Likely To Be Impacted by the Incident		
	High (e.g., Internet Connectivity, Public Web Servers, Firewalls, Customer Data)	Medium (e.g., System Administrator Workstations, File and Print Servers, XYZ Application Data)	Low (e.g., User Workstations)
Root-level access	15 minutes	30 minutes	1 hour
Unauthorized data modification	15 minutes	30 minutes	2 hours
Unauthorized access to sensitive data	15 minutes	1 hour	1 hour
Unauthorized user-level access	30 minutes	2 hours	4 hours
Services unavailable	30 minutes	2 hours	4 hours
Annoyance	30 minutes	Local IT staff	Local IT staff

More than one matrix entry may apply to an incident if it affects multiple resources (e.g., systems, applications, data). The incident handler can identify all applicable

matrix entries and follow the most urgent action first.
Incident handlers should have discretion to deviate from the matrix based on their judgment, especially when unforeseen or unusual circumstances occur.

Organizations should also establish an escalation process for those instances when the team does not respond to an incident within the designated time.

3.2.7 Incident Notification

Incident response policies should include provisions concerning incident reporting—at a minimum, what must be reported to whom and at what times (e.g., initial notification, regular status updates). The exact reporting requirements vary among agencies, but parties that are typically notified include—

- CIO
- Head of information security
- Local information security officer
- Other incident response teams within the organization
- System owner
- Human resources (for cases involving employees, such as harassment through e-mail)
- Public affairs (for incidents that may generate publicity)
- Legal department (for incidents with potential legal ramifications).

The team should plan and prepare several communication methods, and select the methods that are appropriate for a particular incident.

3.3 Containment, Eradication, and Recovery

Figure 3-3. Incident Response Life Cycle (Containment, Eradication and Recovery)

3.3.1 Choosing a Containment Strategy

When an incident has been detected and analyzed, it is important to contain it before the spread of the incident overwhelms resources or the damage increases. Most incidents require containment, so it is important to consider it early in the course of handling each incident. An essential part of containment is decision-making (e.g., shut down a system, disconnect it from a wired or wireless network, disconnect its modem cable, disable certain functions). Such decisions are much easier to make if strategies and procedures for containing the incident have been predetermined. Organizations should define acceptable risks in dealing with incidents and develop strategies accordingly.

Containment strategies vary based on the type of incident. It is highly recommended that organizations create separate containment strategies for each major type of incident. Criteria for determining the appropriate strategy include:

- Potential damage to and theft of resources
- Need for evidence preservation
- Service availability (e.g., network connectivity, services provided to external parties)
- Time and resources needed to implement the strategy
- Effectiveness of the strategy (e.g., partially contains the incident, fully contains the incident)
- Duration of the solution (e.g., emergency workaround to be removed in four hours, temporary workaround to be removed in two weeks, permanent solution).

In certain cases, some organizations delay the containment of an incident so that they can monitor the attacker's activity, usually to gather additional evidence. The incident response team should discuss delayed containment with its legal department to determine if it is feasible. Only a highly experienced incident response team that can monitor all of the attacker's actions and disconnect the attacker in a matter of seconds should attempt this strategy. Even then, the value of delayed containment is usually not worth the high risk that it poses.

Another potential issue regarding containment is that some attacks may cause additional damage when they are contained. Handlers should not assume that just because a host has been disconnected from the network, further damage to the host has been prevented.

3.3.2 Evidence Gathering and Handling

Although the primary reason for gathering evidence during an incident is to resolve the incident, it may also be needed for legal proceedings. In such cases, it is important to clearly document how all evidence, including compromised systems, has been preserved. Evidence should be collected according to procedures that meet all applicable laws and regulations, developed from previous discussions with legal staff and appropriate law enforcement agencies, so that it should be admissible in court. In addition, evidence should be accounted for at all times; whenever evidence is transferred from person to person, chain of custody forms should detail the transfer and include each party's signature. A detailed log should be kept for all evidence, including the following:

- Identifying information (e.g., the location, serial number, model number, hostname, media access control (MAC) address, and IP address of a computer)
- Name, title, and phone number of each individual who collected or handled the evidence during the investigation
- Time and date (including time zone) of each occurrence of evidence handling
- Locations where the evidence was stored.

Collecting evidence from computing resources presents some challenges. It is generally desirable to acquire evidence from a system of interest as soon as one suspects that an incident may have occurred. Many incidents cause a dynamic chain of events to occur; an initial system snapshot may do more good in identifying the problem and its source than most other actions that can be taken at this stage. From an evidentiary standpoint, it is much better to get a snapshot of the system as-is rather than doing so after incident handlers, system administrators, and others have inadvertently altered the state of the machine during the investigation. Users and system administrators should be made aware of the steps that they should take to preserve evidence.

Before copying the files from the affected host, it is often desirable to capture volatile information that may not be recorded in a file system or image backup, such as current network connections, processes, login sessions, open files, network interface configurations, and the contents of memory. This data may hold clues as to the attacker's identity or the attack methods that were used. It is also valuable to document how far the local clock deviates from the actual time. However, risks are associated with acquiring information from the live system. Any action performed on the host itself will alter the state of the machine to some extent. Also, the attacker may currently be on the system and notice the handler's activity, which could have disastrous consequences.

A well-trained and careful incident handler should be able to issue only the minimum commands needed for acquiring the dynamic evidence without inadvertently altering other evidence. Incident handlers can also use write blocker programs that prevent the host from writing to its hard drives.

After acquiring volatile data, an incident handler with computer forensics training should immediately make a full disk image to sanitized write-protectable or write-once media. A disk image preserves all data on the disk, including deleted files and file fragments. If it is possible that evidence may be needed for prosecution or internal disciplinary actions, the handlers should make at least two full images, label them properly, and securely store one of the images to be used strictly as evidence. (All evidence, not just disk images, should be tagged and stored in a secure location.) Occasionally, handlers may acquire and secure the original disk as evidence; the second image can then be restored to another disk as part of system recovery.

Obtaining a disk image is superior to a standard file system backup for computer forensic purposes because it records more data. Imaging is also preferable because it is much safer to analyze an image than it is to perform analysis on the original resource—the analysis may inadvertently alter or damage the original. If the business impact of taking down the system outweighs the risk of keeping the system operational, disk imaging may not be possible. A standard file system backup can capture information on existing files, which may be sufficient for handling many incidents, particularly those that are not expected to lead to prosecution. Both disk imaging and file system backups are valuable regardless of whether the attacker will be prosecuted because they permit the target to be restored while the investigation continues using the image or backup.

Computer forensic software is valuable not only for acquiring disk images, but also for automating much of the analysis process, such as:

- Identifying and recovering file fragments and hidden and deleted files and directories from any location (e.g., used space, free space, slack space)
- Examining file structures, headers, and other characteristics to determine what type of data each file contains, instead of relying on file extensions (e.g., .doc, .jpg, .mp3)
- Displaying the contents of all graphics files
- Performing complex searches
- Graphically displaying the acquired drive's directory structure
- Generating reports.

During evidence acquisition, it is often prudent to acquire copies of supporting log files from other resources—for example, firewall logs that show what IP address an attacker used. As with hard drive and other media acquisition, logs should be copied to sanitized write-protectable or write-once media. One copy of the logs should be stored as evidence, whereas a second copy could be restored to another system for further analysis. Many incident handlers create a message digest for log files and other pieces of digital evidence; this refers to generating a cryptographic checksum for a file. If the file is modified and the checksum is recalculated, there is only an infinitesimal chance that the checksums will be the same. Message digests should be generated using software and a message digest algorithm that are FIPS 140-2 and FIPS 180-2 validated. (Message digests are also useful for other computer forensic purposes—for example, when acquiring media, handlers can generate checksums of the original media and the duplicates to show that integrity was maintained during imaging.) Incident handlers should also document the local clock time on each logging host and what deviation, if any, there is from the actual time.

3.3.3 Identifying the Attacker

During incident handling, system owners and others typically want to identify the attacker. Although this information can be important, particularly if the organization wants to prosecute the attacker, incident handlers should stay focused on containment, eradication, and recovery. Identifying the attacker can be a time-consuming and futile process that can prevent a team from achieving its primary goal—minimizing the business impact. The following items describe the most commonly performed activities for attacker identification:

- Validating the Attacker's IP Address.
- Scanning the Attacker's System. Incident handlers should discuss this issue with legal representatives before performing such scans because the scans may violate organization policies or even break the law.
- Researching the Attacker Through Search Engines.
- Using Incident Databases.
- Monitoring Possible Attacker Communication Channels.

3.3.4 Eradication and Recovery

After an incident has been contained, eradication may be necessary to eliminate components of the incident, such as deleting malicious code and disabling breached user accounts. For some incidents, eradication is either not necessary or is performed during recovery. In recovery, administrators restore systems to normal operation and (if applicable) harden systems to prevent similar incidents. Because eradication and recovery actions are typically operating system (OS) or application-specific, detailed recommendations and guidance regarding them are outside the scope of this document.

3.4 Post-Incident Activity

Figure 3-4. Incident Response Life Cycle (Post-Incident Activity)

3.4.1 Lessons Learned

One of the most important parts of incident response is also the most often omitted: learning and improving. Each incident response team should evolve to reflect new threats, improved technology, and lessons learned. Many organizations have found that holding a "lessons learned" meeting with all involved parties after a major incident, and periodically after lesser incidents, is extremely helpful in improving security measures and the incident handling process itself.

Another important post-incident activity is creating a follow-up report for each incident, which can be quite valuable for future use. First, the report provides a reference that can be used to assist in handling similar incidents. Creating a formal chronology of events (including timestamped information such as log data from systems) is important for legal reasons, as is creating a monetary estimate of the amount of damage the incident caused in terms of any loss of software and files, hardware damage, and staffing costs (including restoring services). This estimate may become the basis for subsequent prosecution activity by entities such as the U.S. Attorney General's office.

3.4.2 Using Collected Incident Data

Lessons learned activities should produce a set of objective and subjective data regarding each incident. Over time, the collected incident data should be useful in several capacities. The data, particularly the total hours of involvement and the cost, may be used to justify additional funding of the incident response team. A study of incident characteristics may indicate systemic security weaknesses and threats, as well as changes in incident trends. This data can be put back into the risk assessment process, ultimately leading to the selection and implementation of additional controls. Another good use of the data is measuring the success of the incident response team.

If incident data is collected and stored properly, it should provide several measures of the success (or at least the activities) of the incident response team. Furthermore, organizations (e.g., Federal agencies) that are required to report incident information will need to collect the necessary data to meet their requirements.

Possible metrics for incident-related data include:

- Number of Incidents Handled.
- Time Per Incident.
- Objective Assessment of Each Incident.
- Subjective Assessment of Each Incident.

Besides using these metrics to measure the team's success, organizations may also find it useful to periodically audit their incident response programs. Audits will identify problems and deficiencies that can then be corrected. At a minimum, an incident response audit should evaluate the following items against applicable regulations, policies, and best practices:

- Incident response policies and procedures
- Tools and resources
- Team model and structure
- Incident handler training and education
- Incident documentation and reports
- The measures of success discussed earlier in this section.

3.4.3 Evidence Retention

Organizations should establish policy for how long evidence from an incident should be retained. Most organizations choose to retain all evidence for months or years after the incident ends. The following factors should be considered during the policy creation:

- **Prosecution**. If it is possible that the attacker will be prosecuted, evidence may need to be retained until all legal actions have been completed.
- **Data Retention**. Most organizations have data retention policies that state how long certain types of data may be kept.
- **Cost**. Original hardware (e.g., hard drives, compromised systems) that is stored as evidence, as well as hard drives and other devices that are used to hold disk images, are individually inexpensive for most organizations.

3.5 Incident Handling Checklist

The checklists provide guidance to handlers on the major steps that should be performed; they do not dictate the exact sequence of steps that should always be followed.

Table 3-5. Initial Incident Handling Checklist

	Action	Completed
	Detection and Analysis	
1.	Determine whether an incident has occurred	
1.1	Analyze the precursors and indications	
1.2	Look for correlating information	
1.3	Perform research (e.g., search engines, knowledge base)	
1.4	As soon as the handler believes an incident has occurred, begin documenting the investigation and gathering evidence	
2.	Classify the incident using the categories presented in Section 3.2.1 (e.g., denial of service, malicious code, unauthorized access, inappropriate usage, multiple component)	
3.	Follow the appropriate incident category checklist; if the incident does not fit into any of the categories, follow the generic checklist	

Table 3-6. Generic Incident Handling Checklist for Uncategorized Incidents

	Action	Completed
	Detection and Analysis	
1.	Prioritize handling the incident based on the business impact	
1.1	Identify which resources have been affected and forecast which resources will be affected	
1.2	Estimate the current and potential technical effect of the incident	
1.3	Find the appropriate cell(s) in the prioritization matrix, based on the technical effect and affected resources	
2	Report the incident to the appropriate internal personnel and external organizations	
	Containment, Eradication, and Recovery	
3.	Acquire, preserve, secure, and document evidence	
4.	Contain the incident	
5.	Eradicate the incident	
5.1	Identify and mitigate all vulnerabilities that were exploited	
5.2	Remove malicious code, inappropriate materials, and other components	
6.	Recover from the incident	
6.1	Return affected systems to an operationally ready state	
6.2	Confirm that the affected systems are functioning normally	
6.3	If necessary, implement additional monitoring to look for future related activity	
	Post-Incident Activity	
7.	Create a follow-up report	
8.	Hold a lessons learned meeting	

3.6 Recommendations

The key recommendations presented in this section for handling incidents are summarized below.

- **Acquire tools and resources that may be of value during incident handling**. The team will be more efficient at handling incidents if various tools and resources are already available to them. Examples include contact lists, encryption software, network diagrams, backup devices, computer forensic software, port lists, and security patches.
- **Prevent incidents from occurring by ensuring that networks, systems, and applications are sufficiently secure**. Preventing incidents is beneficial to the organization and also reduces the workload of the incident response team. Performing periodic risk assessments and reducing the identified risks to an acceptable level are effective in reducing the number of incidents. User, IT staff, and management awareness of security policies and procedures is also very important.
- **Identify precursors and indications through alerts generated by several types of computer security software**. Network and host-based intrusion detection systems, antivirus software, and file integrity checking software are valuable for detecting signs of incidents. Each type of software may detect incidents that the other types of software cannot; therefore, the use of several types of computer security software is highly recommended. Third-party monitoring services can also be helpful.
- **Establish mechanisms for outside parties to report incidents**. Outside parties may want to report incidents to the organization; for example, they may believe that one of the organization's users is attacking them. Organizations should publish a phone number and e-mail address that outside parties can use to report such incidents.
- **Require a baseline level of logging and auditing on all systems, and a higher baseline level on all critical systems**. Logs from operating systems, services, and applications frequently provide value during incident analysis, particularly if auditing was enabled. The logs can provide information such as which accounts were accessed and what actions were performed.
- **Profile networks and systems**. Profiling measures the characteristics of expected activity levels so that changes in patterns can be more easily identified. If the profiling process is automated, deviations from expected activity levels can be detected and reported to administrators quickly, leading to faster detection of incidents and operational issues.
- **Understand the normal behaviors of networks, systems, and applications**. Team members who understand normal behavior should be able to recognize abnormal behavior more easily. This knowledge can best be gained by reviewing log entries and security alerts; the handlers should become familiar with the typical data and can investigate the unusual entries to gain more knowledge.

- **Use centralized logging and create a log retention policy**. Information regarding an incident may be recorded in several places. Organizations should deploy centralized logging servers and configure devices to send duplicates of their log entries to the centralized servers. The team benefits because it can access all log entries at once; also, changes made to logs on individual hosts will not affect the data already sent to the centralized servers. A log retention policy is important because older log entries may show previous instances of similar or related activity.
- **Perform event correlation**. Indications of an incident may be captured in several logs. Correlating events among multiple sources can be invaluable in collecting all the available information for an incident and validating whether the incident occurred. Centralized logging makes event correlation easier and faster.
- **Keep all host clocks synchronized**. If the devices reporting events have inconsistent clock settings, event correlation will be more difficult. Clock discrepancies may also cause issues from an evidentiary standpoint.
- **Maintain and use a knowledge base of information**. Handlers need to reference information quickly during incident analysis; a centralized knowledge base provides a consistent, maintainable source of information. The knowledge base should include general information, such as commonly used port numbers and links to virus information, as well as data on precursors and indications of previous incidents.
- **Create a diagnosis matrix for less experienced staff**. Help desk staff, system administrators, and new incident response team members may need assistance in determining what type of incident may be occurring. A diagnosis matrix that lists incident categories and the symptoms associated with each category can provide guidance as to what type of incident is occurring and how the incident can be validated.
- **Start recording all information as soon as the team suspects that an incident has occurred**. Every step taken, from the time the incident was detected to its final resolution, should be documented and timestamped. Information of this nature can serve as evidence in a court of law if legal prosecution is pursued. Recording the steps performed can also lead to a more efficient and systematic, and less error-prone handling of the problem.
- **Safeguard incident data**. It often contains sensitive information regarding such things as vulnerabilities, security breaches, and users that may have performed inappropriate actions. The team should ensure that access to incident data is restricted properly, both logically and physically.
- **Prioritize incidents by business impact, based on the criticality of the affected resources and the technical impact of the incident**. Because of resource limitations, incidents should not be handled on a first-come, first-served basis. Instead, organizations should establish written guidelines that outline how quickly the team must respond to the incident and what actions should be performed, based on the incident's current and potential business impact. This guidance saves time for the incident handlers and provides a

justification to management and system owners for their actions. Organizations should also establish an escalation process for those instances when the team does not respond to an incident within the designated time.

- **Include provisions regarding incident reporting in the organization's incident response policy**. Organizations should specify which incidents must be reported, when they must be reported, and to whom. The parties most commonly notified are the CIO, head of information security, local information security officer, other incident response teams within the organization, and system owners.
- **Establish strategies and procedures for containing incidents**. It is important to contain incidents quickly and effectively to limit their business impact. Organizations should define acceptable risks in containing incidents and develop strategies and procedures accordingly. Containment strategies should vary based on the type of incident.
- **Follow established procedures for evidence gathering and handling**. The team should clearly document how all evidence has been preserved. Evidence should be accounted for at all times. The team should meet with legal staff and law enforcement agencies to discuss evidence handling, then develop procedures based on those discussions.
- **Capture volatile data from systems as evidence**. This includes lists of network connections, processes, login sessions, open files, network interface configurations, and the contents of memory. Running carefully chosen commands from trusted media can collect the necessary information without damaging the system's evidence.
- **Obtain system snapshots through full forensic disk images, not file system backups**. Disk images should be made to sanitized write-protectable or write-once media. This process is superior to a file system backup for investigatory and evidentiary purposes. Imaging is also valuable in that it is much safer to analyze an image than it is to perform analysis on the original system because the analysis may inadvertently alter the original.
- **Hold lessons learned meetings after major incidents**. Lessons learned meetings are extremely helpful in improving security measures and the incident handling process itself.

4. Handling Denial of Service Incidents

4.1 Incident Definition and Examples

A denial of service (DoS) is an action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as central processing units (CPU), memory, bandwidth, and disk space.

4.2 Preparation

This section provides guidance on preparing to handle DoS incidents and on preventing DoS incidents.

4.2.1 Incident Handling Preparation

In addition to the general guidance presented in Sections 3.1.1 and 3.2.3, other actions should be performed while preparing to handle DoS incidents:

- Talk with the organization's ISPs and their second-tier providers to determine how they can assist in handling network-based DoS attacks.
- Consider investigating the feasibility of participating in a coordinated response to a widespread DoS attack that affects many organizations.
- Deploy and configure intrusion detection software to detect DoS and DDoS traffic.
- Perform ongoing resource monitoring to establish baselines of network bandwidth utilization and critical host resource utilization, and log or alert when there is a significant deviation from the baselines.
- Identify Web sites that provide statistics on latency between various ISPs and between various physical locations. This is often referred to as Internet health monitoring. When a network-based DoS occurs, incident handlers could use such Web sites to attempt to determine if similar attacks are currently affecting other organizations (e.g., a worm causing regional disruptions).
- Meet with network infrastructure administrators to discuss how they can assist in analyzing and containing network-based DoS and DDoS attacks.
- Maintain local copies (electronic and/or paper) of any computer-based information that may be valuable in handling DoS incidents in case the organization's Internet or internal network connectivity is lost during an incident.

4.2.2 Incident Prevention

Section 3.1.2 has guidelines and pointers to resources on incident prevention. The following items provide additional recommendations for preventing DoS incidents:

- Configure the network perimeter to deny all incoming and outgoing traffic that is not expressly permitted. This should include:
  - Blocking the usage of services, such as echo and chargen, that no longer serve a legitimate purpose and are used in DoS attacks.

  - Performing egress and ingress filtering to block obviously spoofed packets.
  - Blocking traffic from unassigned IP address ranges, known as bogon lists. Attack tools that spoof IP addresses may use addresses that have not yet been assigned for Internet usage.
  - Writing and sequencing firewall rules and router access control lists to block traffic properly.
  - Configuring border routers not to forward directed broadcasts.
  - Limiting incoming and outgoing ICMP traffic to only the necessary types and codes.
  - Blocking outgoing connections to common IRC, peer-to-peer service and instant messaging ports if the usage of such services is not permitted.
- Implement rate limiting for certain protocols, such as ICMP, so that they can only consume a designated percentage of the total bandwidth. Rate limiting can be implemented at the organization's network perimeter (e.g., border routers, firewalls) and by the organization's ISPs.
- On Internet-accessible hosts, disable all unneeded services, and restrict the use of services that may be used in DoS attacks.
- Implement DoS prevention software. The software can study network traffic patterns, detect major deviations, and block traffic accordingly. Unfortunately, DoS anomaly detection is not completely accurate, so the software may block legitimate traffic or fail to block a true DoS attack.
- Implement redundancy for key functions (e.g., multiple ISPs, firewalls, Web servers).
- Ensure that networks and systems are not running near maximum capacity, or it could be easy for a minor DoS attack to take up the remaining resources.

4.3 Detection and Analysis

DoS attacks can be detected through particular precursors and indications, primarily those listed in the following tables.

Table 4-1. Denial of Service Precursors

Precursor	Response
DoS attacks are often preceded by reconnaissance activity—generally, a low volume of the traffic that will be used in the actual attack—to determine which attacks may be effective.	If handlers detect unusual activity that appears to be preparation for a DoS attack, the organization may be able to block the attack by quickly altering its security posture—for example, altering firewall rulesets to block a particular protocol from being used or protect a vulnerable host.
A newly released DoS tool could pose a significant threat to the organization.	Investigate the new tool and, if possible, alter security controls so that the tool should not be effective against the organization.

Table 4-2. Denial of Service Indications

Malicious Action	Possible Indications
Network-based DoS against a particular host	• User reports of system unavailability • Unexplained connection losses • Network intrusion detection alerts • Host intrusion detection alerts (until the host is overwhelmed) • Increased network bandwidth utilization • Large number of connections to a single host • Asymmetric network traffic pattern (large amount of traffic going to the host, little traffic coming from the host) • Firewall and router log entries • Packets with unusual source addresses
Network-based DoS against a network	• User reports of system and network unavailability • Unexplained connection losses • Network intrusion detection alerts • Increased network bandwidth utilization • Asymmetric network traffic pattern (large amount of traffic entering the network, little traffic leaving the network) • Firewall and router log entries • Packets with unusual source addresses • Packets with nonexistent destination addresses
DoS against the operating system of a particular host	• User reports of system and application unavailability • Network and host intrusion detection alerts • Operating system log entries • Packets with unusual source addresses
DoS against an application on a particular host	• User reports of application unavailability • Network and host intrusion detection alerts • Application log entries • Packets with unusual source addresses

Although these tables may be helpful in analyzing incidents, they are missing an important component—the indications that are associated with benign activities. Benign and malicious events may present similar symptoms, which can make it difficult for analysts to promptly determine if an incident has occurred. Extending the indications table to include benign activities should assist in distinguishing benign from malicious activity.

4.4 Containment, Eradication, and Recovery

In addition to the general guidance presented in Section 3.3, this section gives specific recommendations for performing containment, and gathering and handling evidence for DoS incidents.

4.4.1 Choosing a Containment Strategy

Containment for a DoS incident usually consists of stopping the DoS. Sometimes this is easy; usually it is not. Often the first thought is to block all traffic from the source

of the activity. However, as previously mentioned, such attacks often have spoofed source addresses or use hundreds or thousands of compromised hosts—in either case, making it difficult or impossible to implement effective filtering based on source IP addresses. Other possible solutions for containing a DoS are as follows:

- **Correct the Vulnerability or Weakness That Is Being Exploited**.
- **Implement Filtering Based on the Characteristics of the Attack**.
- **Have the ISP Implement Filtering**.
- **Relocate the Target**.
- **Attack the Attackers**. For example, administrators may use programs that are designed to remotely shut off attacking DDoS agents, or they may modify network or server configurations to bounce attack traffic back to its source.

4.4.2 Evidence Gathering and Handling

Gathering evidence on DoS attacks is often challenging and time consuming, for any of several reasons:

- **Identifying the Source of Attacks From Observed Traffic**. IP source addresses are frequently spoofed. DDoS attacks may use hundreds or thousands of hosts, each of which may use multiple spoofed addresses. Even if hosts use their actual addresses, these are the intermediate boxes generating the attack traffic, not the system that orchestrated the overall attack.
- **Tracing Attacks Back Through ISPs**. Incident handlers may have to contact several ISPs in turn to trace an attack back to its source, assuming that it is possible, technically and logistically. Because of the amount of time that it can take to get assistance from each ISP, it is likely that the attack will end before it can be traced back.
- **Learning How the Attacking Hosts Were Compromised**. In a DDoS, the agent hosts may have been compromised using dozens of methods over a long period of time. The attacker may not have even been responsible for any of the compromises; instead, the attacker is simply using hosts that others had previously compromised.
- **Reviewing a Large Number of Log Entries**. Because most DoS attacks work by overwhelming resources, it follows that they may generate unusually large numbers of log entries. Depending on logging standards and practices, log entries may be overwritten, eliminating evidence. Also, it may take a long time to review all of the log entries and extract the pertinent information.

4.5 Checklist for Handling Denial of Service Incidents

Note that the exact sequence of steps may vary based on the nature of individual incidents, and on the strategies chosen by the organization for halting DoS attacks that are in progress.

Table 4-3. Denial of Service Incident Handling Checklist

	Action	Completed
	Detection and Analysis	
1.	Prioritize handling the incident based on the business impact	
1.1	Identify which resources have been affected and forecast which resources will be affected	
1.2	Estimate the current and potential technical effect of the incident	
1.3	Find the appropriate cell(s) in the prioritization matrix, based on the technical effect and affected resources	
2.	Report the incident to the appropriate internal personnel and external organizations	
	Containment, Eradication, and Recovery	
3.	Acquire, preserve, secure, and document evidence	
4.	Contain the incident	
5.	Eradicate the incident	
5.1	Identify and mitigate all vulnerabilities that were exploited	
5.2	Remove malicious code, inappropriate materials, and other components	
6.	Recover from the incident	
6.1	Return affected systems to an operationally ready state	
6.2	Confirm that the affected systems are functioning normally	
6.3	If necessary, implement additional monitoring to look for future related activity	
	Post-Incident Activity	
7.	Create a follow-up report	
8.	Hold a lessons learned meeting	

4.6 Recommendations

The key recommendations presented in this section for handling DoS incidents are summarized below.

- **Configure firewall rulesets to prevent reflector attacks**. Most reflector attacks can be stopped through network-based and host-based firewall rulesets that reject suspicious combinations of source and destination ports.
- **Configure border routers to prevent amplifier attacks**. Amplifier attacks can be blocked by configuring border routers not to forward directed broadcasts.
- **Determine how the organization's ISPs and second-tier providers can assist in handling network-based DoS attacks**. ISPs can often filter or limit certain types of traffic, slowing or halting a DoS attack. They can also provide logs of DoS traffic and may be able to assist in tracing the source of the attack. The organization should meet with the ISPs in advance to establish procedures for requesting such assistance.

- **Configure security software to detect DoS attacks**. Intrusion detection software can detect many types of DoS activity. Establishing network and system activity baselines, and monitoring for significant deviations from those baselines, can also be useful in detecting attacks.
- **Configure the network perimeter to deny all incoming and outgoing traffic that is not expressly permitted**. By restricting the types of traffic that can enter and leave the environment, the organization will limit the methods that attackers can use to perform DoS attacks.
- **Create a containment strategy that includes several solutions in sequence**. The decision-making process for containing DoS incidents is easier if recommended solutions are predetermined. Because the effectiveness of each possible solution will vary among incidents, organizations should select several solutions and determine in which order the solutions should be attempted.

5. Handling Malicious Code Incidents

5.1 Incident Definition and Examples

Malicious code refers to a program that is covertly inserted into another program with the intent to destroy data, run destructive or intrusive programs, or otherwise compromise the security or integrity of the victim's data. Generally, malicious code is designed to perform these nefarious functions without the system's user knowledge. Malicious code attacks can be divided into five categories: viruses, Trojan horses, worms, mobile code, and blended.

5.2 Preparation

This section provides guidance on preparing to handle malicious code incidents and on preventing malicious code incidents.

5.2.1 Incident Handling Preparation

In addition to the following general guidance presented in Sections 3.1.1 and 3.2.3, other actions should be taken in preparation for handling malicious code incidents:

- Make Users Aware of Malicious Code Issues.
- Read Antivirus Vendor Bulletins.
- Deploy Host-Based Intrusion Detection Systems to Critical Hosts.

5.2.2 Incident Prevention

The following paragraphs provide specific advice on preventing malicious code incidents:

- Use Antivirus Software.
- Block Suspicious Files.
- Limit the Use of Nonessential Programs With File Transfer Capabilities.
- Educate Users on the Safe Handling of E-mail Attachments.
- Eliminate Open Windows Shares.
- Use Web Browser Security to Limit Mobile Code.
- Configure E-mail Clients to Act More Securely.

5.3 Detection and Analysis

Because malicious code incidents can take many forms, they may be detected via a number of precursors and indications.

Table 5-1. Malicious Code Precursors

Precursor	Response
An alert warns of new malicious code that targets software that the organization uses.	Research the new virus to determine whether it is real or a hoax. This can be done through antivirus vendor Web sites and virus hoax sites. If the malicious code is confirmed as authentic, ensure that antivirus software is updated with virus signatures for the new malicious code. If a virus signature is not yet available, and the threat is serious and imminent, the activity might be blocked through other means, such as configuring e-mail servers or clients to block e-mails matching characteristics of the new malicious code. The team might also want to notify antivirus vendors of the new virus.
Antivirus software detects and successfully disinfects or quarantines a newly received infected file.	Determine how the malicious code entered the system and what vulnerability or weakness it was attempting to exploit. If the malicious code might pose a significant risk to other users and hosts, mitigate the weaknesses that the malicious code used to reach the system and would have used to infect the target host.

Table 5-2. Malicious Code Indications

Malicious Action	Possible Indications
A virus that spreads through e-mail infects a host.	• Antivirus software alerts of infected files • Sudden increase in the number of e-mails being sent and received • Changes to templates for word processing documents, spreadsheets, etc. • Deleted, corrupted, or inaccessible files • Unusual items on the screen, such as odd messages and graphics • Programs start slowly, run slowly, or do not run at all • System instability and crashes • If the virus achieves root-level access, see the indications for "Root compromise of a host" as listed in Table 6-3, Unauthorized Access Indications
A worm that spreads through a vulnerable service infects a host.	• Antivirus software alerts of infected files • Port scans and failed connection attempts targeted at the vulnerable service (e.g., open Windows shares, HTTP) • Increased network usage • Programs start slowly, run slowly, or do not run at all • System instability and crashes • If the worm achieves root-level access, see the indications for "Root compromise of a host" as listed in Table 6-3, Unauthorized Access Indications
A Trojan horse is installed and running on a host.	• Antivirus software alerts of Trojan horse versions of files • Network intrusion detection alerts of Trojan horse client-server communications • Firewall and router log entries for Trojan horse client-server communications • Network connections between the host and unknown remote systems • Unusual and unexpected ports open • Unknown processes running • High amounts of network traffic generated by the host, particularly if directed at external host(s) • Programs start slowly, run slowly, or do not run at all • System instability and crashes • If the Trojan horse achieves root-level access, see the indications for "Root compromise of a host" as listed in Table 6-3, Unauthorized Access Indications

Malicious Action	Possible Indications
Malicious mobile code on a Web site is used to infect a host with a virus, worm, or Trojan horse.	• Indications listed above for the pertinent type of malicious code • Unexpected dialog boxes, requesting permission to do something • Unusual graphics, such as overlapping or overlaid message boxes
Malicious mobile code on a Web site exploits vulnerabilities on a host.	• Unexpected dialog boxes, requesting permission to do something • Unusual graphics, such as overlapping or overlaid message boxes • Sudden increase in the number of e-mails being sent and received • Network connections between the host and unknown remote systems • If the mobile code achieves root-level access, see the indications for "Root compromise of a host" as listed in Table 6-3, Unauthorized Access Indications
A user receives a virus hoax message.	• Original source of the message is not an authoritative computer security group, but a government agency or an important official person • No links to outside sources • Tone and terminology attempt to invoke panic or a sense of urgency • Urges recipients to delete certain files and forward the message to others

5.4 Containment, Eradication, and Recovery

In addition to the general guidance presented in Section 3.3, this section gives specific recommendations for performing containment and for gathering and handling evidence for malicious code incidents.

5.4.1 Choosing a Containment Strategy

Because malicious code works surreptitiously and can propagate to other systems rapidly, early containment of a malicious code incident is needed to stop it from spreading and causing further damage. If the infected system is not critical, disconnecting it from the network immediately is strongly recommended. If the system performs critical functions, it should remain on the network only if the damage to the organization from the services being unavailable is greater than the security risks posed by not immediately disconnecting the system. Other actions that may need to be performed when containing a malicious code incident are as follows:

- Any of the Actions Listed in Section 5.2.2.
- Identifying and Isolating Other Infected Hosts.
- Sending Unknown Malicious Code to Antivirus Vendors.
- Configuring E-mail Servers and Clients to Block E-mails.
- Blocking Particular Hosts.
- Shutting Down E-mail Servers.
- Isolating Networks From the Internet.

Identifying infected hosts and vulnerable hosts is made quite complicated by the dynamic nature of computing. If all hosts were powered on and connected to the network at all times, malicious code cleanup would be relatively easy. The actual situation is that hosts may be infected and powered off, moved to other networks, or left

on while the system owner is out of the office. Vulnerable hosts that are shut off while their owners are on vacation may quickly become infected when they are powered back on. The identification of vulnerable hosts and infected hosts should not rely solely on user participation. However, organizations often lack the personnel and time to track down each machine manually, particularly when there are substantial numbers of mobile users and telecommuters. Automated methods may also be inadequate for identifying all hosts, such as those that can boot to multiple operating systems or use virtual operating system software. Organizations should carefully consider these issues before a large-scale malicious code incident occurs, so that they are prepared to implement effective containment strategies.

5.4.2 Evidence Gathering and Handling

Although it is certainly possible to gather evidence on malicious code incidents, it is often futile because malicious code is either transmitted automatically or is accidentally transmitted by infected users. It is therefore very difficult and time-consuming to identify the source of malicious code. However, collecting a sample of malicious code for further examination might be useful in some cases.

5.4.3 Eradication and Recovery

Antivirus software effectively identifies and removes malicious code infections; however, some infected files cannot be disinfected. (Files can be deleted and replaced with clean backup copies; in the case of an application, the affected application can be reinstalled.) If the malicious code provided attackers with root-level access, it may not be possible to determine what other actions the attackers may have performed. In such cases, the system should either be restored from a previous, uninfected backup or be rebuilt from scratch. The system should then be secured so that it will not be susceptible to another infection from the same malicious code.

5.5 Checklist for Handling Malicious Code Incidents

Note that the exact sequence of steps may vary based on the nature of individual incidents and the strategies chosen by the organization for containing incidents.

Table 5-3. Malicious Code Incident Handling Checklist

	Action	Completed
	Detection and Analysis	
1.	Prioritize handling the incident based on the business impact	
1.1	Identify which resources have been affected and forecast which resources will be affected	
1.2	Estimate the current and potential technical effect of the incident	
1.3	Find the appropriate cell(s) in the prioritization matrix, based on the technical effect and affected resources	
2.	Report the incident to the appropriate internal personnel and external organizations	
	Containment, Eradication, and Recovery	
3.	Contain the incident	
3.1	Identify infected systems	
3.2	Disconnect infected systems from the network	
3.3	Identify and mitigate all vulnerabilities that were exploited	
3.4	If necessary, block the transmission mechanisms for the malicious code	
4.	Eradicate the incident	
4.1	Disinfect, quarantine, delete, and replace infected files	
4.2	Mitigate the exploited vulnerabilities for other hosts within the organization	
5.	Recover from the incident	
5.1	Confirm that the affected systems are functioning normally	
5.2	If necessary, implement additional monitoring to look for future related activity	
	Post-Incident Activity	
6.	Create a follow-up report	
7.	Hold a lessons learned meeting	

5.6 Recommendations

The key recommendations presented in this section for handling malicious code incidents are summarized below.

- **Make users aware of malicious code issues**. Users should be familiar with the methods that malicious code uses to propagate and the symptoms of infections. Holding regular user education sessions helps to ensure that users are aware of the risks that malicious code poses. Teaching users how to safely handle e-mail attachments should reduce the number of infections that occur.
- **Read antivirus bulletins**. Bulletins regarding new malicious code threats provide timely information to incident handlers.
- **Deploy host-based intrusion detection systems, including file integrity checkers, to critical hosts**. Host-based IDS software, particularly file integrity checkers, can detect signs of malicious code incidents, such as configuration changes and modifications to executables.

- **Use antivirus software, and keep it updated with the latest virus signatures**. Antivirus software should be deployed to all hosts and all applications that may be used to transfer malicious code. The software should be configured to detect and disinfect or quarantine malicious code infections. All antivirus software should be kept current with the latest virus signatures so the newest threats can be detected.
- **Configure software to block suspicious files**. Files that are very likely to be malicious should be blocked from the environment, such as those with file extensions that are usually associated with malicious code, as well as files with suspicious combinations of file extensions.
- **Eliminate open Windows shares**. Many worms spread through unsecured shares on hosts running Windows. A single infection may rapidly spread to hundreds or thousands of hosts through unsecured shares.
- **Contain malicious code incidents as quickly as possible**. Because malicious code works surreptitiously and can propagate to other systems rapidly, early containment of a malicious code incident is needed to stop it from spreading and causing further damage. Infected systems should be disconnected from the network immediately. Organizations may need to block malicious code at the e-mail server level, or even temporarily suspend e-mail services to gain control over serious e-mail-borne malicious code incidents.

6. Handling Unauthorized Access Incidents

6.1 Incident Definition and Examples

An unauthorized access incident occurs when a user gains access to resources that the user was not intended to have.

6.2 Preparation

This section provides guidance on preparing to handle unauthorized access incidents and on preventing unauthorized access incidents.

6.2.1 Incident Handling Preparation

In addition to following the general guidance presented in Sections 3.1.1 and 3.2.3, other actions should be performed while preparing to handle unauthorized access incidents:

- Configure network-based and host-based IDS software (such as file integrity checkers and log monitors) to identify and alert on attempts to gain unauthorized access. Each type of intrusion detection software may detect attacks that others are not able to detect.
- Use centralized log servers so pertinent information from hosts across the organization is stored in a single secured location.
- Establish procedures to be followed when all users of an application, system, trust domain, or organization should change their passwords because of a password compromise. The procedures should adhere to the organization's password policy.
- Discuss unauthorized access incidents with system administrators so that they understand their roles in the incident handling process.

6.2.2 Incident Prevention

If the general guidance presented in Section 3.1.2 on incident prevention is applied, the number of unauthorized access incidents should be effectively reduced. Employing a strong layered defense strategy, with several security layers between unauthorized users and the resources they are attempting to exploit, is the recommended practice for reducing incidents. Table 6-1 lists additional steps that support a layered defense strategy.

Table 6-1. Actions to Prevent Unauthorized Access Incidents

Category	Specific Actions
Network Security	• Configure the network perimeter to deny all incoming traffic that is not expressly permitted. • Properly secure all remote access methods, including modems and VPNs. An unsecured modem can provide easily attainable unauthorized access to internal systems and networks. War dialing is the most efficient technique for identifying improperly secured modems.92 When securing remote access, carefully consider the trustworthiness of the clients; if they are outside the organization's control, they should be given as little access to resources as possible, and their actions should be closely monitored. • Put all publicly accessible services on secured demilitarized zone (DMZ) network segments. The network perimeter can then be configured so that external hosts can establish connections only to hosts on the DMZ, not internal network segments. • Use private IP addresses for all hosts on internal networks. This will severely restrict the ability of attackers to establish direct connections to internal hosts.
Host Security	• Perform regular vulnerability assessments to identify serious risks and mitigate the risks to an acceptable level. • Disable all unneeded services on hosts. Separate critical services so they run on different hosts. If an attacker then compromises a host, immediate access should be gained only to a single service. • Run services with the least privileges possible to reduce the immediate impact of successful exploits. • Use host-based firewall software to limit individual hosts' exposure to attacks. • Limit unauthorized physical access to logged-in systems by requiring hosts to lock idle screens automatically and asking users to log off before leaving the office. • Regularly verify the permission settings for critical resources, including password files, sensitive databases and public Web pages. This process can be easily automated to report changes in permissions on a regular basis.
Authentication and Authorization	• Create a password policy that requires the use of complex, difficult-to-guess passwords, forbids password sharing, and directs users to use different passwords on different systems, especially external hosts and applications. • Require sufficiently strong authentication, particularly for accessing critical resources. • Create authentication and authorization standards for employees and contractors to follow when evaluating or developing software. For example, passwords should be strongly encrypted using a FIPS 140-2 validated algorithm when they are transmitted or stored. • Establish procedures for provisioning and deprovisioning user accounts. These should include an approval process for new account requests and a process for periodically disabling or deleting accounts that are no longer needed.
Physical Security	• Implement physical security measures that restrict access to critical resources.

6.3 Detection and Analysis

Table 6-2. Unauthorized Access Precursors

Precursor	Response
Unauthorized access incidents are often preceded by reconnaissance activity to map hosts and services and to identify vulnerabilities. Activity may include port scans, host scans, vulnerability scans, pings, traceroutes, DNS zone transfers, OS fingerprinting, and banner grabbing. Such activity is detected primarily through IDS software, secondarily through log analysis.	Incident handlers should look for distinct changes in reconnaissance patterns–for example, a sudden interest in a particular port number or host. If this activity points out a vulnerability that could be exploited, the organization may have time to block future attacks by mitigating the vulnerability (e.g., patching a host, disabling an unused service, modifying firewall rules).
A new exploit for gaining unauthorized access is released publicly, and it poses a significant threat to the organization.	The organization should investigate the new exploit and, if possible, alter security controls to minimize the potential impact of the exploit for the organization.
Users report possible social engineering attempts–attackers trying to trick them into revealing sensitive information, such as passwords, or encouraging them to download or run programs and file attachments.	The incident response team should send a bulletin to users with guidance on handling the social engineering attempts. The team should determine what resources the attacker was interested in and look for corresponding log-based precursors because it is likely that the social engineering is only part of the reconnaissance.
A person or system may observe a failed physical access attempt (e.g., outsider attempting to open a locked wiring closet door, unknown individual using a cancelled ID badge).	If possible, security should detain the person. The purpose of the activity should be determined, and it should be verified that the physical and computer security controls are strong enough to block the apparent threat. (An attacker who cannot gain physical access may perform remote computing-based attacks instead.) Physical and computer security controls should be strengthened if necessary.

Table 6-3. Unauthorized Access Indications

Malicious Action	Possible Indications
Root compromise of a host	• Existence of unauthorized security-related tools or exploits • Unusual traffic to and from the host (e.g., attacker may use the host to attack other systems) • System configuration changes, including– o Process/service modifications or additions o Unexpected open ports o System status changes (restarts, shutdowns) o Changes to log and audit policies and data o Network interface card set to promiscuous mode (packet sniffing) o New administrative-level user account or group • Modifications of critical files, timestamps and privileges, including executable programs, OS kernels, system libraries, and configuration and data files • Unexplained account usage (e.g., idle account in use, account in use from multiple locations at once, unexpected commands from a particular user, large number of locked-out accounts) • Significant changes in expected resource usage (e.g., CPU, network activity, full logs, or file systems) • User reports of system unavailability • Network and host intrusion detection alerts • New files or directories with unusual names (e.g., binary characters, leading spaces, leading dots) • Highly unusual operating system and application log messages • Attacker contacts the organization to say that he or she has compromised a host
Unauthorized data modification (e.g., Web server defacement, FTP warez server93)	• Network intrusion detection alerts • Increased resource utilization • User reports of the data modification (e.g., defaced Web site) • Modifications to critical files (e.g., Web pages) • New files or directories with unusual names (e.g., binary characters, leading spaces, leading dots) • Significant changes in expected resource usage (e.g., CPU, network activity, full logs or file systems)
Unauthorized usage of standard user account	• Implement physical security measures that restrict access to critical resources.
Physical intruder	• User reports of network or system unavailability • System status changes (restarts, shutdowns) • Hardware is completely or partially missing (i.e., a system was opened and a particular component removed) • Unauthorized new hardware (e.g., attacker connects a packet sniffing laptop to a network or a modem to a host)
Unauthorized data access (e.g., database of customer information, password files)	• Intrusion detection alerts of attempts to gain access to the data through FTP, HTTP, and other protocols • Host-recorded access attempts to critical files

When prioritizing unauthorized access incidents, determining the current and likely future impact of the incident can be very difficult. Because attackers want to elevate user-level access privileges to administrator-level access, ongoing incidents could potentially result in root-level access. The current impact of the incident may be difficult to judge until extensive analysis has been conducted, and the incident may need to be prioritized before the analysis is complete. Therefore, it is best to

prioritize unauthorized access incidents based on an estimate of the current impact, with the assumption that the impact will become more severe without intervention. Timeframes can then be assigned to each impact category by the criticality of the resources that have been accessed without authorization.

6.4 Containment, Eradication, and Recovery

In addition to the general guidance presented in Section 3.3, this section gives specific recommendations for performing containment, and gathering and handling evidence for unauthorized access incidents.

6.4.1 Choosing a Containment Strategy

Response time is critical when attempting to contain an unauthorized access incident. Extensive analysis may be required to determine exactly what has happened; and in the case of an active attack, the state of things may be changing rapidly. In most cases, it is advisable to perform an initial analysis of the incident, prioritize the incident, implement initial containment measures, and then perform further analysis to determine if the containment measures were sufficient.

Incident handlers walk a fine line when choosing containment strategies because if they assume the worst, the containment strategy could be to shut all networks and systems down. Incident handlers should consider more moderate solutions that focus on mitigating the risks to the extent practical, rather than shutting down the whole environment for days at a time (unless, of course, the extent of the malicious activity is so great that a complete shutdown is merited). An appropriate combination of the following actions should be effective in the initial or final containment of an unauthorized access incident:

- Isolate the affected systems.
- Disable the affected service.
- Eliminate the attacker's route into the environment.
- Disable user accounts that may have been used in the attack.
- Enhance physical security measures.

6.4.2 Evidence Gathering and Handling

When handlers suspect that unauthorized access has been gained to a system, they should make a full image backup of the system. Other relevant data, including host and application logs, intrusion detection alerts, and firewall logs, may provide correlating evidence of the unauthorized access. If a physical security breach occurred during the incident, additional evidence may be available through physical security system logs, security camera tapes, and eyewitness accounts. Unauthorized access incidents are more likely than most other incidents to lead to prosecution, so it is important to follow established evidence gathering and handling procedures and to contact law enforcement if the situation merits their involvement.

6.4.3 Eradication and Recovery

Successful attackers frequently install rootkits, which modify or replace dozens or hundreds of files, including system binaries. Rootkits hide much of what they do, making it tricky to identify what was changed. Therefore, if an attacker appears to have gained root access to a system, handlers cannot trust the OS. Typically, the best solution is to restore the system from a known good backup or reinstall the operating system and applications from scratch, and then secure the system properly. Changing all passwords on the system, and possibly on all systems that have trust relationships with the victim system, is also highly recommended. Some unauthorized access incidents involve the exploitation of multiple vulnerabilities, so it is important for handlers to identify all vulnerabilities that were used and to determine strategies for correcting or mitigating each vulnerability. Other vulnerabilities that are present also should be mitigated, or an attacker may use them instead.

If an attacker only gains a lesser level of access than administrator-level, eradication and recovery actions should be based on the extent to which the attacker gained access. Vulnerabilities that were used to gain access should be mitigated appropriately. Additional actions should be performed as merited to identify and address weaknesses systemically.

6.5 Checklist for Handling Unauthorized Access Incidents

Note that the exact sequence of steps may vary based on the nature of individual incidents and on the strategies chosen by the organization for containing incidents.

Table 6-4. Unauthorized Access Incident Handling Checklist

	Action	Completed
	Detection and Analysis	
1.	Prioritize handling the incident based on the business impact	
1.1	Identify which resources have been affected and forecast which resources will be affected	
1.2	Estimate the current and potential technical effect of the incident	
1.3	Find the appropriate cell(s) in the prioritization matrix, based on the technical effect and affected resources	
2	Report the incident to the appropriate internal personnel and external organizations	
	Containment, Eradication, and Recovery	
3.	Perform an initial containment of the incident	
4.	Acquire, preserve, secure, and document evidence	
5.	Confirm the containment of the incident	
5.1	Further analyze the incident and determine if containment was sufficient (including checking other systems for signs of intrusion)	
5.2	Implement additional containment measures if necessary	

	Action	Completed
6.	Eradicate the incident	
6.1	Identify and mitigate all vulnerabilities that were exploited	
6.2	Remove components of the incident from systems	
7.	Recover from the incident	
7.1	Return affected systems to an operationally ready state	
7.2	Confirm that the affected systems are functioning normally	
7.3	If necessary, implement additional monitoring to look for future related activity	
	Post-Incident Activity	
8.	Create a follow-up report	
9.	Hold a lessons learned meeting	

6.6 Recommendations

The key recommendations presented in this section for handling unauthorized access incidents are summarized below.

- **Configure intrusion detection software to alert on attempts to gain unauthorized access**. Network and host-based intrusion detection software (including file integrity checking software) is valuable for detecting attempts to gain unauthorized access. Each type of software may detect incidents that the other types of software cannot, so the use of multiple types of computer security software is highly recommended.
- **Configure all hosts to use centralized logging**. Incidents are easier to detect if data from all hosts across the organization is stored in a centralized, secured location.
- **Establish procedures for having all users change their passwords**. A password compromise may force the organization to require all users of an application, system, or trust domain—or perhaps the entire organization—to change their passwords.
- **Configure the network perimeter to deny all incoming traffic that is not expressly permitted**. By limiting the types of incoming traffic, attackers should be able to reach fewer targets and should be able to reach the targets using designated protocols only. This should reduce the number of unauthorized access incidents.
- **Secure all remote access methods, including modems and VPNs**. Unsecured modems provide easily attainable unauthorized access to internal systems and networks. Remote access clients are often outside the organization's control, so granting them access to resources increases risk.
- **Put all publicly accessible services on secured DMZ network segments**. This permits the organization to allow external hosts to initiate connections to hosts on the DMZ segments only, not to hosts on internal network segments. This should reduce the number of unauthorized access incidents.

- **Disable all unneeded services on hosts and separate critical services**. Every service that is running presents another potential opportunity for compromise. Separating critical services is important because if an attacker compromises a host that is running a critical service, immediate access should be gained only to that one service.
- **Use host-based firewall software to limit individual hosts' exposure to attacks**. Deploying host-based firewall software to individual hosts and configuring it to deny all activity that is not expressly permitted should further reduce the likelihood of unauthorized access incidents.
- **Create and implement a password policy**. The password policy should require the use of complex, difficult-to-guess passwords and ensure that authentication methods are sufficiently strong for accessing critical resources. Weak and default passwords are likely to be guessed or cracked, leading to unauthorized access.
- **Provide change management information to the incident response team**. Indications such as system shutdowns, audit configuration changes, and executable modifications are probably caused by routine system administration, rather than attacks. When such indications are detected, the team should be able to use change management information to verify that the indications are caused by authorized activity.
- **Select containment strategies that balance mitigating risks and maintaining services**. Incident handlers should consider moderate containment solutions that focus on mitigating the risks as much as is practical while maintaining unaffected services.
- **Restore or reinstall systems that appear to have suffered a root compromise**. The effects of root compromises are often difficult to identify completely. The system should be restored from a known good backup, or the operating system and applications should be reinstalled from scratch. The system should then be secured properly so the incident cannot recur.

7. Handling Inappropriate Usage Incidents

7.1 Incident Definition and Examples

An inappropriate usage incident occurs when a user performs actions that violate acceptable computing use policies. Although such incidents are often not security related, handling them is very similar to handling security-related incidents. Therefore, it has become commonplace for incident response teams to handle many inappropriate usage incidents.

7.2 Preparation

This section provides guidance on preparing to handle inappropriate usage incidents and on preventing inappropriate usage incidents.

7.2.1 Incident Handling Preparation

In addition to the general guidance presented in Sections 3.1.1 and 3.2.3, other actions should be performed while preparing to handle inappropriate usage incidents:

- Meet with representatives of the organization's human resources and legal departments to discuss the handling of inappropriate usage incidents.
- Meet with members of the organization's physical security team to discuss interactions with internal users.
- Discuss liability issues with the organization's public affairs and legal departments, particularly for incidents that are targeted at outside parties.
- Configure network-based IDS software to identify certain types of activity, including—
  - The use of unauthorized services, such as peer-to-peer file and music sharing
  - Spam (e.g., e-mail relaying attempts)
  - Any file activity (e.g., e-mail attachments, FTP transfers, Web requests) with suspicious words in the filename (e.g., "confidential," sexually explicit terms)
  - Outbound reconnaissance activity and attacks.
- Log user activities such as FTP commands, Web requests, and e-mail headers. The goal is to log the basic information on such activities without storing sensitive content, such as e-mail text and file attachments. Organizations should balance privacy considerations with the value of such information for investigative and evidentiary purposes.

7.2.2 Incident Prevention

Generally, little can be done to prevent inappropriate usage incidents from occurring, other than increasing user awareness of appropriate behavior, requiring users to read and sign an acceptable use policy, and informing users that their activities are regularly monitored. However, the following suggested actions might be helpful in reducing certain types of inappropriate usage incidents:

- Configure network devices to prevent the use of services that violate organization policies, such as peer-to-peer file sharing and music sharing services.
- Configure the organization's e-mail servers so that they cannot be used for unauthorized mail relaying, a common way to send spam.
- Implement spam filtering software on all e-mail servers.
- Implement uniform resource locator (URL) filtering to prevent access to inappropriate Web sites.
- Consider limiting outbound connections that use encrypted protocols, such as Secure Shell (SSH), HTTP Secure (HTTPS) and IP Security Protocol (IPsec). Permitting unnecessary encrypted connections may allow users to perform actions that security controls cannot monitor.

7.3 Detection and Analysis

Inappropriate usage incidents are most often detected through user reports, such as seeing inappropriate material on a user's screen or receiving a threatening e-mail. There are usually no precursors of inappropriate usage.

Table 7-1. Inappropriate Usage Indications

Inappropriate Action	Possible Indications
Unauthorized service usage (e.g., Web server, file sharing, music sharing)	• Network intrusion detection alerts • Unusual traffic to and from the host • New process/software installed and running on a host • New files or directories with unusual names (e.g., "warez" server style names) • Increased resource utilization (e.g., CPU, file storage, network activity) • User reports • Application log entries (e.g., Web proxies, FTP servers, e-mail servers)
Access to inappropriate materials (e.g., downloading pornography, sending spam)	• Network intrusion detection alerts • User reports • Application log entries (e.g., Web proxies, FTP servers, e-mail servers) • Inappropriate files on workstations, servers, or removable media
Attack against external party	• Network intrusion detection alerts • Outside party reports • Network, host, and application log entries

The incident response team should be cautious about assisting with reports of inappropriate usage that are not clearly incidents.

Analyzing inappropriate usage incidents is typically straightforward, except for incidents that have been reported by outside parties. The key to analyzing such incidents is determining whether the organization was really the source of the attack or if spoofing has simply created that appearance. This should be fairly easy to determine if proper logging is being performed and the organization has good security controls in place.

Another factor that can complicate incident analysis is when the identity of the person causing the incident cannot be determined by reviewing existing data. Increasing the monitoring of computing or physical resources is usually effective in identifying the individual if the activity continues. A more challenging alternative is to generate a profile of the suspected perpetrator's usage characteristics and goals, and to work with human resources on expanding the profile.

Inappropriate usage incidents are generally easy to prioritize. Unless a crime is involved or the organization's reputation is likely to sustain major damage, these incidents do not need to be handled with the same urgency as other incidents.

Table 7-2. Sample Service Level Agreement for Inappropriate Usage Incidents

Current Impact or Likely Future Impact of the Incident	Nature of Incident	
	Criminal Activity	Noncriminal Activity
Major damage to the organization's reputation	Within 15 minutes, initial response begins Within 1 hour, team contacts public affairs, human resources, legal department, and law enforcement	Within 1 hour, initial response begins Within 2 hours, team contacts public affairs and human resources
Minor damage to the organization's reputation	Within 2 hours, initial response begins Within 4 hours, team contacts human resources, legal department, and law enforcement	Within 4 hours, initial response begins Within 8 hours, team contacts human resources
No damage to the organization's reputation	Within 4 hours, initial response begins Within 8 hours, team contacts human resources, legal department, and law enforcement	Within 1 day, initial response begins Within 2 days, team contacts human resources

There is one caveat to prioritizing inappropriate usage incidents. A sizable number of these incidents are actually follow-on activities to previous incidents, such as a root compromise of a host or a successful malicious code infection. Section 8 discusses prioritizing an incident that encompasses two or more incidents.

7.4 Containment, Eradication, and Recovery

Inappropriate usage incidents typically require no containment, eradication, or recovery actions, other than possibly deleting objectionable materials or uninstalling unauthorized software. For most inappropriate usage incidents, evidence acquisition is important. Evidence may be needed for prosecuting or disciplining an individual and for limiting liability by demonstrating that the organization did its best to prevent, detect, and halt the activity. Evidence storage is particularly important because internal users have physical access to many facilities. Addressing the threat of having evidence altered or destroyed may require coordination with the organization's physical security staff.

7.5 Checklist for Handling Inappropriate Usage Incidents

Note that the sequence of steps may vary based on the nature of individual incidents.

Table 7-3. Inappropriate Usage Incident Handling Checklist

	Action	Completed
	Detection and Analysis	
1.	Prioritize handling the incident based on the business impact	
1.1	Identify which resources have been affected and forecast which resources will be affected	
1.2	Estimate the current and potential technical effect of the incident	
1.3	Find the appropriate cell(s) in the prioritization matrix, based on the technical effect and affected resources	
2	Report the incident to the appropriate internal personnel and external organizations	
	Containment, Eradication, and Recovery	
3.	Acquire, preserve, secure, and document evidence	
4.	If necessary, contain and eradicate the incident (e.g., remove inappropriate materials)	
	Post-Incident Activity	
5.	Create a follow-up report	
6.	Hold a lessons learned meeting	

7.6 Recommendations

The key recommendations presented in this section for handling inappropriate usage incidents are summarized below.

- **Discuss the handling of inappropriate usage incidents with the organization's human resources and legal departments**. Processes for monitoring and logging user activities should comply with the organization's policies and all applicable laws. Procedures for handling incidents that directly involve employees should incorporate discretion and confidentiality.
- **Discuss liability issues with the organization's legal department**. Liability issues may arise during inappropriate usage incidents, particularly for incidents that are targeted at outside parties. Incident handlers should understand when they should discuss incidents with the allegedly attacked party and what information they should reveal.
- **Configure network-based intrusion detection software to detect certain types of inappropriate usage**. Intrusion detection software has built-in capabilities to detect certain inappropriate usage incidents, such as the use of unauthorized services, outbound reconnaissance activity and attacks, and improper e-mail relay usage (e.g., sending spam).

- **Log basic information on user activities**. Basic information on user activities (e.g., FTP commands, Web requests, and e-mail headers) may be valuable for investigative and evidentiary purposes.
- **Configure all e-mail servers so they cannot be used for unauthorized mail relaying**. Mail relaying is commonly used to send spam.
- **Implement spam filtering software on all e-mail servers**. Spam filtering software can block much of the spam sent by external parties to the organization's users, as well as spam sent by internal users.
- **Implement URL filtering software**. URL filtering software prevents access to many inappropriate Web sites. Users should be required to use the software, typically by preventing access to external Web sites unless the traffic passes through a server that performs URL filtering.

8. Handling Multiple Component Incidents

8.1 Incident Definition and Examples

A multiple component incident is a single incident that encompasses two or more incidents.

1. Malicious code spread through e-mail compromises an internal workstation.
2. An attacker (who may or may not be the one who sent the malicious code) uses the infected workstation to compromise additional workstations and servers.
3. An attacker (who may or may not have been involved in Steps 1 or 2) uses one of the compromised hosts to launch a DDoS attack against another organization.

8.2 Preparation, Detection, and Analysis

Multiple component incidents are often difficult to analyze. Incident handlers may know about one portion of the incident only but may not realize that the incident is composed of several stages. Handlers may also be aware of multiple incidents but not realize that they are related. Further complicating the analysis is that the incident's stages may occur over a period of weeks or months. Unless the organization has excellent logging and log archiving processes in place, the evidence of earlier stages of the incident may be gone. Even if the data is available, it can be challenging for the analyst to determine which indications are related among all the data.

The main preparation for handling multiple component incidents is the same as that previously noted for each individual incident category. Another helpful activity is to conduct exercises in which the incident response team reviews scenarios involving multiple component incidents. The use of centralized logging and correlation software has already been recommended for facilitating more efficient incident analysis. This is particularly true for analyzing multiple component incidents, which typically have several precursor and indication sources. Incident handlers should diagnose an incident as having multiple components more quickly if all precursors and indications are accessible from a single point of view.

8.3 Containment, Eradication, and Recovery

Every incident that is detected could be a multiple component incident, but it could take an extended period of time for a handler to authoritatively determine that an incident has only a single component. Meanwhile, the initial incident has not been contained. It is generally better to contain the initial incident and then search for signs of other components. Experienced handlers should be able to make an educated guess as to whether an incident has other components. It can be generally assumed that unauthorized access incidents are more likely to have multiple components, and other types of incidents are less likely to have multiple components.

Handlers who are aware of multiple components of an incident should separately prioritize the handling of each component because not enough resources will likely be available to handle all components simultaneously. If the organization has created prioritization guidelines that address all incident categories, the handler can identify the specified response time for each component and handle the most urgent need

first. Another factor to consider is how current each component is—a DoS attack in progress should usually be addressed more quickly than a malicious code infection that occurred six weeks ago. Furthermore, if one component creates a path for attackers to reach targets, the handler may be able to contain the whole incident by containing just that one component. (Note that other components will still need to be handled, just not as urgently.) Handlers should be cautious, though, because attackers may have created or discovered additional paths to the targets.

8.4 Checklist for Handling Multiple Component Incidents

Note that the sequence of steps may vary based on the nature of individual incidents and on the strategies chosen by the organization for containing incidents.

Table 8-1. Multiple Component Incident Handling Checklist

	Action	Completed
	Detection and Analysis	
1.	Prioritize handling the incident based on the business impact	
1.1	Identify which resources have been affected and forecast which resources will be affected	
1.2	Estimate the current and potential technical effect of the incident	
1.3	Find the appropriate cell(s) in the prioritization matrix, based on the technical effect and affected resources	
2	Report the incident to the appropriate internal personnel and external organizations	
	Containment, Eradication, and Recovery	
3.	Follow the Containment, Eradication, and Recovery steps for each component, based on the results of Step 1	
	Post-Incident Activity	
4.	Create a follow-up report	
5.	Hold a lessons learned meeting	

8.5 Recommendations

The key recommendations presented in this section for handling multiple component incidents are summarized below.

- **Use centralized logging and event correlation software**. Incident handlers should identify an incident as having multiple components more quickly if all precursors and indications are accessible from a single point of view.
- **Contain the initial incident and then search for signs of other incident components**. It can take an extended period of time for a handler to authoritatively determine that an incident has only a single component; meanwhile, the initial incident has not been contained. It is usually better to contain the initial incident first.
- **Separately prioritize the handling of each incident component**. Resources are probably too limited to handle all incident components simultaneously. Components should be prioritized based on response guidelines for each component and how current each component is.

Appendix A – Recommendations

The first group of recommendations applies to organizing an incident response capability. The remaining recommendations have been grouped by the phases of the incident response life cycle—preparation; detection and analysis; containment, eradication, and recovery; and post-incident activity.

A.1 Organizing a Computer Security Incident Response Capability

- **Establish a formal incident response capability**. Organizations should be prepared to respond quickly and effectively when computer security defenses are breached. The Federal Information Security Management Act (FISMA) requires Federal agencies to establish incident response capabilities.

A.1.1 Incident Response Policy and Procedure Creation

- **Create an incident response policy and use it as the basis for incident response procedures**. The incident response policy is the foundation of the incident response program. It defines which events are considered incidents, establishes the organizational structure for incident response, defines roles and responsibilities, and lists the requirements for reporting incidents, among other items.
- **Establish policies and procedures regarding incident-related information sharing**. The organization will want or be required to communicate incident details with outside parties, such as the media, law enforcement agencies, and incident reporting organizations. The incident response team should discuss this requirement at length with the organization's public affairs staff, legal advisors, and management to establish policies and procedures regarding information sharing. The team should comply with existing organization policy on interacting with the media and other outside parties.
- **Provide pertinent information on incidents to the appropriate incident reporting organization**. Federal civilian agencies are required to report incidents to the Federal Computer Incident Response Center (FedCIRC). Reporting benefits the agencies because the incident reporting organizations use the reported data to provide information to the agencies regarding new threats and incident trends.

A.1.2 Incident Response Team Structure and Services

- **Consider the relevant factors when selecting an appropriate incident response team model**. Organizations should carefully weigh the advantages and disadvantages of each possible team structure model and staffing model in the context of the organization's needs and available resources.
- **Select people with appropriate skills for the incident response team**. The credibility and proficiency of the team depend largely on the technical skills of its members. Poor technical judgment can undermine the team's credibility and cause incidents to worsen. Critical technical skills include system administration, network administration, programming, technical support, and intrusion detection. Teamwork and communications skills are also needed for effective incident handling.

- **Identify other groups within the organization that may need to participate in incident handling**. Every incident response team relies on the expertise and judgment of other teams, including management, information security, information technology (IT) support, legal, public affairs, and facilities management.
- **Determine which services the team should offer**. Although the main focus of the team is incident response, most teams perform additional functions. Examples include distributing security advisories, performing vulnerability assessments, educating users on security, and monitoring intrusion detection sensors.

A.2 Preparation

- **Acquire tools and resources that may be of value during incident handling**. The team will be more efficient at handling incidents if various tools and resources are already available to them. Examples include contact lists, encryption software, network diagrams, backup devices, computer forensic software, port lists, and security patches.
- **Prevent incidents from occurring by ensuring that networks, systems, and applications are sufficiently secure**. Preventing incidents is beneficial to the organization and reduces the workload of the incident response team. Performing periodic risk assessments and reducing the identified risks to an acceptable level are effective in reducing the number of incidents. User, IT staff, and management awareness of security policies and procedures is also very important.

A.2.1 Denial of Service Incidents

- **Configure firewall rulesets to prevent reflector attacks**. Most reflector attacks can be stopped through network-based and host-based firewall rulesets that reject suspicious combinations of source and destination ports.
- **Configure border routers to prevent amplifier attacks**. Amplifier attacks can be blocked by configuring border routers not to forward directed broadcasts.
- **Determine how the organization's Internet service providers (ISP) and second-tier providers can assist in handling network-based DoS attacks**. ISPs can often filter or limit certain types of traffic, slowing or halting a DoS attack. They can also provide logs of DoS traffic and may be able to assist in tracing the source of the attack. The organization should meet with the ISPs in advance to establish procedures for requesting such assistance.
- **Configure security software to detect DoS attacks**. Intrusion detection software can detect many types of DoS activity. Establishing network and system activity baselines, and monitoring for significant deviations from those baselines, can also be useful in detecting attacks.
- **Configure the network perimeter to deny all incoming and outgoing traffic that is not expressly permitted**. By restricting the types of traffic that can enter and leave the environment, the organization will limit the methods that attackers can use to perform DoS attacks.

A.2.2 Malicious Code Incidents

- **Make users aware of malicious code issues**. Users should be familiar with the methods that malicious code uses to propagate and the symptoms of infections. Holding regular user education sessions helps to ensure that users are aware of the risks that malicious code poses. Teaching users how to safely handle e-mail attachments should reduce the number of infections that occur.
- **Read antivirus bulletins**. Bulletins regarding new malicious code threats provide timely information to incident handlers.
- **Deploy host-based intrusion detection systems, including file integrity checkers, to critical hosts**. Host-based IDS software, particularly file integrity checkers, can detect signs of malicious code incidents, such as configuration changes and modifications to executables.
- **Use antivirus software, and keep it updated with the latest virus signatures**. Antivirus software should be deployed to all hosts and all applications that may be used to transfer malicious code. The software should be configured to detect and disinfect or quarantine malicious code infections. All antivirus software should be kept current with the latest virus signatures so the newest threats can be detected.
- **Configure software to block suspicious files**. Files that are very likely to be malicious should be blocked from the environment, such as those with file extensions that are usually associated with malicious code and files with suspicious combinations of file extensions.
- **Eliminate open Windows shares**. Many worms spread through unsecured shares on hosts running Windows. A single infection may rapidly spread to hundreds or thousands of hosts through unsecured shares.

A.2.3 Unauthorized Access Incidents

- **Configure intrusion detection software to alert on attempts to gain unauthorized access**. Network and host-based intrusion detection software (including file integrity checking software) is valuable for detecting attempts to gain unauthorized access. Each type of software may detect incidents that the other types of software cannot, so the use of multiple types of computer security software is highly recommended.
- **Configure all hosts to use centralized logging**. Incidents are easier to detect if data from all hosts across the organization is stored in a centralized, secured location.
- **Establish procedures for having all users change their passwords**. A password compromise may force the organization to require all users of an application, system, or trust domain—or perhaps the entire organization—to change their passwords.
- **Configure the network perimeter to deny all incoming traffic that is not expressly permitted**. By limiting the types of incoming traffic, attackers should be able to reach fewer targets and should be able to reach the

targets using designated protocols only. This should reduce the number of unauthorized access incidents.

- **Secure all remote access methods, including modems and virtual private networks (VPN)**. Unsecured modems provide easily attainable unauthorized access to internal systems and networks. Remote access clients are often outside the organization's control, so granting them access to resources increases risk.
- **Put all publicly accessible services on secured demilitarized zone (DMZ) network segments**. This action permits the organization to allow external hosts to initiate connections to hosts on the DMZ segments only, not to hosts on internal network segments. This should reduce the number of unauthorized access incidents.
- **Disable all unneeded services on hosts and separate critical services**. Every service that is running presents another potential opportunity for compromise. Separating critical services is important because if an attacker compromises a host that is running a critical service, immediate access should be gained only to that one service.
- **Use host-based firewall software to limit individual hosts' exposure to attacks**. Deploying host-based firewall software to individual hosts and configuring it to deny all activity that is not expressly permitted should further reduce the likelihood of unauthorized access incidents.
- **Create and implement a password policy**. The password policy should require the use of complex, difficult-to-guess passwords and should ensure that authentication methods are sufficiently strong for accessing critical resources. Weak and default passwords are likely to be guessed or cracked, leading to unauthorized access.

A.2.4 Inappropriate Usage Incidents

- **Discuss the handling of inappropriate usage incidents with the organization's human resources and legal departments**. Processes for monitoring and logging user activities should comply with the organization's policies and all applicable laws. Procedures for handling incidents that directly involve employees should incorporate discretion and confidentiality.
- **Discuss liability issues with the organization's legal departments**. Liability issues may arise during inappropriate usage incidents, particularly for incidents that are targeted at outside parties. Incident handlers should understand when they should discuss incidents with the allegedly attacked party and what information they should reveal.
- **Configure network-based intrusion detection software to detect certain types of inappropriate usage**. Intrusion detection software has built-in capabilities to detect certain inappropriate usage incidents, such as the use of unauthorized services, outbound reconnaissance activity and attacks, and improper mail relay usage (e.g., sending spam).

- **Log basic information on user activities**. Basic information on user activities such as File Transfer Protocol (FTP) commands, Web requests, and e-mail headers may be valuable for investigative and evidentiary purposes.
- **Configure all e-mail servers so they cannot be used for unauthorized mail relaying**. Mail relaying is commonly used to send spam.
- **Implement spam filtering software on all e-mail servers**. Spam filtering software can block much of the spam sent by external parties to the organization's users and spam sent by internal users.
- **Implement uniform resource locator (URL) filtering software**. URL filtering software prevents access to many inappropriate Web sites. Users should be required to use the software, typically by preventing access to external Web sites unless the traffic passes through a server that performs URL filtering.

A.2.5 Multiple Component Incidents

- **Use centralized logging and event correlation software**. Incident handlers should identify an incident as having multiple components more quickly if all precursors and indications are accessible from a single point of view.

A.3 Detection and Analysis

- **Identify precursors and indications through alerts generated by several types of computer security software**. Network and host-based intrusion detection systems, antivirus software, and file integrity checking software are valuable for detecting signs of incidents. Each type of software may detect incidents that the other types of software cannot, so the use of several types of computer security software is highly recommended. Third-party monitoring services can also be helpful.
- **Establish mechanisms for outside parties to report incidents**. Outside parties may want to report incidents to the organization; for example, they may believe that one of the organization's users is attacking them. Organizations should publish a phone number and e-mail address that outside parties can use to report such incidents.
- **Require a baseline level of logging and auditing on all systems, and a higher baseline level on all critical systems**. Logs from operating systems, services, and applications frequently provide value during incident analysis, particularly if auditing was enabled. The logs can provide information such as which accounts were accessed and what actions were performed.
- **Profile networks and systems**. Profiling measures the characteristics of expected activity levels so that changes in patterns can be more easily identified. If the profiling process is automated, deviations from expected activity levels can be detected and reported to administrators quickly, leading to faster detection of incidents and operational issues.

- **Understand the normal behaviors of networks, systems, and applications**. Team members who understand what normal behavior is should be able to recognize abnormal behavior more easily. This knowledge can best be gained by reviewing log entries and security alerts; the handlers should become familiar with the typical data and can investigate the unusual entries to gain more knowledge.
- **Use centralized logging and create a log retention policy**. Information regarding an incident may be recorded in several places. Organizations should deploy centralized logging servers and configure devices to send duplicates of their log entries to the centralized servers. The team benefits because it can access all log entries at once; also, changes made to logs on individual hosts will not affect the data already sent to the centralized servers. A log retention policy is important because older log entries may show previous instances of similar or related activity.
- **Perform event correlation**. Indications of an incident may be captured in several logs. Correlating events among multiple sources can be invaluable in collecting all the available information for an incident and validating whether the incident occurred. Centralized logging makes event correlation easier and faster.
- **Keep all host clocks synchronized**. If the devices reporting events have inconsistent clock settings, event correlation will be more difficult. Clock discrepancies may also cause issues from an evidentiary standpoint.
- **Maintain and use a knowledge base of information**. Handlers need to reference information quickly during incident analysis; a centralized knowledge base provides a consistent, maintainable source of information. The knowledge base should include general information, such as commonly used port numbers and links to virus information, and data on precursors and indications of previous incidents.
- **Create a diagnosis matrix for less experienced staff**. Help desk staff, system administrators, and new incident response team members may need assistance in determining what type of incident may be occurring. A diagnosis matrix that lists incident categories and the symptoms associated with each category can provide guidance as to what type of incident is occurring and how the incident can be validated.
- **Start recording all information as soon as the team suspects that an incident has occurred**. Every step taken, from the time the incident was detected to its final resolution, should be documented and timestamped. Information of this nature can serve as evidence in a court of law if legal prosecution is pursued. Recording the steps performed can also lead to a more efficient and systematic, and less error-prone handling of the problem.
- **Safeguard incident data**. It often contains sensitive information regarding such elements as vulnerabilities, security breaches, and users that may have performed inappropriate actions. The team should ensure that access to incident data is restricted properly, both logically and physically.

- **Prioritize incidents by business impact, based on the criticality of the affected resources and the technical impact of the incident**. Because of resource limitations, incidents should not be handled on a first-come, first-served basis. Instead, organizations should establish written guidelines that outline how quickly the team must respond to the incident and what actions should be performed, based on the incident's current and potential business impact. This guidance saves time for the incident handlers and provides a justification to management and system owners for their actions. Organizations should also establish an escalation process for those instances when the team does not respond to an incident within the designated time.
- **Include provisions regarding incident reporting in the organization's incident response policy**. Organizations should specify which incidents must be reported, when they must be reported, and to whom. The parties most commonly notified are the chief information officer (CIO), head of information security, local information security officer, other incident response teams within the organization, and system owners.

A.4 Containment, Eradication, and Recovery

- **Establish strategies and procedures for containing incidents**. It is important to contain incidents quickly and effectively to limit their business impact. Organizations should define acceptable risks in containing incidents and develop strategies and procedures accordingly. Containment strategies should vary based on the type of incident.
- **Follow established procedures for evidence gathering and handling**. The team should clearly document how all evidence has been preserved. Evidence should be accounted for at all times. The team should meet with legal staff and law enforcement agencies to discuss evidence handling, then develop procedures based on those discussions.
- **Capture volatile data from systems as evidence**. This effort includes lists of network connections, processes, login sessions, open files, network interface configurations, and the contents of memory. Running carefully chosen commands from trusted media can collect the necessary information without damaging the system's evidence.
- **Obtain system snapshots through full forensic disk images, not file system backups**. Disk images should be made to sanitized write-protectable or write-once media. This process is superior to a file system backup for investigatory and evidentiary purposes. Imaging is also valuable in that it is much safer to analyze an image than it is to perform analysis on the original system because the analysis may inadvertently alter the original.

A.4.1 Denial of Service Incidents

- **Create a containment strategy that includes several solutions in sequence**. The decision-making process for containing DoS incidents is easier if recommended solutions are predetermined. Because the effectiveness of each possible solution will vary among incidents, organizations should select several

solutions and determine the sequence in which the solutions should be attempted.

A.4.2 Malicious Code Incidents

- **Contain malicious code incidents as quickly as possible**. Because malicious code works surreptitiously and can propagate to other systems rapidly, early containment of a malicious code incident is needed to stop it from spreading and causing further damage. Infected systems should be disconnected from the network immediately. Organizations may need to block malicious code at the e-mail server level, or even temporarily suspend e-mail services to gain control over serious e-mail-borne malicious code incidents.

A.4.3 Unauthorized Access Incidents

- **Provide change management information to the incident response team**. Indications such as system shutdowns, audit configuration changes, and executable modifications are probably caused by routine system administration, rather than attacks. When such indications are detected, the team should be able to use change management information to verify that the indications are caused by authorized activity.
- **Select containment strategies that balance mitigating risks and maintaining services**. Incident handlers should consider moderate containment solutions that focus on mitigating the risks as much as is practical while maintaining unaffected services.
- **Restore or reinstall systems that appear to have suffered a root compromise**. The effects of root compromises are often difficult to identify completely. The system should be restored from a known good backup, or the operating system and applications should be reinstalled from scratch. The system should then be secured properly so the incident cannot recur.

A.4.4 Multiple Component Incidents

- **Contain the initial incident and then search for signs of other incident components**. It can take an extended period of time for a handler to authoritatively determine that an incident has only a single component; meanwhile, the initial incident has not been contained. It is generally better to contain the initial incident first.

A.5 Post-Incident Activity

- **Hold lessons learned meetings after major incidents**. Lessons learned meetings are extremely helpful in improving security measures and the incident handling process itself.

A.5.1 Unauthorized Access Incidents

- **Separately prioritize the handling of each incident component**. Resources are probably too limited to handle all incident components simultaneously. Components should be prioritized based on response guidelines for each component and how current each component is.

Section 2 – Special Publication 800-86

Guide to Integrating Forensic Techniques into Incident Response

Recommendations of the National Institute of Standards and Technology

Karen Kent
Suzanne Chevalier
Tim Grance
Hung Dang

August 2006
Abridged by Guidance Software, Inc.

Because different organizations are subject to different laws and regulations, this publication should not be used as a guide to executing a digital forensic investigation, construed as legal advice, or used as the basis for investigations of criminal activity. Instead, organizations should use this guide as a starting point for developing a forensic capability in conjunction with extensive guidance provided by legal advisors, law enforcement officials, and management.

This guide provides general recommendations for performing the forensic process. It also provides detailed information about using the analysis process with four major categories of data sources: files, operating systems, network traffic, and applications. The guide focuses on explaining the basic components and characteristics of data sources within each category, as well as techniques for the collection, examination, and analysis of data from each category. The guide also provides recommendations for how multiple data sources can be used together to gain a better understanding of an event.

Implementing the following recommendations should facilitate efficient and effective digital forensic activities for Federal departments and agencies.

Organizations should ensure that their policies contain clear statements addressing all major forensic considerations, such as contacting law enforcement, performing monitoring, and conducting regular reviews of forensic policies and procedures.

Organizations should create and maintain procedures and guidelines for performing forensic tasks, based on the organization's policies and all applicable laws and regulations.

Organizations should ensure that their policies and procedures support the reasonable and appropriate use of forensic tools.

Organizations should ensure that their IT professionals are prepared to participate in forensic activities.

1. Introduction

1.1 Authority

(Refer to the prefix of this guide)

1.2 Purpose and Scope

This publication is intended to help organizations in investigating computer security incidents and troubleshooting some information technology (IT) operational problems by providing practical guidance on performing computer and network forensics. **The guide presents forensics from an IT view, not a law enforcement view.** Specifically, the publication describes the processes for performing effective forensics activities and provides advice regarding different data sources, including files, operating systems (OS), network traffic, and applications.

The publication is not to be used as an all-inclusive step-by-step guide for executing a digital forensic investigation or construed as legal advice. Its purpose is to inform readers of various technologies and potential ways of using them in performing incident

response or troubleshooting activities. Readers are advised to apply the recommended practices only after consulting with management and legal counsel for compliance concerning laws and regulations (i.e., local, state, Federal, and international) that pertain to their situation.

1.3 Audience

This publication has been created for incident response teams; forensic analysts; system, network, and security administrators; and computer security program managers who are responsible for performing forensics for investigative, incident response, or troubleshooting purposes. The practices recommended in this guide are designed to highlight key principles associated with the handling and examination of electronic evidence. Because of the constantly changing nature of electronic devices and software, and forensic procedures and tools, readers are expected to refer to other resources, including those listed in this guide, for more current and detailed information than that presented in this guide.

2. Establishing and Organizing a Forensics Capability

2.1 The Need for Forensics

Over the last decade, the number of crimes that involve computers has grown, spurring an increase in companies and products that aim to assist law enforcement in using computer-based evidence to determine the who, what, where, when, and how for crimes. As a result, computer and network forensics has evolved to assure proper presentation of computer crime evidentiary data into court. Forensic tools and techniques are most often thought of in the context of criminal investigations and computer security incident handling used to respond to an event by investigating suspect systems, gathering and preserving evidence, reconstructing events, and assessing the current state of an event. However, forensic tools and techniques are also useful for many other types of tasks, such as the following:

- **Operational Troubleshooting.**
- **Log Monitoring.**
- **Data Recovery.**
- **Data Acquisition.**
- **Due Diligence/Regulatory Compliance.**

Regardless of the situation, the forensic process comprises the following basic phases:

- **Collection**. The first phase in the process is to identify, label, record, and acquire data from the possible sources of relevant data, while following guidelines and procedures that preserve the integrity of the data. Collection is typically performed in a timely manner because of the likelihood of losing dynamic data such as current network connections, as well as losing data from battery-powered devices (e.g., cell phones, PDAs).
- **Examination**. Examinations involve forensically processing large amounts of

collected data using a combination of automated and manual methods to assess and extract data of particular interest, while preserving the integrity of the data.

- **Analysis**. The next phase of the process is to analyze the results of the examination, using legally justifiable methods and techniques, to derive useful information that addresses the questions that were the impetus for performing the collection and examination.
- **Reporting.** The final phase is reporting the results of the analysis, which may include describing the actions used, explaining how tools and procedures were selected, determining what other actions need to be performed (e.g., forensic examination of additional data sources, securing identified vulnerabilities, improving existing security controls), and providing recommendations for improvement to policies, guidelines, procedures, tools, and other aspects of the forensic process. The formality of the reporting step varies greatly depending on the situation.

A more in-depth discussion of the forensic process is presented in Section 3. Sections 4 through 7 provide additional information on collecting, examining, and analyzing different types of forensic data.

2.2 Forensic Staffing

Practically every organization needs to have some capability to perform computer and network forensics. Without such a capability, an organization will have difficulty determining what events have occurred within its systems and networks, such as exposures of protected, sensitive data. Although the extent of this need varies, the primary users of forensic tools and techniques within an organization usually can be divided into the following three groups:

- **Investigators.** Investigators within an organization are most often from the Office of Inspector General (OIG), and they are responsible for investigating allegations of misconduct.
- **IT Professionals**. This group includes technical support staff and system, network, and security administrators.
- **Incident Handlers.** This group responds to a variety of computer security incidents, such as unauthorized data access, inappropriate system usage, malicious code infections, and denial of service attacks.

Many organizations rely on a combination of their own staff and external parties to perform forensic tasks. For example, some organizations perform standard tasks themselves and use outside parties only when specialized assistance is needed.

When deciding which internal or external parties should handle each aspect of forensics, organizations should keep the following factors in mind:

- **Cost**. There are many potential costs. Software, hardware, and equipment used to collect and examine data may carry significant costs (e.g., purchase price, software updates and upgrades, maintenance), and may also require additional

physical security measures to safeguard them from tampering. Other significant expenses involve staff training and labor costs, which are particularly significant for dedicated forensic specialists.

- **Response Time**. Personnel located on-site might be able to initiate computer forensic activity more quickly than could off-site personnel.
- **Data Sensitivity.** Because of data sensitivity and privacy concerns, some organizations might be reluctant to allow external parties to image hard drives and perform other actions that provide access to data.

Incident handlers performing forensic tasks need to have a reasonably comprehensive knowledge of forensic principles, guidelines, procedures, tools, and techniques, as well as anti-forensic tools and techniques that could conceal or destroy data.

On an incident handling team, more than one team member should be able to perform each typical forensic activity so that the absence of any single team member will not severely impact the team.s abilities. Hands-on exercises and external IT and forensic training courses can also be helpful in building and maintaining skills. In addition, it might be beneficial to have team members see demonstrations of new tools and technologies or try out forensic and anti-forensic tools in a lab. Incident handlers need to stay current with new forensic technologies, techniques, and procedures.

2.3 Interactions with Other Teams

It is not feasible for any one person to be well-versed in every technology (including all software) used within an organization; therefore, individuals performing forensic actions should be able to reach out to other teams and individuals within their organization as needed for additional assistance. Organizations should ensure that IT professionals throughout the organization, especially incident handlers and other first responders to incidents, understand their roles and responsibilities for forensics, receive ongoing training and education on forensic-related policies, guidelines, and procedures, and are prepared to cooperate with and assist others when the technologies that they are responsible for are part of an incident or other event.

In addition to IT professionals and incident handlers, others within an organization may also need to participate in forensic activities in a less technical capacity. Examples include management, legal advisors, human resources personnel, auditors, and physical security staff. Legal advisors should carefully review all forensic policy and high-level guidelines and procedures, and they can provide additional guidance when needed to ensure that forensic actions are performed lawfully.

2.4 Policies

At a high level, policies should allow authorized personnel to monitor systems and networks and perform investigations for legitimate reasons under appropriate circumstances. Organizations may also have a separate policy for incident handlers and others with forensic roles; this policy would provide more detailed rules for appropriate behavior.

2.4.1 Defining Roles and Responsibilities

The policy should clearly indicate who should contact which internal teams and external organizations under different circumstances. The policy should also discuss jurisdictional conflicts —a crime that involves multiple jurisdictions, which could be investigated by multiple law enforcement agencies —and explain how to resolve them. As mentioned in Section 2.2, some organizations have an Office of Inspector General (OIG) that is responsible for investigating allegations of misconduct; the OIG may also be well-suited to resolving jurisdictional conflicts. In some organizations, if a crime may have been committed, the OIG immediately takes over the investigation.

2.4.2 Providing Guidance for Forensic Tool Use

To ensure that tools are used reasonably and appropriately, the organization's policies, guidelines, and procedures should clearly explain what forensic actions should and should not be performed under various circumstances. Policies, guidelines, and procedures should clearly define the specific actions that are permitted and forbidden for each applicable role under normal circumstances (e.g., typical duties) and special circumstances (e.g., incident handling).

Because forensic tools may record sensitive information, policies, guidelines, and procedures should also describe the necessary safeguards for the information. There should also be requirements for handling inadvertent exposures of sensitive information, such as an incident handler seeing passwords or patient medical information.

2.4.3 Supporting Forensics in the Information System Life Cycle

Many incidents can be handled more efficiently and effectively if forensic considerations have been incorporated into the information system life cycle. Examples of such considerations are as follows:

- Performing regular backups of systems and maintaining previous backups for a specific period of time
- Enabling auditing on workstations, servers, and network devices
- Forwarding audit records to secure centralized log servers
- Configuring mission-critical applications to perform auditing, including recording all authentication attempts
- Maintaining a database of file hashes for the files of common OS and application deployments, and using file integrity checking software on particularly important assets
- Maintaining records (e.g., baselines) of network and system configurations
- Establishing data retention policies that support performing historical reviews of system and network activity, complying with requests or requirements to preserve data relating to ongoing litigation and investigations, and destroying data that is no longer needed.

2.5 Guidelines and Procedures

As mentioned in Section 2.4, an organization should create and maintain guidelines and procedures for performing forensic tasks, based on the organization's policies, incident response staffing models, and other teams identified as participants in forensic activities. Even if the activities are performed by external parties, the organization's internal staff will still interact with them and participate to some extent in the activities, such as notifying the external party of a need for assistance, granting physical or logical access to systems, and securing an incident scene until an investigator arrives. The internal staff should work closely with the external parties to ensure that the organization's policies, guidelines, and procedures are understood and followed.

An organization's forensic guidelines should include general methodologies for investigating an incident using forensic techniques, since it is not feasible to develop comprehensive procedures tailored to every possible situation. However, organizations also should consider developing step-by-step procedures for performing routine tasks, such as imaging a hard disk, capturing and recording volatile information from systems, or securing physical evidence (e.g., removable media).

The guidelines and procedures should support the admissibility of evidence into legal proceedings, including information on gathering and handling evidence properly, preserving the integrity of tools and equipment, maintaining the chain of custody, and storing evidence securely. Although it may not be feasible to record every event or action taken in response to an incident, having a record of the major events and actions taken helps ensure that nothing has been overlooked, and helps explain to others how the incident was handled. This documentation can be useful for case management, report writing, and testifying. Also, handling evidence in a forensically sound manner puts decision makers in a position where they can confidently take the necessary actions.

It is also important to maintain the guidelines and procedures once they are created so that they remain accurate.

2.6 Recommendations

The key recommendations on establishing and organizing a forensic capability are as follows:

- **Organizations should have a capability to perform computer and network forensics.** Forensics is needed for various tasks within an organization, including investigating crimes and inappropriate behavior, reconstructing computer security incidents, troubleshooting operational problems, supporting due diligence for audit record maintenance, and recovering from accidental system damage. Without such a capability, an organization will have difficulty determining what events have occurred within its systems and networks, such as exposures of protected, sensitive data. Also, handling evidence in a forensically sound manner puts decision makers in a position where they can confidently take the necessary actions.

- **Organizations should determine which parties should handle each aspect of forensics**. Most organizations rely on a combination of their own staff and external parties to perform forensic tasks. Organizations should decide which parties should take care of which tasks based on skills and abilities, cost, response time, and data sensitivity.
- **Incident handling teams should have robust forensic capabilities.** More than one team member should be able to perform each typical forensic activity. Hands-on exercises and IT and forensic training courses can be helpful in building and maintaining skills, as can demonstrations of new tools and technologies.
- **Many teams within an organization should participate in forensics.** Individuals performing forensic actions should be able to reach out to other teams and individuals within an organization, as needed, for additional assistance. Examples of teams that may provide assistance in these efforts include IT professionals, management, legal advisors, human resources personnel, auditors, and physical security staff. Members of these teams should understand their roles and responsibilities in forensics, receive training and education on forensic-related policies, guidelines, and procedures, and be prepared to cooperate with and assist others on forensic actions.

This document does not describe the computer and network forensic requirements placed on law enforcement. For further information regarding computer and network forensic requirements for law enforcement, see Electronic Crime Scene Investigation: A Guide for First Responders and Forensic Examination of Digital Evidence: A Guide for Law Enforcement, which are both available at http://www.ncjrs.gov/app/topics/topic.aspx?topicid=.

- **Forensic considerations should be clearly addressed in policies.**
  - Forensic policy should clearly define the roles and responsibilities of all people performing or assisting with the organization's forensic activities.
  - The organization's policies, guidelines, and procedures should clearly explain what forensic actions should and should not be performed under normal and special circumstances and should address the use of anti-forensic tools and techniques. Policies, guidelines, and procedures should also address the handling of inadvertent exposures of sensitive information.
  - Incorporating forensic considerations into the information system life cycle can lead to more efficient and effective handling of many incidents.
- **Organizations should create and maintain guidelines and procedures for performing forensic tasks.** The guidelines should include general methodologies for investigating an incident using forensic techniques, and step-by-step procedures should explain how to perform routine tasks. The guidelines and procedures should support the admissibility of evidence into legal proceedings.

3. Performing the Forensic Process

This section describes the basic phases of the forensic process: collection, examination, analysis, and reporting.8 During collection, data related to a specific event is identified, labeled, recorded, and collected, and its integrity is preserved. In the second phase, examination, forensic tools and techniques appropriate to the types of data that were collected are executed to identify and extract the relevant information from the collected data while protecting its integrity. Examination may use a combination of automated tools and manual processes. The next phase, analysis, involves analyzing the results of the examination to derive useful information that addresses the questions that were the impetus for performing the collection and examination. The final phase involves reporting the results of the analysis, which may include describing the actions performed, determining what other actions need to be performed, and recommending improvements to policies, guidelines, procedures, tools, and other aspects of the forensic process.

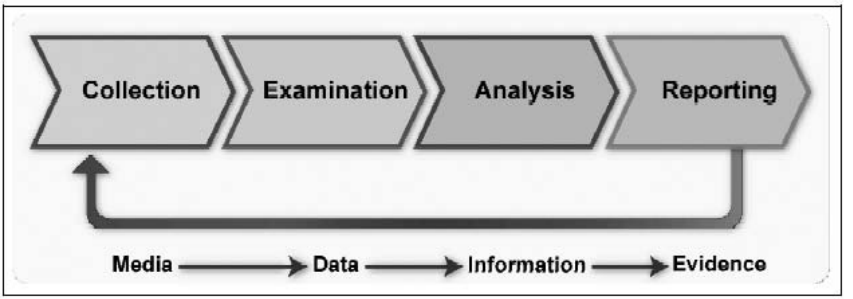

Figure 3-1. Forensic Process

3.1 Data Collection

The first step in the forensic process is to identify potential sources of data and acquire data from them.

3.1.1 Identifying Possible Sources of Data

The increasingly widespread use of digital technology for both professional and personal purposes has led to an abundance of data sources. The most obvious and common sources of data are desktop computers, servers, network storage devices, and laptops. These systems typically have internal drives that accept media, such as CDs and DVDs, and also have several types of ports (e.g., Universal Serial Bus [USB], Firewire, Personal Computer Memory Card International Association [PCMCIA]) to which external data storage media and devices can be attached. Analysts should be able to survey a physical area, such as an office, and recognize the possible sources of data.

Organizations can take ongoing proactive measures to collect data that may be useful for forensic purposes. For example, as described in Section 5.1.1, most OSs can be configured to audit and record certain types of events, such as authentication attempts

and security policy changes, as part of normal operations. Audit records can provide valuable information, including the time that an event occurred and the origin of the event. Another helpful action is to implement centralized logging, which means that certain systems and applications forward copies of their logs to secure central log servers. Centralized logging prevents unauthorized users from tampering with logs and employing anti-forensic techniques to impede analysis. Performing regular backups of systems allows analysts to view the contents of the system as they were at a particular time. In addition, as described in Sections 6 and 7, security monitoring controls such as intrusion detection software, antivirus software, and spyware detection and removal utilities can generate logs that show when and how an attack or intrusion took place.

3.1.2 Acquiring the Data

After identifying potential data sources, the analyst needs to acquire the data from the sources. Data acquisition should be performed using a three-step process: developing a plan to acquire the data, acquiring the data, and verifying the integrity of the acquired data.

1. **Develop a plan to acquire the data.** The analyst should create a plan that prioritizes the sources, establishing the order in which the data should be acquired. Important factors for prioritization include the following:
   - **Likely Value.**
   - **Volatility.** In many cases, acquiring volatile data should be given priority over non-volatile data.
   - **Amount of Effort Required.**
2. **Acquire the data.** Data acquisition can be performed either locally or over a network. Although it is generally preferable to acquire data locally because there is greater control over the system and data, local data collection is not always feasible (e.g., system in locked room, system in another location). When acquiring data over a network, decisions should be made regarding the type of data to be collected and the amount of effort to use. For instance, it might be necessary to acquire data from several systems through different network connections, or it might be sufficient to copy a logical volume from just one system.
3. **Verify the integrity of the data.** Data integrity verification typically consists of using tools to compute the message digest of the original and copied data, then comparing the digests to make sure that they are the same.

Before the analyst begins to collect any data, a decision should be made by the analyst or management (in accordance with the organization's policies and legal advisors) on the need to collect and preserve evidence in a way that supports its use in future legal or internal disciplinary proceedings. In such situations, a clearly defined chain of custody should be followed to avoid allegations of mishandling or tampering of evidence. This involves keeping a log of every person who had physical custody of the evidence, documenting the actions that they performed

on the evidence and at what time, storing the evidence in a secure location when it is not being used, making a copy of the evidence and performing examination and analysis using only the copied evidence, and verifying the integrity of the original and copied evidence. If it is unclear whether or not evidence needs to be preserved, by default it generally should be preserved.

In addition, several other steps should be taken. Throughout the process, a detailed log should be kept of every step that was taken to collect the data, including information about each tool used in the process. The documentation allows other analysts to repeat the process later if needed.

To assist the analyst with evidence collection, the necessary resources, such as forensic workstations, backup devices, blank media, and evidence handling supplies (e.g., hard-bound notebooks, chain of custody forms, evidence storage bags and tags, evidence tape, digital cameras) should be prepared beforehand.

There also may be situations where a law enforcement representative should handle the data collection for legal reasons. This includes, but is not limited to, obtaining ISP records and collecting data from external computer systems and unusual devices and media. Based on guidance from legal advisors, organizations should determine in advance what types of data are best collected by law enforcement officials.

Analysts should take into account what will be done with the collected data and plan for the potential ramifications. In some cases, the data may be turned over to a law enforcement agency or another external party for examination and analysis. This could result in the collected hardware being unavailable for an extended period of time. If the original media needs to be kept secured for legal proceedings, it could be unavailable for years. Another concern is that sensitive information unrelated to the investigation (e.g., medical records, financial information) might be inadvertently captured along with the desired data.

3.1.3 Incident Response Considerations

When performing forensics during incident response, an important consideration is how and when the incident should be contained. This decision should be based on existing policies and procedures regarding incident containment, as well as the team's assessment of the risk posed by the incident, so that the chosen containment strategy or combination of strategies sufficiently mitigates risk while maintaining the integrity of potential evidence whenever possible.

The organization should also consider in advance the impact that various containment strategies may have on the ability of the organization to operate effectively. Significant downtime could result in substantial monetary losses to the organization. Therefore, care should be taken to minimize disruptions to an organization's operations.

3.2 Examination

After data has been collected, the next phase is to examine the data, which involves assessing and extracting the relevant pieces of information from the collected data. An acquired hard drive may contain hundreds of thousands of data files; identifying the data files that contain information of interest, including information concealed through file compression and access control, can be a daunting task.

Fortunately, various tools and techniques can be used to reduce the amount of data that has to be sifted through. Text and pattern searches can be used to identify pertinent data, such as finding documents that mention a particular subject or person, or identifying e-mail log entries for a particular e-mail address. Another helpful technique is to use a tool that can determine the type of contents of each data file, such as text, graphics, music, or a compressed file archive. Knowledge of data file types can be used to identify files that merit further study, as well as to exclude files that are of no interest to the examination. There are also databases containing information about known files, which can also be used to include or exclude files from further consideration. Specific information about examination tools and techniques is presented in Sections 4.3, 5.3, 6.4, and 7.4.

3.3 Analysis

Once the relevant information has been extracted, the analyst should study and analyze the data to draw conclusions from it.

3.4 Reporting

The final phase is reporting, which is the process of preparing and presenting the information resulting from the analysis phase. Many factors affect reporting, including the following:

- **Alternative Explanations.** Analysts should use a methodical approach to attempt to prove or disprove each possible explanation that is proposed.
- **Audience Consideration.** Knowing the audience to which the data or information will be shown is important. An incident requiring law enforcement involvement requires highly detailed reports of all information gathered, and may also require copies of all evidentiary data obtained.
- **Actionable Information.** Reporting also includes identifying actionable information gained from data that may allow an analyst to collect new sources of information.

As part of the reporting process, analysts should identify any problems that may need to be remedied, such as policy shortcomings or procedural errors. Many forensic and incident response teams hold formal reviews after each major event.

3.5 Recommendations

The key recommendations presented in this section for the forensic process are as follows:

- **Organizations should perform forensics using a consistent process.** This guide presents a four-phase forensic process, with collection, examination, analysis, and reporting phases. The exact details of each phase may vary based on the need for forensics.
- **Analysts should be aware of the range of possible data sources.** Analysts should be able to survey a physical area and recognize possible sources of data. Analysts should also think of possible data sources located elsewhere within an organization and outside the organization.
- **Organizations should be proactive in collecting useful data.** Configuring auditing on OSs, implementing centralized logging, performing regular system backups, and using security monitoring controls can all generate sources of data for future forensic efforts.
- **Analysts should perform data collection using a standard process.**
- **Analysts should use a methodical approach to studying the data.**
- **Analysts should review their processes and practices.**

4. Using Data from Data Files

A data file (also called a file) is a collection of information logically grouped into a single entity and referenced by a unique name, such as a filename. A file can be of many data types, including a document, an image, a video, or an application. Successful forensic processing of computer media depends on the ability to collect, examine, and analyze the files that reside on the media.

4.1 File Basics

Before attempting to collect or examine files, analysts should have a reasonably comprehensive understanding of files and filesystems. First, analysts should be aware of the variety of media that may contain files.

4.1.1 File Storage Media

The widespread use of computers and other digital devices has resulted in a significant increase in the number of different media types that are used to store files.

Table 4-1. Commonly Used Media Types

Media Type	Reader	Typical Capacity	Comments
Primarily Used in Personal Computers			
Floppy Disk	Floppy disk drive	1.44 megabytes (MB)	3.5-inch disks; decreasing in popularity
CD-ROM	CD-ROM drive	650 MB-800 MB	Includes write-once (CD-R) and rewritable (CD-RW) disks; most commonly used media
DVD-ROM	DVD-ROM drive	1.67 gigabytes (GB)-15.9 GB	Includes write-once (DVD±R) and rewritable (DVD±RW) single and dual layer disks
Hard Drive	N/A	20 GB-400 GB	Higher capacity drives used in many file servers
Zip Disk	Zip drive	100 MB-750 MB	Larger than a floppy disk
Jaz Disk	Jaz drive	1 GB-2 GB	Similar to Zip disks; no longer manufactured
Backup tape	Compatible tape drive	80 MB-320 GB	Many resemble audio cassette tapes; fairly susceptible to corruption from environmental conditions
Magneto optical (MO) disk	Compatible MO drive	600 MB-9.1 GB	5.25-inch disks; less susceptible to environmental conditions than backup tapes
Advanced Technology Attachment (ATA) flash card	PCMCIA slot	8 MB-2 GB	PCMCIA flash memory card; measures 85.6 x 54 x 5 mm

Media Type	Reader	Typical Capacity	Comments
Used by many types of Digital Devices			
Flash/Jump drive	USB interface	16 MB-2 GB	Also known as thumb drives because of their size
Compact Flash card	PCMCIA adapter or memory card reader	16 MB-6 GB	Type I cards measure 43 x 36 x 3.3 mm; Type II cards measure 43 x 36 x 5 mm
Microdrive	PCMCIA adapter or memory card reader	340 MB-4 GB	Same interface and form factor as CompactFlash Type II cards
Multimedia Card (MMC)	PCMCIA adapter or memory card reader	16 MB-512 MB	Measures 24 x 32 x 1.4 mm
Secure Digital (SD) Card	PCMCIA adapter or memory card reader	32 MB-1 GB	Compliant with Secure Digital Music Initiative (SDMI) requirements; provides built-in data encryption of file contents; similar in form factor to MMCs
Memory Stick	PCMCIA adapter or memory card reader	16 MB-2 GB	Includes Memory Stick (50 x 21.5 x 2.8 mm), Memory Stick Duo (31 x 20 x 1.6 mm), Memory Stick PRO, Memory Stick PRO Duo; some are compliant with SDMI requirements and provide built-in encryption of file contents
Smart Media Card	PCMCIA adapter or memory card reader	8 MB-128 MB	Measures 37 x 45 x 0.76 mm
xD-Picture Card	PCMCIA adapter or xD-Picture card reader	16 MB-512 MB	Currently used only in Fujifilm and Olympus digital cameras; measures 20 x 25 x 1.7 mm

4.1.2 Filesystems

Before media can be used to store files, the media must usually be partitioned and formatted into logical volumes. Partitioning is the act of logically dividing a media into portions that function as physically separate units. A logical volume is a partition or a collection of partitions acting as a single entity that has been formatted with a filesystem.

A filesystem defines the way that files are named, stored, organized, and accessed on logical volumes. Many different filesystems exist, each providing unique features and data structures.

Some commonly used filesystems are as follows:

- **FAT12.17**
- **FAT16.**
- **FAT32.18**
- **NTFS.**
- **High-Performance File System (HPFS).**
- **Second Extended Filesystem (ext2fs).20**
- **Third Extended Filesystem (ext3fs).**
- **ReiserFS.21**
- **Hierarchical File System (HFS).22**
- **HFS Plus.23 .**
- **UNIX File System (UFS).24**
- **Compact Disk File System (CDFS).**
- **International Organization for Standardization (ISO) 9660 and Joliet.**
- **Universal Disk Format (UDF).**

4.1.3 Other Data on Media

- **Deleted Files.** When a file is deleted, it is typically not erased from the media; instead, the information in the directory's data structure that points to the location of the file is marked as deleted. This means that the file is still stored on the media but is no longer enumerated by the OS.
- **Slack Space.** As noted previously, filesystem's use file allocation units to store files. Even if a file requires less space than the file allocation unit size, an entire file allocation unit is still reserved for the file. For example, if the file allocation unit size is 32 kilobytes (KB) and a file is only 7 KB, the entire 32 KB is still allocated to the file, but only 7 KB is used, resulting in 25 KB of unused space. This unused space is referred to as file slack space, and it may hold residual data such as portions of deleted files.
- **Free Space.** Free space is the area on media that is not allocated to any partition; it includes unallocated clusters or blocks.

4.2 Collecting Files

During data collection, the analyst should make multiple copies of the relevant files or filesystems — typically a master copy and a working copy.

4.2.1 Copying Files from Media

Files can be copied from media using two different techniques:

- **Logical Backup.** A logical backup copies the directories and files of a logical volume. It does not capture other data that may be present on the media, such as deleted files or residual data stored in slack space.

- **Bit Stream Imaging.** Also known as disk imaging, bit stream imaging generates a bit-for-bit copy of the original media, including free space and slack space. Bit stream images require more storage space and take longer to perform than logical backups.

If evidence may be needed for prosecution or disciplinary actions, the analyst should get a bit stream image of the original media, label the original media, and store it securely as evidence. All subsequent analysis should be performed using the copied media to ensure that the original media is not modified and that a copy of the original media can always be recreated if necessary. All steps that were taken to create the image copy should be documented. Doing so should allow any analyst to produce an exact duplicate of the original media using the same procedures. In addition, proper documentation can be used to demonstrate that evidence was not mishandled during the collection process. Besides the steps that were taken to record the image, the analyst should document supplementary information such as the hard drive model and serial number, media storage capacity, and information about the imaging software or hardware that was used (e.g., name, version number, licensing information). All of these actions support the maintenance of the chain of custody.

Hardware tools can acquire data from drives that use common types of controllers, such as Integrated Drive Electronics (IDE) and Small Computer System Interface (SCSI). Software solutions generally consist of a startup diskette, CD, or installed programs that run on a workstation to which the media to be imaged is attached. Some software solutions create logical copiesof files or partitions and may ignore free or unallocated drive space, wheras others create a bit-by-bit image copy of the media.

In addition to their primary function, some disk imaging tools can also perform forensic recordkeeping, such as automated audit trails and chain of custody. The use of such tools can support consistency in the examination process and the accuracy and reproducibility of results. An increasing number of disk imaging tools are becoming available. In response to this proliferation and the lack of a standard for testing them, NIST's Computer Forensics Tool Testing (CFTT) project has developed rigorous testing procedures for validating the tools. results. Currently, only a few disk imaging tools have undergone CFTT testing.

Generally, tools that perform bit stream imaging should not be used to acquire bit-by-bit copies of an entire physical device from a live system — a system currently in use — because the files and memory on such a system are changing constantly and therefore cannot be validated.32 However, a bit-by-bit copy of the logical areas of a live system can be completed and validated. When logical backups are being performed, it is still preferable not to copy files from a live system; changes might be made to files during the backup, and files that are held open by a process might not be easy to copy. Accordingly, analysts should decide whether copying files from a live system is feasible based on which files need to be obtained, how accurate and complete the copying needs to be, and how important the live system is.

4.2.2 Data File Integrity

During backups and imaging, the integrity of the original media should be maintained.

After a backup or imaging is performed, it is important to verify that the copied data is an exact duplicate of the original data. Computing the message digest of the copied data can be used to verify and ensure data integrity.

4.2.3 File Modification, Access, and Creation Times

It is often important to know when a file was created, used, or manipulated, and most OSs keep track of certain timestamps related to files. The most commonly used timestamps are the modification, access, and creation (MAC) times, as follows:

- **Modification Time.**
- **Access Time.**
- **Creation Time.**

If an analyst needs to establish an accurate timeline of events, then the file times should be preserved. Accordingly, analysts should be aware that not all methods for collecting data files can preserve file times. Bit stream images can preserve file times because a bit-for-bit copy is generated; performing a logical backup using some tools may cause file creation times to be altered when the data file is copied.

Analysts should also be aware that file times may not always be accurate. Among the reasons for such inaccuracies are the following:

- The computer's clock does not have the correct time.
- The time may not be recorded with the expected level of detail, such omitting the seconds or minutes.
- An attacker may have altered the recorded file times.

4.2.4 Technical Issues

Several technical issues may arise in collecting data files. As noted in Section 4.2.1, the primary issue is the collection of deleted files and remnants of files existing in free and slack space on media.

Another common issue is the collection of hidden data. Some applications and OSs hide configuration files to reduce the chance that users will accidentally modify or delete them. Also, on some OSs, directories that have been deleted may be marked as hidden. Hidden data may contain a wealth of information; for example, a hidden partition could contain a separate OS and many data files. Many collection tools can recognize some or all of these methods of hiding data and recover the associated data.

Yet another issue that may arise is collection of data from RAID arrays that use striping (e.g., RAID-0, RAID-5). In this configuration, a striped volume consists of equal-sized partitions that reside on separate disk drives. When data is written to the

volume, it is evenly distributed across the partitions to improve disk performance. Some imaging tools can acquire striped volumes and preserve unused data areas of the volume, such as free space and slack space.

4.3 Examining Data Files

After a logical backup or bit stream imaging has been performed, the backup or image may have to be restored to another media before the data can be examined. This is dependent on the forensic tools that will be used to perform the analysis. Some tools can analyze data directly from an image file, whereas others require that the backup or image be restored to a medium first. Regardless of whether an image file or a restored image is used in the examination, the data should be accessed only as read-only to ensure that the data being examined is not modified and that it will provide consistent results on successive runs.

4.3.1 Locating the Files

Manually extracting data from unused space can be a time-consuming and difficult process, because it requires knowledge of the underlying filesystem format. Fortunately, several tools are available that can automate the process of extracting data from unused space and saving it to data files, as well as recovering deleted files and files within a recycling bin.

4.3.2 Extracting the Data

The rest of the examination process involves extracting data from some or all of the files. To make sense of the contents of a file, an analyst needs to know what type of data the file contains. Analysts should not assume that file extensions are accurate.

Analysts can more accurately identify the type of data stored in many files by looking at their file headers. A file header contains identifying information about a file and possibly metadata that provides information about the file's contents.

4.3.3 Using a Forensic Toolkit

Many forensic products allow the analyst to perform a wide range of processes to analyze files and applications, as well as collecting files, reading disk images, and extracting data from files. Most analysis products also offer the ability to generate reports and to log all errors that occurred during the analysis. The following processes are among those that an analyst should be able to perform with a variety of tools:

- **Using File Viewers.** Using viewers instead of the original source applications to display the contents of certain types of files is an important technique for scanning or previewing data, and is more efficient because the analyst does not need native applications for viewing each type of file.
- **Uncompressing Files.** Compressed files may contain files with useful information, as well as other compressed files. Therefore, it is important that the analyst locate and extract compressed files. Uncompressing files should be performed early in the forensic process to ensure that the contents of compressed files are included in searches and other actions.

- **Graphically Displaying Directory Structures.** This practice makes it easier and faster for analysts to gather general information about the contents of media, such as the type of software installed and the likely technical aptitude of the user(s) who created the data.
- **Identifying Known Files.** The benefit of finding files of interest is obvious, but it is also often beneficial to eliminate unimportant files, such as known good OS and application files, from consideration. Analysts should use validated hash sets, such as those created by NIST's National Software Reference Library (NSRL) project or personally created hash sets that have been validated, as a basis for identifying known benign and malicious files.
- **Performing String Searches and Pattern Matches.** String searches aid in perusing large amounts of data to find key words or strings. Various searching tools are available that can use Boolean, fuzzy logic, synonyms and concepts, stemming, and other search methods. Examples of common searches include searching for multiple words in a single file and searching for misspelled versions of certain words. Developing concise sets of search terms for common situations can help the analyst reduce the volume of information to review. Some considerations or possible difficulties in performing string searches are as follows:
  - Some proprietary file formats cannot be string searched without additional tools. In addition, compressed, encrypted, and password-protected files require additional pre-processing before a string search.
  - The use of multi-character data sets that include foreign or Unicode characters can cause problems with string searches; some searching tools attempt to overcome this by providing language translation functions.
  - Another possible issue is the inherent limitations of the search tool or algorithm. For example, a match might not be found for a search string if part of the string resided in one cluster and the rest of the string resided in a nonadjacent cluster.
- **Accessing File Metadata.** File metadata provides details about any given file.

4.4 Analysis

After the examination has been completed, the next step is to perform analysis of the extracted data. As mentioned in Section 4.3.3, there are many tools available that can be helpful in analysis of different types of data. When using these tools or performing manual reviews of data, analysts should be aware of the value of using system times and file times. Knowing when an incident occurred, a file was created or modified, or an e-mail was sent can be critical to forensic analysis. For example, such information can be used to reconstruct a timeline of activities.

If multiple tools are used to complete an examination and analysis, the analyst should understand how each tool extracts, modifies, and displays file modification, access, and creation (MAC) times. For instance, some tools modify the last access time of a

file or directory if the filesystem has been mounted with write permissions by the OS. Write-blockers can be used to prevent these tools from modifying the MAC times; however, although write-blockers can prevent these times from being modified on the media, they cannot prevent the OS from caching the changes in memory (i.e., storing the changes in random access memory [RAM]). The OS might then report the cached MAC times rather than the actual times, thereby returning inaccurate results. The analyst should be aware that the last access time for data files and directories might change between queries, depending on the tool used to perform the query. Because of these issues, analysts should take care in choosing a MAC viewing method and record the details of that method.

4.5 Recommendations

The key recommendations presented in this section for using data from data files are as follows.

- **Analysts should examine copies of files, not the original files..**
- **Analysts should preserve and verify file integrity.** The integrity of copied data should be verified by computing and comparing the message digests of files.
- **Analysts should rely on file headers, not file extensions, to identify file content types.**
- **Analysts should have a forensic toolkit for data examination and analysis.** The toolkit should contain various tools that provide the ability to perform quick reviews of data as well as in-depth analysis.

5. Using Data from Operating Systems

5.1 OS Basics

OS data exists in both non-volatile and volatile states. Non-volatile data refers to data that persists even after a computer is powered down, such as a filesystem stored on a hard drive. Volatile data refers to data on a live system that is lost after a computer is powered down, such as the current network connections to and from the system. Many types of non-volatile and volatile data may be of interest from a forensics perspective.

5.1.1 Non-Volatile Data

The primary source of non-volatile data within an OS is the filesystem. The filesystem is also usually the largest and richest source of data within the OS, containing most of the information recovered during a typical forensic event. The filesystem provides storage for the OS on one or more media. A filesystem typically contains many types of files, each of which may be of value to analysts in different situations. In addition, as noted in Section 4.1.2, important residual data can be recovered from unused filesystem space. Several types of data that are commonly found within OS filesystems are as follows:

- **Configuration Files.** The OS may use configuration files to store OS and application settings.
- **Logs.** OS log files contain information about various OS events, and may also hold application-specific event information. The types of information typically found in OS logs are as follows:
- **System Events.** System events are operational actions performed by OS components, such as shutting down the system or starting a service.
  - **Audit Records**. Audit records contain security event information such as successful and failed authentication attempts and security policy changes.
  - **Application Events.** Application events are significant operational actions performed by applications, such as application startup and shutdown, application failures, and major application configuration changes.
  - **Command History.** Some OSs have separate log files (typically for each user) that contain a history of the OS commands performed by each user.
  - **Recently Accessed Files.** An OS might log the most recent file accesses or other usage, creating a list of the most recently accessed files.
- **Application Files**. Applications can be composed of many types of files, including executables, scripts, documentation, configuration files, log files, history files, graphics, sounds, and icons.
- **Data Files.** Data files store information for applications. Examples of common data files are text files, word processing documents, spreadsheets, databases, audio files, and graphics files.

- **Swap Files**. Most OSs use swap files in conjunction with RAM to provide temporary storage for data often used by applications. Swap files may contain a broad range of OS and application information, such as usernames, password hashes, and contact information.
- **Dump Files**. Some OSs have the ability to store the contents of memory automatically during an error condition to assist in subsequent troubleshooting. The file that holds the stored memory contents is known as a dump file.
- **Hibernation Files.** A hibernation file is created to preserve the current state of a system (typically a laptop) by recording memory and open files before shutting off the system.
- **Temporary Files.** During the installation of an OS, application, or OS or application updates and upgrades, temporary files are often created. Although such files are typically deleted at the end of the installation process, this does not always occur. In addition, temporary files are created when many applications are run; again, such files are usually deleted when the application is terminated, but this does not always happen.

5.1.2 Volatile Data

OSs execute within the RAM of a system. While the OS is functioning, the contents of RAM are constantly changing. At any given time, RAM might contain many types of data and information that could be of interest. In addition, like filesystems, RAM can contain residual data in slack and free space, as follows:

- **Slack Space.** Memory slack space is much less deterministic than file slack space. For example, an OS generally manages memory in units known as pages or blocks, and allocates them to requesting applications. Sometimes, although an application might not request an entire unit, it is given one anyway. Residual data could therefore reside in the unit of memory allocated to an application, although it might not be addressable by the application.
- **Free Space**. Memory pages are allocated and deallocated much like file clusters. When they are not allocated, memory pages are often collected into a common pool of available pages — a process often referred to as garbage collection. It is not uncommon for residual data to reside in these reusable memory pages, which are analogous to unallocated file clusters.

Some other significant types of volatile data that might exist within an OS are as follows:

- **Network Configuration.** Although many elements of networking, such as network interface card (NIC) drivers and configuration settings, are typically stored in the filesystem, networking is dynamic in nature. Users also may be able to alter network interface configurations from the defaults, such as manually changing IP addresses. Accordingly, analysts should use the current network configuration, not the stored configuration, whenever possible.

- **Network Connections**. The OS facilitates connections between the system and other systems. Most OSs can provide a list of current incoming and outgoing network connections, and some OSs can list recent connections as well. For incoming connections, the OS typically indicates which resources are being used, such as file shares and printers. Most OSs can also provide a list of the ports and IP addresses at which the system is listening for connections.
- **Running Processes.** Processes are the programs that are currently executing on a computer. Processes include services offered by the OS and applications run by administrators and users. Identifying the running processes is also helpful for identifying programs that should be running but have been disabled or removed, such as antivirus software and firewalls.
- **Open Files.** OSs may maintain a list of open files, which typically includes the user or process that opened each file.
- **Login Sessions**. OSs typically maintain information about currently logged-in users (and the start time and duration of each session), previous successful and failed logons, privileged usage, and impersonation.65 Logon records can help to determine a user.s computer usage habits and confirm whether a user account was active when a given event occurred.
- **Operating System Time.** The OS maintains the current time and stores daylight savings time and time zone information. This information can be useful when building a timeline of events or correlating events among different systems. Analysts should be aware that the time presented by the OS might differ from that presented by the BIOS because of OS-specific settings, such as time zone.

5.2.1 Collecting Volatile OS Data

Volatile OS data involving an event can be collected only from a live system that has not been rebooted or shut down since the event occurred. Every action performed on the system, whether initiated by a person or by the OS itself, will almost certainly alter the volatile OS data in some way. Therefore, analysts should decide as quickly as possible whether the volatile OS data should be preserved. The importance of this decision cannot be stressed enough, because powering off the system or even disconnecting it from a network can eliminate the opportunity to collect potentially important information.

On the other hand, collecting volatile OS data from a running computer has inherent risks. For instance, the possibility always exists that files on the computer might change and other volatile OS data might be altered. In addition, a malicious party might have installed rootkits designed to return false information, delete files, or perform other malicious acts.

5.2.1.1 Forensic Tool Preparation

When collecting volatile OS data, all forensic tools that might be needed should be placed on a floppy disk, CD-ROM, or USB flash drive, from which the tools should be executed. Doing so enables analysts to collect OS data with the least amount of

disturbance to the system. In addition, only forensic tools should be used, since a user might have replaced system commands with malicious programs, such as one to format a hard disk or return false information. However, use of forensic tools is no guarantee that the data retrieved will be accurate. If a system has been fully compromised, it is possible that rootkits and other malicious utilities have been installed that alter the system's functionality at the kernel level. This can cause false data to be returned to user-level tools.

When creating a collection of forensic tools, statically linked binary files should be used. Such an executable file contains all of the functions and library functions that it references, so separate dynamic link libraries (DLL) and other supporting files are not needed. This eliminates the need to place the appropriate versions of DLLs on the tool media and increases the reliability of the tools. The analyst should know how each tool affects or alters the system before collecting the volatile data. The message digest of each tool should be computed and stored safely to verify file integrity. Licensing and version information also should be documented for each forensic tool. In addition, the exact commands that were used to run each forensic tool should be documented (i.e., command line arguments and switches). It may be helpful to place a script on the tool media that can be run to capture which commands were run, at what time, and with what output.

5.2.1.2 Types of Volatile OS Data

The following list shows several types of volatile OS data and explains how forensic tools can be used in collecting each type of data:

- **Contents of Memory.**
- **Network Configuration.**
- **Network Connections.**
- **Running Processes.**
- **Open Files.**
- **Login Sessions.**
- **Operating System Time.**

In addition to the tools in the preceding list, it is often useful to include some general-purpose tools in the forensic toolkit, such as the following:

- **OS Command Prompt.**
- **SHA-1 Checksum.**
- **Directory List.**
- **String Search.**
- **Text Editor.**

5.2.1.3 Prioritizing Data Collection

The types of volatile data that should be collected with the toolkit depend on the specific need. When in doubt, it is usually a good idea to collect as much volatile

data as possible because all opportunities to collect such data will be lost once the computer is powered down. Later, a determination can be made as to which collected volatile data should be examined.

Because volatile data has a propensity to change over time, the order and timeliness with which volatile data is collected is important. In most cases, analysts should first collect information on network connections and login sessions, because network connections may time out or be disconnected and the list of users connected to a system at any single time may vary. Volatile data that is less likely to change, such as network configuration information, should be collected later. The recommended order in which volatile data generally should be collected, from first to last, is as follows:

1. Network connections
2. Login sessions
3. Contents of memory
4. Running processes
5. Open files
6. Network configuration
7. Operating system time.

5.2.2 Collecting Non-Volatile OS Data

Some tools are able to perform collection actions on running systems without any problems, while other tools are best run on systems that have been shut down.

Once the filesystem data has been collected, tools can be used to acquire specific types of data from the filesystem. The following list describes several other types of non-volatile OS data and explains how tools can be useful in acquiring each type from the filesystem:

- **Users and Groups**. Operating systems maintain a list of users and groups that have access to the system.
- **Passwords.** Most OSs maintain password hashes for users. passwords on disk.
- **Network Shares.** A system may enable local resources to be shared across a network.
- **Logs.** Logs that are not stored in text files might necessitate use of log extraction utilities.

5.2.3 Technical Issues with Collecting Data

Technical issues might also impede collection of OS data. Section 4 describes several filesystem-related issues; this section focuses on additional collection issues and provides guidance on what, if anything, can be done to mitigate them. The intent of this section is not to provide an exhaustive discussion of all possible issues, but to provide some basic information on common ones.

- **OS Access.** Collecting volatile data can be difficult because the analyst might

not be able to readily gain access to the OS. For instance, a user might run a password-protected screen saver or have the system locked. In these cases, the analyst will need to circumvent this protection or find another way to gain access to volatile OS data.

- **Log Modification.** The user might try to reduce the usefulness of logs by disabling log features, modifying log settings so that there is little storage available for logs, or writing many spurious events to the logs.
- **Hard Drives with Flash Memory.** Occasionally, an analyst might come across a hard drive that also contains flash memory. This flash memory could contain a password that is needed to access the drive, even when the drive has been removed from the computer. Typically, the analyst must then find, guess, or crack the password to gain access to the drive.
- **Key Remapping.** The best way to avoid key remapping problems is by collecting data from the computer without using its keyboard. For example, the analyst could attach a forensic workstation to a computer of interest using a crossover network cable and run scripts from the forensic workstation.

5.3 Examining and Analyzing OS Data

Various tools and techniques can be used to support the examination process. Many of the tools and techniques discussed in Section 4.3 for examining collected data files can also be used with collected OS data. In addition, as described in Section 7, security applications, such as file integrity checkers and host IDSs, can be very helpful in identifying malicious activity against OSs. If intrusion detection software is installed on the computer, it might contain logs that indicate the actions performed against the OS.

Analysts often want to gather additional information about a particular program running on a system, such as the process's purpose and manufacturer. After obtaining a list of the processes currently running on a system, analysts can look up the process name to obtain such additional information. However, users might change the names of programs to conceal their functions, such as naming a Trojan program calculator.exe. Therefore, process name lookups should be performed only after verifying the identity of the process's files by computing and comparing their message digests. Similar lookups can be performed on library files, such as DLLs on Windows systems, to determine which libraries are loaded and what their typical purposes are.

5.4 Recommendations

The key recommendations presented in this section for using data from OSs are as follows.

- **Analysts should act appropriately to preserve volatile OS data.**
- **Analysts should use a forensic toolkit for collecting volatile OS data.**
- **Analysts should choose the appropriate shutdown method for each system.** Each method of shutting down a particular OS can cause different types of data to be preserved or corrupted; analysts should be aware of the typical shutdown behavior of each OS.

6. Using Data from Network Traffic

Analysts can use data from network traffic to reconstruct and analyze network-based attacks and inappropriate network usage, as well as to troubleshoot various types of operational problems. The term network traffic refers to computer network communications that are carried over wired or wireless networks between hosts.82 This section provides an introduction to network traffic, including descriptions of major sources of network traffic data (e.g., intrusion detection software, firewalls).

6.1 TCP/IP Basics

TCP/IP communications are composed of four layers that work together.

Figure 6-1. TCP/IP Layers

Application Layer. This layer sends and receives data for particular applications, such as Domain Name System (DNS), Hypertext Transfer Protocol (HTTP), and Simple Mail Transfer Protocol (SMTP).
Transport Layer. This layer provides connection-oriented or connectionless services for transporting application layer services between networks. The transport layer can optionally ensure the reliability of communications. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are commonly used transport layer protocols.
Internet Protocol Layer (also known as Network Layer). This layer routes packets across networks. IP is the fundamental network layer protocol for TCP/IP. Other commonly used protocols at the network layer are Internet Control Message Protocol (ICMP) and Internet Group Management Protocol (IGMP).
Hardware Layer (also known as Data Link Layer). This layer handles communications on the physical network components. The best known data link layer protocol is Ethernet

6.1.5 Layers. Significance in Network Forensics

Network forensic analysis relies on all of the layers. When analysts begin to examine data, they typically have limited information — most likely an IP address of interest and perhaps protocol and port information. Nevertheless, this is enough information to support searching common data sources for more information. In most cases, the application layer contains the actual activity of interest — most attacks are against vulnerabilities in applications (including services), and nearly all misuse involves misuse of applications. Analysts need IP addresses so that they can identify the hosts that may have been involved in the activity. The hosts may also contain additional data that would be of use in analyzing the activity. Although some events of interest may not have relevant application-level data (e.g., a distributed denial of service attack designed to consume all network bandwidth), most do; network forensics provides important support to the analysis of application-layer activities.

6.2 Network Traffic Data Sources

Organizations typically have several types of information sources concerning network traffic that might be useful for network forensics. These sources collectively capture important data from all four TCP/IP layers. The following subsections highlight the major categories of network traffic data sources — firewalls and routers, packet sniffers and protocol analyzers, IDSs, remote access, security event management software, and network forensic analysis tools — as well as several other types of data sources.

6.2.1 Firewalls and Routers

Firewalls and routers are usually configured to log basic information for most or all denied connection attempts and connectionless packets; some log every packet.91 Information logged typically includes the date and time the packet was processed, the source and destination IP addresses, the transport layer protocol (e.g., TCP, UDP, ICMP), and basic protocol information (e.g., TCP or UDP port numbers, ICMP type and code). The content of packets is usually not recorded.

6.2.2 Packet Sniffers and Protocol Analyzers

Packet sniffers are designed to monitor network traffic on wired or wireless networks and capture packets. For example, if IDS alerts indicate unusual network activity between two hosts, a packet sniffer could record all of the packets traveling between the hosts, potentially providing additional information for analysts.

6.2.3 Intrusion Detection Systems

Network IDSs perform packet sniffing and analyze network traffic to identify suspicious activity and record relevant information. Host IDSs monitor characteristics of a particular system and events occurring within the system, which can include network traffic. Unlike network IDS sensors, which can monitor all network traffic on a particular network segment, host IDS software is intended to monitor network traffic only for the host on which it is installed. For each suspicious event, IDS software typically records the same basic event characteristics that firewalls and routers record (e.g., date and time, source and destination IP addresses, protocol, basic protocol characteristics), as well as application-specific information (e.g., username, filename, command, status code). IDS software also records information that indicates the possible intent of the activity. Examples include the type of attack (e.g., buffer overflow), the targeted vulnerability, the apparent success or failure of the attack, and pointers to more information about the attack.

6.2.4 Remote Access

Remote access servers are devices such as VPN gateways and modem servers that facilitate connections between networks. Remote access servers typically record the origin of each connection and might also indicate which user account was authenticated for each session.

6.2.5 Security Event Management Software

Security event management (SEM) software is capable of importing security event information from various network traffic-related security event data sources (e.g., IDS logs, firewall logs) and correlating events among the sources. It generally works by receiving copies of logs from various data sources over secure channels, normalizing the logs into a standard format, then identifying related events by matching IP addresses, timestamps, and other characteristics. SEM products usually do not generate original event data; instead, they generate meta-events based on imported event data.

Many SEM products not only can identify malicious activity, such as attacks and virus infections, but also can detect misuse and inappropriate usage of systems and networks. SEM software can be helpful in making many sources of network traffic information accessible through a single interface.

6.2.6 Network Forensic Analysis Tools

Network forensic analysis tools (NFAT) typically provide the same functionality as packet sniffers, protocol analyzers, and SEM software in a single product. Whereas SEM software concentrates on correlating events among existing data sources (which typically include multiple network traffic-related sources), NFAT software focuses primarily on collecting, examining, and analyzing network traffic. NFAT software also offers additional features that further facilitate network forensics, such as the following:

- Reconstructing events by replaying network traffic within the tool,
- Visualizing the traffic flows and the relationships among hosts.
- Building profiles of typical activity and identifying significant deviations.
- Searching application content for keywords (e.g., .confidential., .proprietary.).

6.2.7 Other Sources

Most organizations have other sources of network traffic information that can be of use for forensics in some capacity, including the following:

- **Dynamic Host Configuration Protocol Servers.**
- **Network Monitoring Software.**
- **Internet Service Provider Records.**
- **Client/Server Applications.**
- **Hosts. Network Configurations and Connections.**

6.3 Collecting Network Traffic Data

Although collecting network traffic data is typically straightforward, there are several important legal and technical issues that can make data collection more complicated.

6.3.1 Legal Considerations

Collecting network traffic can pose legal issues. Among these issues is the capture (intentional or incidental) of information with privacy or security implications, such as passwords or the contents of e-mails. Organizations should have policies in place regarding the handling of inadvertent disclosures of sensitive information. Another problem with capturing data such as e-mails and text documents is that long-term storage of such information might violate an organization's data retention policy. It is also important to have policies regarding the monitoring of networks and to have warning banners on systems that indicate that activity may be monitored.

Although most network traffic data collection occurs as part of regular operations, it can also occur as part of troubleshooting or incident handling. In the latter case, it is important to follow consistent processes and to document all actions performed. Organizations should have policies that clearly explain what types of monitoring can and cannot be performed without approval, and that describe or reference the procedures that detail the request and approval process.

Another potential legal issue is the preservation of original logs. As described in Section 6.4, many organizations send copies of network traffic logs to centralized devices, as well as use tools that interpret and analyze network traffic. In cases where logs may be needed as evidence, organizations may wish to collect copies of the original log files, the centralized log files, and interpreted log data, in case there are any questions regarding the fidelity of the copying and interpretation processes.

As privacy has become a greater concern to organizations, many have become less willing to share information with each other, including network forensic data. For example, most ISPs now require a court order before providing any information related to suspicious network activity that might have passed through their infrastructure. Although this preserves privacy and reduces the burden on and liability of the ISPs, it also slows down the investigative process. This is particularly challenging when an organization is attempting to trace an ongoing network-based attack to its source, especially if the traffic passes through several ISPs.

6.3.2 Technical Issues

Several technical issues might impede the collection of data about network traffic. This section describes several of the major issues and provides guidance on what, if anything, can be done to mitigate each.

- **Data Storage.** When there is a large volume of network activity, particularly during adverse events such as attacks, logs may record many events in a short time. If insufficient storage is available, information about recent activity may be overwritten and lost. Organizations should estimate typical and peak log usage, determine how many hours. or days. worth of data should be retained, and ensure that systems and applications have sufficient storage available to meet those goals.
- **Encrypted Traffic.** When protocols such as IP Security (IPsec), SSH, and Secure Sockets Layer (SSL) are used to encrypt network traffic, devices monitoring network traffic along the encrypted path can see only the most basic characteristics of the traffic, such as source and destination IP addresses. If VPNs or other tunneling techniques are being used, the IP addresses might be for the tunnel itself and not the true source and destination of the activity. Organizations should consider establishing policies that specify the appropriate use of traffic encryption technologies, so that security controls such as IDS sensors can monitor the contents of traffic that does not need to be or should not be encrypted.

- **Services Running on Unexpected Ports.** Applications such as IDSs and protocol analyzers often rely on port numbers to identify which service is in use for a given connection. Because traffic involving services running on unexpected port numbers may not be captured, monitored, or analyzed properly, use of unauthorized services (e.g., providing Web services on an atypical port) might not be detected. There are several ways to attempt to identify unexpected port usage, including the following:
  - Configuring IDS sensors to alert on connections involving unknown server ports
  - Configuring application proxies or IDS sensors that perform protocol analysis to alert on connections that use an unexpected protocol (e.g., FTP traffic using the standard HTTP port)
  - Performing traffic flow monitoring and identifying new and unusual traffic flows
  - Configuring a protocol analyzer to analyze a particular stream as something else.
- **Alternate Access Points.** Attackers often enter networks from alternate access points to avoid detection by security controls that are monitoring major access points, such as the organization's Internet gateway. Organizations typically address this potential problem by limiting alternate access points, such as modems and wireless access points, and ensuring that each is monitored and restricted through firewalls, IDS sensors, and other controls.
- **Monitoring Failures.** Inevitably, systems and applications will occasionally experience failures or outages for various reasons (e.g., system maintenance, software failures, attacks). In the case of dedicated monitoring systems such as IDS sensors, use of redundant equipment (e.g., two sensors monitoring the same activity) can lessen the impact of monitoring failures. Another strategy is to perform multiple levels of monitoring, such as configuring network-based and host-based firewalls to log connections.

6.4 Examining and Analyzing Network Traffic Data

When an event of interest has been identified, analysts assess, extract, and analyze network traffic data with the goal of determining what has happened and how the organization's systems and networks have been affected. This process might be as simple as reviewing a few log entries on a single data source and determining that the event was a false alarm, or as complex as sequentially examining and analyzing dozens of sources (most of which might contain no relevant data), manually correlating data among several sources, then analyzing the collective data to determine the probable intent and significance of the event. However, even the relatively simple case of validating a few log entries can be surprisingly involved and time-consuming.

Although current tools (e.g., SEM software, NFAT software) can be helpful in gathering and presenting network traffic data, such tools have rather limited analysis abilities and can be used effectively only by well-trained, experienced analysts. In addition

to understanding the tools, analysts should also have reasonably comprehensive knowledge of networking principles, common network and application protocols, network and application security products, and network-based threats and attack methods. It is also very important that analysts have knowledge of the organization's environment, such as the network architecture and the IP addresses used by critical assets (e.g., firewalls, publicly accessible servers), as well as knowledge of the information supporting the applications and OSs used by the organization. If analysts understand the organization's normal computing baseline, such as typical patterns of usage on systems and networks across the enterprise, they should be able to perform their work easier and faster. Analysts should also have a firm understanding of each of the network traffic data sources, as well as access to supporting materials, such as intrusion detection signature documentation. Analysts should understand the characteristics and relative value of each data source so that they can locate the relevant data quickly.

Given the potential complexities of the analysis process and the extensive knowledge of networking and information security required for analyzing network traffic data effectively, a full description of techniques needed for analyzing data and drawing conclusions in complex situations is beyond the scope of this document. Instead, the section focuses on the basic steps of the examination and analysis processes and highlights some significant technical issues that analysts should consider.

6.4.1 Identify an Event of Interest

The first step in the examination process is the identification of an event of interest. Typically, this identification is made through one of two methods:

- Someone within the organization (e.g., help desk agent, system administrator, security administrator) receives an indication, such as an automated alert or a user complaint, that there is a security or operational-related issue.
- During a review of security event data (e.g., IDS monitoring, network monitoring, firewall log review), which is part of the analyst's regular duties, the analyst identifies an event of interest and determines that it should be researched further.

When an event of interest has been identified, the analyst needs to know some basic information about the event as a basis for research.

6.4.2 Examine Data Sources

For initial event data examination, analysts typically rely on a few primary data sources, such as an IDS console that displays alerts from all IDS sensors, or SEM or NFAT software that consolidates many other data sources and organizes the data. Not only is this an efficient solution, but also in most cases the event of interest will be identified by an alert from one of these primary data sources.

For each data source examined, analysts should consider its fidelity. In general, analysts should have more confidence in original data sources than in data sources that receive normalized (modified) data from other sources. In addition, analysts

should validate data that is based on interpretation, such as IDS and SEM alerts. No tool for identifying malicious activity is completely accurate; they produce both false positives (incorrectly reporting benign activity as malicious) and false negatives (incorrectly classifying malicious activity as benign). Tools such as NFAT and IDS might also produce inaccurate alerts if they do not process all packets within a connection. Validation should be based on an analysis of additional data (e.g., raw packets, supporting information captured by other sources), a review of available information on alert validity (e.g., vendor comments on known false positives), and past experience with the tool in question. In many cases, an experienced analyst can quickly examine the supporting data and determine that an alert is a false positive and does not need further investigation.

Analysts may also need to examine secondary network traffic data sources, such as host-based firewall logs and packet captures, and non-network traffic data sources, such as host OS audit logs and antivirus software logs. The most common reasons for doing this are as follows:

- **No Data on Primary Sources.** In some cases, the typical primary network traffic data sources do not contain evidence of the activity.
- **Insufficient or Unvalidated Data on Primary Sources.** Analysts might need to examine secondary data sources if primary data sources do not contain sufficient information or analysts need to validate the data. After reviewing one or more primary data sources, analysts should query the appropriate secondary data sources based on the pertinent data from the primary data sources.
- **Best Source of Data Elsewhere.** Occasionally, the best sources of network traffic data are located on a particular host, such as host-based firewall and IDS logs on a system that was attacked. Although such data sources can be very helpful, their data may be altered or destroyed during a successful attack.

If additional data is needed but cannot be located and the suspicious activity is still occurring, analysts might need to perform more data collection activities. For example, an analyst could perform packet captures at an appropriate point on the network to gather more information. Other ways to collect more information include configuring firewalls or routers to log more information on certain activity, setting an IDS signature to capture packets for the activity, and writing a custom IDS signature that alerts when a specific activity occurs. Collecting additional data may be helpful if the activity is ongoing or intermittent; if the activity has ended, there is no opportunity to collect additional data.

6.4.2.1 Data Source Value

As described in Section 6.2, organizations typically have many different sources of network traffic data. Because the information collected by these sources varies, the sources may have different value to the analyst, both in general and for specific cases. The following items describe the typical value of the most common data sources in network forensics:

- **IDS Software.** IDS data is often the starting point for examining suspicious activity. Not only do IDSs typically attempt to identify malicious network traffic at all TCP/IP layers, but also they log many data fields (and sometimes raw packets) that can be useful in validating events and correlating them with other data sources. Nevertheless, as noted previously, IDS software does produce false positives, so IDS alerts should be validated.
- **SEM Software.** Ideally, SEM can be extremely useful for forensics because it can automatically correlate events among several data sources, then extract the relevant information and present it to the user. However, because SEM software functions by bringing in data from many other sources, the value of SEM depends on which data sources are fed into it, how reliable each data source is, and how well the software can normalize the data and correlate events.
- **NFAT Software.** NFAT software is designed specifically to aid in network traffic analysis, so it is valuable if it has monitored an event of interest.
- **Firewalls, Routers, Proxy Servers, and Remote Access Servers**. By itself, data from these sources is usually of little value. Analyzing the data over time can indicate overall trends, such as an increase in blocked connection attempts.
- **DHCP Servers.** DHCP servers typically can be configured to log each IP address assignment and the associated MAC address, along with a timestamp. This information can be helpful to analysts in identifying which host performed an activity using a particular IP address. However, analysts should be mindful of the possibility that attackers on an organization's internal networks falsified their MAC addresses or IP addresses, a practice known as spoofing.
- **Packet Sniffers.** Of all the network traffic data sources, packet sniffers can collect the most information on network activity. However, sniffers might capture huge volumes of benign data as well — millions or billions of packets — and typically provide no indication as to which packets might contain malicious activity. In most cases, packet sniffers are best used to provide more data on events that other devices or software has identified as possibly malicious.
- **Network Monitoring.** Network monitoring software is helpful in identifying significant deviations from normal traffic flows, such as those caused by DDoS attacks, during which, hundreds or thousands of systems launch simultaneous attacks against particular hosts or networks. Network monitoring software can document the impact of these attacks on network bandwidth and availability, as well as providing information about the apparent targets. Traffic flow data can also be helpful in investigating suspicious activity identified by other sources.
- **ISP Records.** Information from an ISP is primarily of value in tracing an attack back to its source, particularly when the attack uses spoofed IP addresses.

6.4.2.2 Examination and Analysis Tools

Because network forensics can be performed for many purposes with dozens of data source types, analysts may use several different tools on a regular basis, each well-

suited to certain situations. Analysts should be aware of the possible approaches to examining and analyzing network traffic data and should select the best tools for each case, rather than applying the same tool to every situation. Analysts should also be mindful of the shortcomings of tools. It can be helpful to have an alternate tool available that might not have the same deficiency.

6.4.3 Draw Conclusions

One of the most challenging aspects of network forensics is that the available data is typically not comprehensive. In many cases, if not most, some network traffic data has not been recorded and consequently has been lost. The analyst should eventually locate, validate, and analyze enough data to be able to reconstruct the event, understand its significance, and determine its impact. In many cases, additional data is available from sources other than network traffic-related sources (e.g., data files or host OSs).

Generally, analysts should focus on identifying the most important characteristics of the activity and assessing the negative impact it has caused or may cause the organization. Other actions, such as determining the identity of an external attacker, are typically time-intensive and difficult to accomplish, and do not aid the organization in correcting the operational issues or security weaknesses. Determining the intent of an attacker is also very difficult; for example, an unusual connection attempt could be caused by an attacker, malicious code, misconfigured software, or an incorrect keystroke, among other causes. Although understanding intent is important in some cases, the negative impact of the event should be the primary concern. Establishing the identity of the attacker might be important to the organization, particularly when criminal activity has occurred, but in other cases it should be weighed against other important goals to put it into perspective. The focus of the investigation should be determined at the onset by the appropriate parties, who should decide if learning the identity of the attacker is vital. It is particularly important to seek the advice of legal counsel when developing policies and procedures related to making such decisions, as well as when guidance is needed for a particular situation.

Organizations should be interested not only in analyzing real events, but also in understanding the causes of false alarms. For example, analysts are often well-positioned to identify the root causes of IDS false positives. As merited, analysts should recommend changes to security event data sources that improve detection accuracy.

6.4.4 Attacker Identification

When analyzing most attacks, identifying the attacker is not an immediate, primary concern: ensuring that the attack is stopped and recovering systems and data are the main interests. If an attack is ongoing, such as an extended denial of service attack, organizations might want to identify the IP address used by the attacker so that the attack can be stopped. Unfortunately, this is often not as simple as it sounds. The

following items explain potential issues involving the IP addresses apparently used to conduct an attack:

- **Spoofed IP Addresses.**
- **Many Source IP Addresses.**
- **Validity of the IP Address.**

Several ways of validating the identity of a suspicious host are as follows:

- **Contact the IP Address Owner.** The Regional Internet Registries, such as the American Registry for Internet Numbers (ARIN), provide WHOIS query mechanisms on their Web sites for identifying the organization or person that owns.is responsible for.a particular IP address. This information can be helpful in analyzing some attacks, such as seeing that three different IP addresses generating suspicious activity are all registered to the same owner. However, in most cases, analysts should not contact the owner directly; instead, they should provide information about the owner to the management and legal advisors of the analyst's organization, who can initiate contact with the organization or give the analyst approval to do so if needed.
- **Send Network Traffic to the IP Address.** Organizations should not send network traffic to an apparent attacking IP address to validate its identity. If the IP address is spoofed, sending unsolicited network traffic to the system could be interpreted as unauthorized use or an attack. Under no circumstances should individuals attempt to gain access to others. systems without permission.
- **Seek ISP Assistance.** ISPs generally require a court order before providing any information to an organization about suspicious network activity. Accordingly, ISP assistance is generally an option during only the most serious network-based attacks.
- **Research the History of the IP Address.** Analysts can look for previous suspicious activity associated with the same IP address or IP address block. The organization's own network traffic data archives and incident tracking databases might show previous activity. Possible external sources include Internet search engines and online incident databases that allow searches by IP address.
- Look for Clues in Application Content. Application data packets related to an attack might contain clues to the attacker's identity. In addition to IP addresses, valuable information could include an e-mail address or an Internet relay chat (IRC) nickname.

In most cases, organizations do not need to positively identify the IP address used for an attack.

6.5 Recommendations

Key recommendations presented in this section for using data from network traffic are as follows:

- **Organizations should have policies regarding privacy and sensitive information.** The use of forensic tools and techniques might inadvertently disclose sensitive information to analysts and others involved in forensic activities. Also, long-term storage of sensitive information inadvertently captured by forensic tools might violate data retention policies. Policies should also address the monitoring of networks, as well as requiring warning banners on systems that indicate activity may be monitored.
- **Organizations should provide adequate storage for network activity-related logs.** Organizations should estimate typical and peak log usage, determine how many hours. or days. worth of data should be retained based on the organization's policies, and ensure that systems and applications have sufficient storage available.
- **Organizations should configure data sources to improve the collection of information.** Over time, operational experience should be used to improve the organization's forensic analysis capabilities. Organizations should periodically review and adjust the configuration settings of data sources to optimize capture of relevant information.
- **Analysts should have reasonably comprehensive technical knowledge.** Because current tools have rather limited analysis abilities, analysts should be well-trained, experienced, and knowledgeable in networking principles, common network and application protocols, network and application security products, and network-based threats and attack methods.
- **Analysts should consider the fidelity and value of each data source.** Analysts should have more confidence in original data sources than in data sources that receive normalized data from other sources. The analysts should validate any unusual or unexpected data that is based on interpretation of data, such as IDS and SEM alerts.
- **Analysts should generally focus on the characteristics and impact of the event.** Determining the identity of an attacker and other similar actions are typically time-intensive and difficult to accomplish, and do not aid the organization in correcting operational issues or security weaknesses. Establishing the identity and intent of an attacker may be important, especially if a criminal investigation will ensue, but should be weighed against other important goals, such as stopping an attack and recovering systems and data.

7. Using Data from Applications

Applications such as e-mail, Web browsers, and word processors are what make computers valuable to users. OSs, files, and networks are all needed to support applications: OSs to run the applications, networks to send application data between systems, and files to store application data, configuration settings, and logs. From a forensic perspective, applications bring together files, OSs, and networks. This section describes application architectures — the components that typically make up applications — and provides insights into the types of applications that are most often the focus of forensics. The section also provides guidance on collecting, examining, and analyzing application data.

7.1 Application Components

All applications contain code in the form of executable files (and related files, such as shared code libraries) or scripts. In addition to code, many applications have one or more of the following components: configuration settings, authentication, logs, data, and supporting files.

7.1.1 Configuration Settings

From a forensics perspective, many settings are trivial (e.g., specifying background colors), but others might be very important, such as the host and directory where data files and logs are stored or the default username. Configuration settings may be temporary — set dynamically during a particular application session — or permanent. Many applications have some settings that apply to all users, and also support some user-specific settings. Configuration settings may be stored in several ways, including the following:

- **Configuration File.**
- **Runtime Options.**
- **Added to Source Code.**

7.1.2 Authentication

Some applications verify the identity of each user attempting to run the application. Common authentication methods include the following:

- **External Authentication.** The application may use an external authentication service, such as a directory server. Although the application may contain some records related to authentication, the external authentication service is likely to contain more detailed authentication information.
- **Proprietary Authentication.** The application may have its own authentication mechanism, such as user accounts and passwords that are part of the application, not the OS.
- **Pass-Through Authentication.** Pass-through authentication refers to passing OS credentials (typically, username and password) unencrypted from the OS to the application.

- **Host/User Environment.** Within a controlled environment (e.g., managed workstations and servers within an organization), some applications may be able to rely on previous authentication performed by the OS. This technique is effective only if users cannot alter the user identity in the workstation environment.

7.1.3 Logs

Although some applications (primarily very simple ones) do not record any information to logs, most applications perform some type of logging. An application may record log entries to an OS-specific log (e.g., syslog on UNIX systems, event logs on Windows systems), a text file, a database, or a proprietary file format. Some applications record different types of events to different logs. Common types of log entries are as follows:

- **Event.** Event log entries typically list actions that were performed, the date and time each action occurred, and the result of each action.
- **Audit.** Audit log entries, also known as security log entries, contain information pertaining to audited activities, such as successful and failed logon attempts, security policy changes, file access, and process execution.
- **Error.** Some applications create error logs, which record information regarding application errors, typically with timestamps.
- **Installation.** Information recorded in an installation log varies widely but is likely to include the status of various phases of the installation. The log may also indicate the source of the installation files, the locations where the application components were placed, and options involving the application's configuration.
- **Debugging.** Some applications can be run in a debugging mode, which means that they log far more information than usual regarding the operation of the application.

7.1.5 Data

Nearly every application is specifically designed to handle data in one or more ways, such as creating, displaying, transmitting, receiving, modifying, deleting, protecting, and storing data. Application data often resides temporarily in memory, and temporarily or permanently in files. The format of a file containing application data may be generic (e.g., text files, bitmap graphics) or proprietary. Some applications create temporary files during a session, which may contain application data. If an application fails to shut down gracefully, it may leave temporary files on media. Most OSs have a directory designated for temporary files; however, some applications have their own temporary directory, and other applications place temporary files in the same directory where data is stored. Applications may also contain data file templates and sample data files (e.g., databases, documents).

7.1.5 Supporting Files

Applications often include one or more types of supporting files, such as documentation and graphics. Supporting files tend to be static, but that does not mean that they are not of value for forensics. Types of supporting files include the following:

- **Documentation.**
- **Links.**
- **Graphics.**

7.1.6 Application Architecture

Every application has an architecture, which refers to the logical separation of its components and the communication mechanisms used between components. Most applications are designed to follow one of three major application architecture categories, as follows:

- **Local.** A local application is intended to be contained mainly within a single system. The code, configuration settings, logs, and supporting files are located on the user's system.
- **Client/Server.** A client/server application is designed to be split among multiple systems. A two-tiered client/server application stores its code, configuration settings, and supporting files on each user's workstation, and its data on one or more central servers accessed by all users.
- **Peer-to-Peer.** A peer-to-peer application is designed so that individual client hosts directly communicate and share data with each other. Typically, the clients first communicate with a centralized server that provides information about other clients; this information is then used to establish direct connections that do not need to go through the centralized server.

7.2 Types of Applications

Applications exist for nearly every purpose imaginable. Although forensic techniques can be applied to any application, certain types of applications are more likely to be the focus of forensic analysis, including e-mail, Web usage, interactive messaging, file sharing, document usage, security applications, and data concealment tools. Nearly every computer has at least a few applications installed from these categories.

7.2.1 E-mail

Each e-mail message consists of a header and a body. The body of the e-mail contains the actual content of the message, such as a memorandum or a personal letter. The header of the e-mail includes various pieces of information regarding the e-mail. By default, most e-mail client applications display only a few header fields for each message: the sender's and recipients. e-mail addresses, the date and time the message

was sent, and the subject of the message. However, the header typically includes several other fields, including the following:

- Message ID
- Type of e-mail client used to create the message
- Importance of the message, as indicated by the sender (e.g., low, normal, high)
- Routing information — which e-mail servers the message passed through in transit and when each server received it
- Message content type, which indicates whether the e-mail content simply consists of a text body or also has file attachments, embedded graphics, etc.

Most e-mail clients also provide an address book that can hold contact information, such as e-mail addresses, names, and phone numbers. Encryption programs are sometimes used in conjunction with e-mail clients to encrypt an e-mail's body and/or attachments.

From end to end, information regarding a single e-mail message may be recorded in several places — the sender's system, each e-mail server that handles the message, and the recipient's system, as well as antivirus, spam, and content filtering servers.

7.2.2 Web Usage

Typically, the richest sources of information regarding Web usage are the hosts running the Web browsers. Information that may be retrieved from Web browsers include a list of favorite Web sites, a history (with timestamps) of Web sites visited, cached Web data files, and cookies (including their creation and expiration dates). Another good source of Web usage information is Web servers, which typically keep logs of the requests they receive. Data that is often logged by Web servers for each request includes a timestamp; the IP address, Web browser version, and OS of the host making the request; the type of request (e.g., read data, write data); the resource requested; and the status code. The response to each request includes a three-digit status code that indicates the success or failure of the request. For successful requests, the status code explains what action was performed; for failures, the status code explains why the request failed.

Several other types of devices and software, in addition to Web browsers and servers, might also log related information. For example, Web proxy servers and application proxying firewalls might perform detailed logging of HTTP activity, with a level of detail similar to that of Web server logs. Routers, non-proxying firewalls, and other network devices might log the basic aspects of HTTP network connections, such as source and destination IP addresses and ports. Organizations that use Web content monitoring and filtering services might find useful data in the services. logs, particularly regarding denied Web requests.

7.2.3 Interactive Communications

Unlike e-mail messages, which typically take minutes to go from sender to recipient, interactive communications services provide real-time (or near-real-time) communications. Types of applications commonly used for interactive communications include the following:

- **Group Chat.**
- **Instant Messaging Applications.**
- **Audio and Video.**

7.2.4 File Sharing

Users can share files through many different programs. File sharing programs can be grouped by architecture, as follows:

- **Client/Server.** Traditional file sharing services use client/server architectures, with a central file server containing a repository of files. Clients can use the server by initiating connections to it, authenticating to it (if required), reviewing the list of available files (if needed), then transferring files to or from the server.
- **Peer-to-Peer.** Most peer-to-peer file sharing services are primarily used to trade music, graphics, or software across the Internet. Unlike client/server file sharing, in which a single server holds the file repository, peer-to-peer file sharing is distributed, with files located on many different hosts. Peer-to-peer file sharing services typically have a central server that gives clients information on where other clients are located, but the server does not participate in the transmission of files or file information. Peer-to-peer file sharing services typically require no user authentication. All file browsing and transfers occur directly between the clients (peers).

7.2.5 Document Usage

Many users spend much of their time working with documents, such as letters, reports, and charts. Documents may contain any type of data, so they are often of interest to analysts. Documents often have user or system information embedded in them, such as the name or username of the person who created or most recently edited the document, or the license number of the software or the MAC address of the system used to create the document.

7.2.6 Security Applications

Hosts often run one or more security applications that attempt to protect the host from misuse and abuse occurring through commonly used applications, such as e-mail clients and Web browsers. Some commonly used security applications are antivirus software, spyware detection and removal utilities, content filtering (e.g., anti-spam measures), and host-based intrusion detection software. The logs of security applications may contain detailed records of suspicious activity and may also indicate

whether a security compromise occurred or was prevented. If the security application is part of an enterprise deployment, such as centrally managed and controlled antivirus software, logs may be available both on individual hosts and on a centralized application log.

7.2.7 Data Concealment Tools

Some people use tools that conceal data from others. This might be done for benign purposes, such as protecting the confidentiality and integrity of data against access by unauthorized parties, or for malicious purposes, such as concealing evidence of improper activities. Examples of data concealment tools include file encryption utilities, steganographic tools, and system cleanup tools. The use of most data concealment tools is unlikely to be captured in logs. Analysts should be aware of the capabilities of these tools so that they can identify such tools on a system and recognize the tools effects.

7.3 Collecting Application Data

As described in Section 7.1, application-related data may be located within filesystems, volatile OS data, and network traffic. The types of application data that these sources may contain are as follows:

- **Filesystems.** Filesystems may contain many types of files related to applications, including executable files and scripts, configuration files, supporting files (e.g., documentation), logs, and data files.
- Volatile OS Data. Volatile OS data may contain information about network connections used by applications, the application processes running on a system and the command line arguments used for each process, and the files held open by applications, as well as other types of supporting information.
- **Network Traffic.** The most relevant network traffic data involves user connections to a remote application and communications between application components on different systems. Other network traffic records might also provide supporting information, such as network connections for remote printing from an application, and DNS lookups by the application client or other components to resolve application components. domain names to IP addresses.

Analysts often face a major challenge in determining which data should be collected. In many cases, the analyst must first decide which application is of interest. In some situations, collecting the necessary data might involve identifying all components of the application, deciding which were most likely to be of interest (based on the details of the situation and the need), finding the location of each component, and collecting data from those components.

7.4 Examining and Analyzing Application Data

Examining and analyzing application data largely consists of looking at specific portions of application data — filesystems, volatile OS data, and network traffic

— using the tools and techniques described in Sections 4.3 and 4.4, 5.3, and 6.4, respectively.

In some cases, analysts bring together pertinent application data from several varied application data sources; this is largely a manual process. Detailed analysis of application-related events and event reconstruction usually require a skilled and knowledgeable analyst who understands the information presented by all the sources. The analyst can review the results of the examination and analysis of individual application data sources and see how the information fits together. Tools that may be helpful to analysts include security event management software (described in Section 6.2.5), which can correlate some application-related events among multiple data sources, and log analysis software (including some types of host-based intrusion detection software), which can be run against certain types of logs to identify suspicious activity.

7.5 Recommendations

The key recommendations presented in this section for using data from applications are as follows:

- **Analysts should consider all possible application data sources.** Application events might be recorded by many different data sources. In addition, applications might be used through multiple mechanisms, such as multiple client programs installed on a system and Web-based client interfaces. In such situations, analysts should identify all application components, decide which are most likely to be of interest, find the location of each component of interest, and collect the data.
- **Analysts should bring together application data from various sources.** The analyst should review the results of the examination and analysis of individual application data sources and determine how the information fits together, to perform a detailed analysis of application-related events and event reconstruction.

8. Using Data from Multiple Sources

This section of the guide presents two examples of the use of multiple data sources during digital forensics. Each example describes a scenario, indicates a specific need for forensic analysis, and presents an explanation of how the forensic process might be performed. The explanations also illustrate how complex the process can be. The examples presented in this section are as follows:

- Determining which worm has infected a system and identifying the worm.s characteristics
- Reconstructing the sequence of cyber events involving a threatening e-mail.

8.1 Suspected Network Service Worm Infection

An organization's help desk receives several calls in a short time from users complaining about a particular server providing slow responses. The help desk sends a trouble ticket to the monitoring group. That group's network IDSs have recently reported several unusual alerts involving the server, and the analyst who reviewed the alerts believes that they may be accurate. The data in the alerts indicates that some suspicious activity was directed at the server, and the server is now generating identical activity directed at other systems. The intrusion detection analyst's initial hypothesis is that a worm may have attacked a vulnerable network service and infected the server, which is now attempting to infect other systems. The monitoring group contacts the incident handler on duty to investigate the possible incident on the server.

For the incident, this particular incident handler's role is to determine the type of worm that has infected the system and to identify the distinguishing characteristics of the worm. This information is critical to the incident response team's ability to effectively perform containment, eradication, and recovery activities and to prevent other systems within the organization from becoming infected. If the incident handler's investigation shows that the incident was probably caused by something other than a worm, the characteristics the handler identifies should be very helpful in determining what actually occurred.

Information regarding this incident may be recorded in several different places. The incident handler should check the data sources that are most likely to have relevant information first, based on the handler's previous experience with the data sources and the initial information available regarding the incident. For example, because network IDS sensors saw the suspicious activity, other network-based data sources monitoring the same network segment might also contain relevant information. If the organization uses security event management or network forensic analysis tool software, which bring together data from many different sources, the incident handler may be able to gather all necessary information just by running a few queries from the SEM or NFAT console. If a centralized source of data is not available, the handler should check individual potential sources of attack characteristics, such as the following:

- **Network IDS.**
- **Network-Based Firewall.**
- **Host IDS and Firewall.**
- **Antivirus Software.**
- **Application Logs.**

The goal in the initial information gathering effort is to identify enough characteristics for positive identification of the worm. This can be challenging, particularly for worms that have dozens of variants; these variants often have similar characteristics but have different effects on systems. Analysts can perform queries on antivirus vendors. malware databases, searching for identified characteristics such as product name, service name or port number, text strings within the malware, and files or settings modified on the target. Virtually any instance of malware, other than the latest threats (e.g., released in the past several hours), is likely to be included in major malware databases. Each database entry typically contains extensive information on how the worm spreads, how it affects systems (e.g., what changes it makes), and how it can be eradicated, including measures concerning prevention of infections on other systems.

If a search of malware databases does not lead to identification of the worm, then the incident handler may need to perform additional research and analysis to discover the information usually provided by malware database entries. Although the organization can send a copy of the worm to the organization's antivirus vendor for analysis and identification, the organization should perform its own analysis in the meantime, since the time frame for the vendor's response is unknown. To gather more information, the analyst can examine the infection through the following methods:

- **Current State of the Host.** The analyst can examine the host to look at several aspects of its current state. In this case, it is probably most effective to examine the network connections listing to identify unusual connections (e.g., large number, unexpected port number usage, unexpected hosts) and unexpected listening ports (e.g., backdoors created by the worm). Other steps that may be useful include identifying unknown processes in the running processes list and examining the host's logs to reveal any unusual entries that may be related to the infection.
- **Host's Network Activity.** The analyst can collect worm traffic being generated by the infected server through a packet sniffer and protocol analyzer. This may provide enough additional information regarding the characteristics of the worm to enable the analyst to locate it in major malware databases.

Worm incidents often necessitate as rapid a response as possible, because an infected system may be attacking other systems inside and outside the organization. In addition, worms often install backdoors and other tools on systems that permit attackers to gain remote access to infected systems, which can create additional damage. Accordingly, organizations may choose to disconnect infected systems from networks immediately, instead of performing data collection for the host first. This step may make it considerably more difficult for analysts to identify a worm and to determine

its effects on systems — for example, if systems are disconnected from the network, network activity and certain aspects of the host state will not be available. In such cases, the analyst may need to perform a more detailed forensic analysis of the server, such as collecting its filesystems and examining them for signs of malicious activity (e.g., altered system executables) to determine exactly what happened to the server. The analyst can also examine non-volatile characteristics of the server's OS, such as looking for administrative-level user accounts and groups that may have been added by the worm. Ultimately, the analyst should gather enough information to identify the worm's behavior in sufficient detail to enable the incident response team to act effectively to contain, eradicate, and recover from the incident.

8.2 Threatening E-mail

An incident handler responds to a request for assistance with an internal investigation. An employee has been accused of sending a threatening e-mail to another employee through the organization's e-mail system. The incident handler has been asked to help investigators find all data sources that may contain records of the e-mail. This information will be helpful to investigators in determining who sent the e-mail. Because e-mail can be forged easily, it is important to use all available data sources to reconstruct the sequence of events for creating, sending, and receiving the e-mail. Also, the incident handler needs to perform all work using forensically sound tools, techniques, and procedures, and to document all actions performed.

The threatening e-mail is key to the investigation, and its header contains the most important information to the incident handler. It should contain the domain name and the IP address of the host that sent the e-mail, the type of e-mail client used to send the e-mail, the e-mail's message ID, and the date and time the e-mail was sent. The e-mail header should also list each e-mail server (domain name and IP address) that the message passed through and the date and time each server processed the e-mail. Because the e-mail was supposedly sent using the organization's e-mail system, the e-mail header should only list systems within the organization. Assuming that this is the case, the incident handler can check each system on the list for correlating information.

Because of the importance of the threatening e-mail, the incident handler should focus first on collecting a copy of the e-mail, including its header. Depending on the type of e-mail client used by the recipient and its configuration, the e-mail may have been downloaded to the recipient's workstation, or it may remain on the e-mail server. It is also possible that the e-mail is still stored in both locations. The incident handler should collect copies of the e-mail from multiple sources, if possible, to confirm that the content of the e-mail has been unchanged in transit or by the recipient.

After reviewing the header, the incident handler should next gather more information about the sending of the e-mail. The header should list the IP address and the e-mail client used by the sender; the incident handler should determine which host was using that IP address at the time the e-mail was sent. There are three possibilities for the IP address:

- **Local E-mail Client.** In this case, the incident handler should be able to use network records, such as DHCP logs, to identify the desktop, laptop, PDA, or other device used to send the e-mail. The incident handler can then create images of the identified device and examine an image copy to look for malware and for records related to the e-mail. If the message cannot be found intact on the system, collecting data from the device's memory and filesystems, including deleted and temporary files, might lead to the identification of fragments of the e-mail. In addition, security controls on the device, such as spam filtering and antivirus software, might have scanned the outgoing e-mail and logged a record of it. It is also possible, but unlikely, that a copy of the e-mail is stored on an e-mail server. In addition to looking for records of the e-mail on the local host, the incident handler should analyze the authentication records on the host to determine which user account was in use when the e-mail was sent.
- **Server-Based E-mail Client.** If the organization offers a server-based client, such as a Web-based e-mail interface, then the IP address may correspond to that server. Typically, the use of such a server requires users to authenticate themselves, so there may be authentication records that indicate when the alleged sender logged on to the server and what IP address the user's system was using. The incident handler can then determine which system was assigned that IP address at the time, perform bit stream imaging for the identified system, and examine a copy of the image for the malware and the e-mail. For example, temporary files from the Web browser may contain a copy of the e-mail.s contents.
- **Spoofed.** If the IP address was fabricated — for example, if it is not a valid address within the organization's networks — the incident handler should rely on other data sources to identify the host that actually sent the e-mail message.

The organization's e-mail servers are another likely source of information. Each of the server IP addresses listed in the e-mail header should contain some record of the e-mail, including the message ID value, which should facilitate quick identification of pertinent records. As mentioned previously, the final e-mail server in the list may contain a copy of the e-mail. Backups of that server may contain a copy of the e-mail, but only if it was held there for delivery for several hours or more. Other services associated with e-mail, such as antivirus software and spam filters, may also contain basic records of e-mail activity, but are unlikely to contain many details. Another possible source of information is authentication records. Although few e-mail servers require users to authenticate to send e-mail, they typically do require authentication to deliver e-mail to users. Because users often send and receive e-mail during a single session, authentication logs may contain records for receiving e-mail that can be helpful in determining who may have sent a particular e-mail.

Another possible source of information is a record of the network traffic generated by sending or receiving the e-mail. Packet sniffers or network forensic analysis tools that were monitoring network activity might have captured the activity, including the actual IP addresses of the sending or receiving hosts, the contents and header of the e-mail, and any associated authentication activity.

Ultimately, the incident handler should identify the hosts that were used to send and receive the e-mail, as well as all intermediate hosts that transferred the e-mail from sender to receiver. The incident handler should collect copies of the e-mail and supporting information from each relevant host and, using the timestamps in records, should re-create the sequence of events from a cyber perspective. For example, a possible sequence might be as follows:

A user logged on to a particular desktop computer at 8:37 a.m. At 10:02 a.m., the threatening e-mail was sent from that computer using its built-in e-mail client. The e-mail passed through three of the organization's e-mail servers and was stored on Server 4 to await retrieval by the intended recipient. The recipient user logged onto a particular laptop computer at 11:20 a.m. and downloaded e-mail, including the threatening e-mail, at 11:23 a.m. The contents of the e-mail on the recipient's computer, as well as user-provided header fields (e.g., From, To, Subject), were identical to a copy saved in the Sent folder on the first user's desktop computer.

This information can be used as a basis for further investigation; although it documents cyber activities, it does not tell the whole story. For example, it cannot determine which person actually sent the e-mail from the desktop computer, only which user account was in use at the time. The incident handler could analyze the desktop computer in question to verify its integrity, such as comparing its security settings and controls to the organization's baseline settings, checking its clock settings, and checking the system for signs of compromise and other breaches of security.

8.3 Recommendations

The key recommendations presented in this section for using data from multiple sources are as follows:

- **Analysts can handle many situations most effectively by analyzing several individual data sources and then correlating events among them.** The techniques and processes for collecting, examining, and analyzing different types of data sources are fundamentally different. Many applications have data captured in data files, OSs, and network traffic.
- **Organizations should be aware of the technical and logistical complexity of analysis.** A single event can generate records on many different data sources and produce more information than analysts can feasibly review. Tools such as SEM can assist analysts by bringing information from many data sources together in a single place.

Appendix A – Recommendations

A.1 Organizing a Forensics Capability

- **Organizations should have a capability to perform computer and network forensics.** Forensics is needed for various tasks within an organization, including investigating crimes and inappropriate behavior, reconstructing computer security incidents, troubleshooting operational problems, supporting due diligence for audit record maintenance, and recovering from accidental system damage. Without such a capability, an organization will have difficulty determining what events have occurred within its systems and networks, such as exposures of protected, sensitive data. Also, handling evidence in a forensically sound manner puts decision makers in a position where they can confidently take the necessary actions.

A1.1 Forensic Participants

- **Organizations should determine which parties should handle each aspect of forensics.** Most organizations rely on a combination of their own staff and external parties to perform forensic tasks. Organizations should decide which parties should take care of which tasks based on skills and abilities, cost, response time, and data sensitivity.
- **Analysts should have reasonably comprehensive technical knowledge.** Because current tools have rather limited analysis abilities, analysts should be well-trained, experienced, and knowledgeable in networking principles, common network and application protocols, network and application security products, and network-based threats and attack methods.
- **Incident handling teams should have robust forensic capabilities**. More than one team member should be able to perform each typical forensic activity. Hands-on exercises and IT and forensic training courses can be helpful in building and maintaining skills, as can demonstrations of new tools and technologies.
- **Many teams within an organization should participate in forensics.** Individuals performing forensic actions should be able to reach out to other teams and individuals within an organization, as needed, for additional assistance. Examples of teams that may provide assistance in these efforts include IT professionals, management, legal advisors, human resources personnel, auditors, and physical security staff. Members of these teams should understand their roles and responsibilities in forensics, receive training and education on forensic-related policies, guidelines, and procedures, and be prepared to cooperate with and assist others on forensic actions.

Forensic Policies, Guidelines, and Procedures

- **Forensic considerations should be clearly addressed in policies.** At a high level, policies should allow authorized personnel to monitor systems and networks and perform investigations for legitimate reasons under appropriate

circumstances. Organizations may also have a separate forensic policy for incident handlers and others with forensic roles that provides more detailed rules for appropriate behavior. Everyone who may be called upon to assist with any forensic efforts should be familiar with and understand the forensic policy. Additional policy considerations are as follows:

- Forensic policy should clearly define the roles and responsibilities of all people performing or assisting with the organization's forensic activities. The policy should include all internal and external parties that may be involved and should clearly indicate who should contact which parties under different circumstances.
- The organization's policies, guidelines, and procedures should clearly explain what forensic actions should and should not be performed under normal and special circumstances and should address the use of anti-forensic tools and techniques. Policies, guidelines, and procedures should also address the handling of inadvertent exposures of sensitive information.
- Incorporating forensic considerations into the information system life cycle can lead to more efficient and effective handling of many incidents.
- The organization's policies should address inadvertent disclosures and long-term storage of sensitive information captured by forensic tools, and should ensure that this does not violate the organization's privacy or data retention policies.
- The organization's policies should also address the monitoring of networks, as well as requiring warning banners on systems that indicate that activity might be monitored. The policies should take into account reasonable expectations of user privacy.

- Organizations should create and maintain guidelines and procedures for performing forensic tasks. The guidelines should include general methodologies for investigating an incident using forensic techniques, and step-by-step procedures should explain how to perform routine tasks. The guidelines and procedures should support the admissibility of evidence into legal proceedings. Because electronic logs and other records can be altered or otherwise manipulated, organizations should be prepared, through their policies, guidelines, and procedures, to demonstrate the reliability and integrity of such records. The guidelines and procedures should also be reviewed regularly and maintained so that they are accurate.

A.1.3 Technical Preparation

- Analysts should have a forensic toolkit for data collection, examination, and analysis. It should contain various tools that provide the ability to collect and examine volatile and non-volatile data and to perform quick reviews of data as well as in-depth analysis. The toolkit should allow its applications to be

run quickly and efficiently from removable media (e.g., floppy disk, CD) or a forensic workstation.

- Organizations should provide adequate storage for network activity-related logs. Organizations should estimate typical and peak log usage, determine how many hours or days. worth of data should be retained based on the organization's policies, and ensure that systems and applications have sufficient storage available. Logs related to computer security incidents might need to be kept for a substantially longer period of time than other logs.

A.2 Performing the Forensics Process

- Organizations should perform forensics using a consistent process. This guide presents a four-phase forensics process, with collection, examination, analysis, and reporting phases. The exact details of the phases may vary based on the need for forensics.

A.2.1

Data Collection

- **Organizations should be proactive in collecting useful data.** Configuring auditing on OSs, implementing centralized logging, performing regular system backups, and using security monitoring controls can all generate sources of data for future forensic efforts.
- **Analysts should be aware of the range of possible data sources.** Analysts should be able to survey a physical area and recognize possible sources of data. Analysts should also think of possible data sources located elsewhere within an organization and outside the organization. Analysts should be prepared to use alternate data sources if it is not feasible to collect data from a primary source.
- **Analysts should consider all possible application data sources.** Application events might be recorded by many different data sources. In addition, applications might be used through multiple mechanisms, such as multiple client programs installed on a system and Web-based client interfaces. In such situations, analysts should identify all application components, decide which are most likely to be of interest, find the location of each component of interest, and acquire the data.
- **Analysts should perform data collection using a standard process.** The recommended steps are identifying sources of data, developing a plan to acquire the data, acquiring the data, and verifying the integrity of the data. The plan should prioritize the data sources, establishing the order in which the data should be acquired, based on the likely value of the data, the volatility of the data, and the amount of effort required. Before data collection begins, a decision should be made by the analyst or management regarding the need to collect and preserve evidence in a manner that supports its use in future legal or internal disciplinary proceedings. In such situations, a clearly defined chain of custody should be followed to avoid allegations of mishandling or tampering of evidence.

- **Analysts should act appropriately to preserve volatile OS data.** The criteria for determining whether volatile OS data must be preserved should be documented in advance so that analysts can make informed decisions as quickly as possible. To determine whether the effort required to collect volatile OS data is warranted, the risks associated with such collection should be weighed against the potential for recovering important information.
- **Analysts should use a forensic toolkit for collecting volatile OS data.** Use of a forensic toolkit enables accurate OS data to be collected while minimizing disturbance to the system and protecting the tools from changes. The analyst should know how each tool is likely to affect or alter the system during collection of data.
- **Analysts should choose the appropriate shutdown method for each system.** Each way of shutting down a particular OS can cause different types of data to be preserved or corrupted; analysts should be aware of the typical shutdown behavior of each OS.
- **Analysts should preserve and verify file integrity.** Using a write-blocker during backups and imaging prevents a computer from writing to its storage media. The integrity of copied data should be verified by computing and comparing the message digests of files. Backups and images should be accessed as read-only whenever possible; write-blockers can also be used to prevent writes to the backup or image file or restored backup or image.

A.2.2 Examination and Analysis

- **Analysts should use a methodical approach to studying the data**. The foundation of forensics is using a methodical approach in analyzing the available data so that analysts can either draw appropriate conclusions based on the available data, or determine that no conclusion can yet be drawn. If evidence might be needed for legal or internal disciplinary actions, analysts should carefully document the findings and the steps that were taken.
- **Analysts should examine copies of files, not the original files.** During the collection phase, the analyst should make multiple copies of the desired files or filesystems, typically a master copy and a working copy. The analyst can then work with the working copy of the files without affecting the original files or the master copy. A bit stream image should be performed if evidence may be needed for prosecution or disciplinary actions, or if preserving file times is important.
- **Analysts should consider the fidelity and value of each data source.** Analysts should have more confidence in original data sources than in data sources that receive normalized data from other sources. Analysts should validate any unusual or unexpected data that is based on interpreting data, such as IDS and SEM alerts.
- **Analysts should rely on file headers, not file extensions, to identify file content types.** Because users can assign any file extension to a file, analysts should not assume that file extensions are accurate. Analysts can identify the

type of data stored in many files by examining their file headers. Although people can alter file headers to conceal actual file types, this is much less common than altering file extensions.

- **Analysts should generally focus on the characteristics and impact of the event.** Determining the identity of an attacker and other similar actions are typically time-intensive and difficult to accomplish, and do not aid the organization in correcting operational issues or security weaknesses. Establishing the identity and intent of an attacker may be important, especially if a criminal investigation will ensue, but it should be weighed against other important goals.
- **Organizations should be aware of the technical and logistical complexity of analysis.** A single event can generate records on many different data sources and produce more information than analysts can feasibly review. Tools such as SEM can assist analysts by bringing information from many data sources together in a single place.
- **Analysts should bring together data from various sources.** The analyst should review the results of the examination and analysis of individual data sources, such as data files, OSs, and network traffic, and determine how the information fits together, to perform a detailed analysis of application-related events and event reconstruction.

A.2.3 Reporting

- **Analysts should review their processes and practices.** Reviews of current and recent forensic actions can help identify policy shortcomings, procedural errors, and other issues that might need to be remedied, as well as ensuring that the organization stays current with trends in technology and changes in law.

Glossary

Selected terms used in the Guide to Integrating Forensic Techniques into Incident Response are defined below.

Agent: A program used in distributed denial of service (DDoS) attacks that sends malicious traffic to hosts based on the instructions of a handler.

Analysis: The third phase of the computer and network forensic process, which involves using legally justifiable methods and techniques, to derive useful information that addresses the questions that were the impetus for performing the collection and examination.

Anti-Forensic: A technique for concealing or destroying data so that others cannot access it.

Baselining: Monitoring resources to determine typical utilization patterns so that significant deviations can be detected.

Blended Attack: Malicious code that uses multiple methods to spread.

Bit Stream Imaging: A bit-for-bit copy of the original media, including free space and slack space. Also known as disk imaging.

Boot Sector Virus: A virus that plants itself in a system's boot sector and infects the master boot record.

Cluster: A group of contiguous sectors.

Computer Forensics: The practice of gathering, retaining, and analyzing computer-related data for investigative purposes in a manner that maintains the integrity of the data.

Computer Security Incident: See "incident."

Computer Security Incident Response Team (CSIRT): A capability set up for the purpose of assisting in responding to computer security-related incidents; also called a Computer Incident Response Team (CIRT) or a CIRC (Computer Incident Response Center, Computer Incident Response Capability).

Collection: The first phase of the computer and network forensics process, which involves identifying, labeling, recording, and acquiring data from the possible sources of relevant data, while following guidelines and procedures that preserve the integrity of the data.

Data: Distinct pieces of digital information that have been formatted in a specific way.

Denial of Service (DoS): An attack that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources.

Digital Forensics: The application of science to the identification, collection, examination, and analysis, of data while preserving the integrity of the information and maintaining a strict chain of custody for the data.

Directory: Organizational structures that are used to group files together.

Disk Imaging: Generating a bit-for-bit copy of the original media, including free space and slack space. Also known as a bit stream image.

Disk-to-Disk Copy: Copying the contents of media directly to another media.

Disk-to-File Copy: Copying the contents of media to a single logical data file.

Distributed Denial of Service (DDoS): A DoS technique that uses numerous hosts to perform the attack.

Egress Filtering: The process of blocking outgoing packets that use obviously false Internet Protocol (IP) addresses, such as source addresses from internal networks.

Event: Any observable occurrence in a network or system.

Examination: The second phase of the computer and network forensics process, which involves forensically processing large amounts of collected data using a combination of automated and manual methods to assess and extract data of particular interest, while preserving the integrity of the data.

False Negative: Incorrectly classifying malicious activity as benign.

False Positive: Incorrectly classifying benign activity as malicious.

File: A collection of information logically grouped into a single entity and referenced by a unique name, such as a filename.

File Allocation Unit: A group of contiguous sectors, also known as a cluster.

File Header: Data within a file that contains identifying information about the file and possibly metadata with information about the file contents.

File Infector Virus: A virus that attaches itself to a program file, such as a word processor, spreadsheet application, or game.

File Integrity Checker: Software that generates, stores, and compares message digests for files to detect changes to the files.

Filename: A unique name used to reference a file.

Filesystem: A method for naming, storing, organizing, and accessing files on logical volumes.

Forensics: See "computer forensics."

Forensic Science: The application of science to the law.

Forensically Clean: Digital media that is completely wiped of all data, including nonessential and residual data, scanned for malware, and verified before use.

Free Space: An area on media or within memory that is not allocated.

Handler: A type of program used in DDoS attacks to control agents distributed throughout a network. Also refers to an incident handler, which refers to a person who performs incident response work.

Honeypot: A host that is designed to collect data on suspicious activity and has no authorized users other than its administrators.

Inappropriate Usage: A person who violates acceptable computing use policies.

Incident: A violation or imminent threat of violation of computer security policies, acceptable use policies, or standard computer security practices.

Incident Handling: The mitigation of violations of security policies and recommended practices.

Incident Response: See "incident handling."

Indication: A sign that an incident may have occurred or may be currently occurring.

Ingress Filtering: The process of blocking incoming packets that use obviously false IP addresses, such as reserved source addresses.

Intrusion Detection System (IDS): Software that looks for suspicious activity and alerts administrators.

Logical Backup: A copy of the directories and files of a logical volume.

Logical Volume: A partition or a collection of partitions acting as a single entity that has been formatted with a filesystem.

Macro Virus: A virus that attaches itself to documents and uses the macro programming capabilities of the document's application to execute and propagate.

Malicious Code: A virus, worm, Trojan horse, or other code-based entity that infects a host.

Message Digest: A hash that uniquely identifies data. Changing a single bit in the data stream used to generate the message digest will yield a completely different message digest.

Metadata: Data about data. For filesystems, metadata is data that provides information about a file's contents.

Mobile Code: Software that is transmitted from a remote system to a local system, then executed on the local system without the user's explicit instruction; examples of mobile code software are Java, JavaScript, VBScript, and ActiveX.

Multiple Component Incident: A single incident that encompasses two or more incidents.

Network Address Translation: The process of mapping addresses on one network to addresses on another network.

Network Intrusion Detection System: Software that performs packet sniffing and network traffic analysis to identify suspicious activity and record relevant information.

Network Traffic: Computer network communications that are carried over wired or wireless networks between hosts.

Non-Volatile Data: Data that persists even after a computer is powered down.

Normalize: The process by which differently formatted data is converted into a standardized format and labeled consistently.

Operating System: A program that runs on a computer and provides a software platform on which other programs can run.

Packet: The logical unit of network communications produced by the transport layer.

Packet Sniffer: Software that monitors network traffic on wired or wireless networks and captures packets.

Partition: A logical portion of a media that functions as though it were physically separate from other logical portions of the media.

Patch Management: The process of acquiring, testing, and distributing patches to the appropriate administrators and users throughout the organization.

Port Scanning: Using a program to remotely determine which ports on a system are open (e.g., whether systems allow connections through those ports).

Precursor: A sign that an attacker may be preparing to cause an incident.

Process: An executing program.

Profiling: Measuring the characteristics of expected activity so that changes to it can be more easily identified.

Protocol Analyzer: Software that can reassemble streams from individual packets and decode communications that use various protocols.

Proxy: Software that receives a request from a client, then sends a request on the client's behalf to the desired destination.

Remote Access Server: Devices, such as virtual private network gateways and modem servers, that facilitate connections between networks.

Reporting: The final phase of the computer and network forensic process, which involves reporting the results of the analysis; this may include describing the actions used, explaining how tools and procedures were selected, determining what other actions need to be performed (e.g., forensic examination of additional data sources, securing identified vulnerabilities, improving existing security controls), and providing

recommendations for improvement to policies, guidelines, procedures, tools, and other aspects of the forensic process. The formality of the reporting step varies greatly depending on the situation.

Risk: The probability that one or more adverse events will occur.

Rootkit: A set of tools used by an attacker after gaining root-level access to a host to conceal the attacker's activities on the host and permit the attacker to maintain root-level access to the host through covert means.

Scanning: Sending packets or requests to another system to gain information to be used in a subsequent attack.

Sector: The smallest unit that can be accessed on media.

Security Event Management Software: Software that imports security event information from multiple data sources, normalizes the data, and correlates events among the data sources.

Signature: A recognizable, distinguishing pattern associated with an attack, such as a binary string in a virus or a particular set of keystrokes used to gain unauthorized access to a system.

Slack Space: The unused space in a file allocation block or memory page that may hold residual data.

Social Engineering: An attempt to trick someone into revealing information (e.g., a password) that can be used to attack systems or networks.

Steganography: Embedding data within other data to conceal it.

Subdirectory: A directory contained within another directory.

Threat: The potential source of an adverse event.

Trojan Horse: A nonself-replicating program that seems to have a useful purpose, but in reality has a different, malicious purpose.

Unauthorized Access: A person gains logical or physical access without permission to a network, system, application, data, or other resource.

Victim: A machine that is attacked.

Virus: A self-replicating program that runs and spreads by modifying other programs or files.

Virus Hoax: An urgent warning message about a nonexistent virus.

Volatile Data: Data on a live system that is lost after a computer is powered down.

Vulnerability: A weakness in a system, application, or network that is subject to exploitation or misuse.

Wiping: Overwriting media or portions of media with random or constant values to hinder the collection of data.

Write-Blocker: A tool that prevents all computer storage media connected to a computer from being written to or modified.

Worm: A self-replicating, self-propagating, self-contained program that uses networking mechanisms to spread itself.

World Headquarters
215 North Marengo Avenue
Pasadena, CA 91101
Phone: 626.229.9191
Fax: 626.229.9199

New York Office
551 5th Avenue, Suite 400
New York, NY 10176
Phone: 212.277.3700
Fax: 212.277.3707

San Francisco Office
2200 Powell Street, Suite 800
Emeryville, CA 94608
Phone: 510.652.5011
Fax: 510.652.5018

Washington D.C. Office
Loudoun Tech Center
21400 Ridgetop Circle, Suite 101
Sterling, VA 20166
Phone: 703.433.5400
Fax: 703.433.5368

Houston Office
1300 Post Oak Blvd., Suite 550
Houston, TX 77056
Phone: 832.200.9068
Fax: 832.200.9069

EMEA Headquarters
Thames Central, Fifth Floor
Hatfield Road
Slough, Berkshire SL1 1QE
Phone: +44 (0)175.355.2252
Fax: +44 (0)175.355.2232

www.guidancesoftware.com