National Institute of
Standards and Technology
Technology Administration
U.S. Department of Commerce

Special Publication 800-86
(Draft)

Guide to Computer and Network Data Analysis: Applying Forensic Techniques to Incident Response

Recommendations of the National Institute of Standards and Technology

Tim Grance
Suzanne Chevalier
Karen Kent
Hung Dang

NIST Special Publication 800-86 (Draft)

Guide to Computer and Network Data Analysis: Applying Forensic Techniques to Incident Response (Draft)

Recommendations of the National Institute of Standards and Technology

Tim Grance, Suzanne Chevalier, Karen Kent, Hung Dang

COMPUTER SECURITY

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

August 2005

U.S. Department of Commerce

Carlos M. Gutierrez, Secretary

Technology Administration

Michelle O'Neill, Acting Under Secretary
of Commerce for Technology

National Institute of Standards and Technology

William A. Jeffrey, Director

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.

National Institute of Standards and Technology Special Publication 800-86 (Draft)
Natl. Inst. Stand. Technol. Spec. Publ. 800-86, 107 pages (August 2005)

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.

Acknowledgements

The authors, Tim Grance of the National Institute of Standards and Technology (NIST), and Suzanne Chevalier, Karen Kent, and Hung Dang of Booz Allen Hamilton, wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content. The authors would like to acknowledge Rick Ayers, Wayne Jansen, Peter Mell, and Murugiah Souppaya of NIST, and Adam Feldman, Mike Noblett, and Joseph Nusbaum of Booz Allen Hamilton, for their keen and insightful assistance throughout the development of the document. Additional acknowledgements will be added to the final version of the publication.

Trademarks

Table of Contents

List of Appendices

List of Figures

List of Tables

Executive Summary

The term *data* refers to distinct pieces of digital information that have been formatted in a specific way. Organizations have an ever-increasing amount of data from many sources; for example, data may be stored or transferred by standard computer systems, networking equipment, computing peripherals, personal digital assistants (PDA), and consumer electronics devices, among other data sources. Data analysis can be used for many purposes, such as reconstructing computer security incidents, troubleshooting operational problems, and recovering from accidental system damage. Practically every organization needs to have some capability to perform computer and network data analysis. Accordingly, this guide provides detailed information on establishing data analysis capabilities, including the development of policies and procedures.

Traditionally, computer data analysis has been associated with data on a computer's storage media, while network data analysis has been associated with data passing through a network. As analysis tools and techniques have matured, the two disciplines have become more intertwined. A combined computer and network data analysis capability is increasingly important for incident handling and operational support. For both computer and network data analysis, the analysis process is composed of the following phases:

1. **Acquisition**: acquiring data from the possible sources of relevant data, following procedures that preserve the integrity of the data.

2. **Examination**: using automated methods to sift through acquired data and extract and identify data of particular interest.

3. **Utilization**: reporting the results of the examination, which may include the actions used in the examination and recommendations for improvement.

4. **Review**: performing reviews of processes and practices within the context of the current task to identify policy shortcomings, procedural errors, and other problems that need to be addressed. Lessons learned during the review phase should be incorporated into future data analysis efforts.

This guide provides general recommendations for performing the data analysis process. It also provides detailed information on using the 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 acquisition and examination 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.

The data analysis techniques and processes presented in this guide are based on principles of digital forensics. Forensic science is generally defined as the application of science to the law. Digital forensics, also known as computer and network forensics, has many definitions. Generally, digital forensics is considered to be the application of science to the identification, collection, analysis, and examination of digital evidence while preserving the integrity of the information and maintaining a strict chain of custody for the evidence. Computer and network data analysis is similar to digital forensics and uses many of the same techniques and tools, but data analysis does not necessarily include all of the actions necessary for preserving the integrity of all information collected, nor does it include keeping a chain of custody or other evidence preservation actions. **Accordingly, this publication should not be used as a guide for executing a digital forensic investigation, construed as legal advice, or used as the basis for performing investigations of criminal activity.**

Implementing the following recommendations should facilitate efficient and effective computer and network data analysis activities for Federal departments and agencies.

Organizations should ensure that their policies contain clear statements that address all major data analysis considerations, such as performing monitoring and conducting regular reviews of data analysis policies and procedures.

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 data analysis policy for incident handlers and others with data analysis roles that provides more detailed rules for appropriate behavior. Data analysis policy should clearly define the roles and responsibilities of all people and external organizations performing or assisting with the organization's data analysis activities. The policy should clearly indicate who should contact which internal teams and external organizations under different circumstances.

Organizations should create and maintain procedures for performing data analysis tasks, based on the organization's policies and data analysis activity participants, as well as all applicable laws and regulations.

Procedures should focus on general methodologies for investigating incidents using data analysis techniques, since it is not feasible to develop comprehensive procedures tailored to every possible situation. However, organizations should consider developing step-by-step procedures for performing routine tasks. The procedures should facilitate consistent, effective, and accurate data analysis actions. Procedures should be reviewed periodically, as well as when significant changes are made to the team's policies and procedures.

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

Organizations' policies and procedures should clearly explain what data analysis actions should and should not be performed under various circumstances, as well as describing the necessary safeguards for sensitive information that might be recorded by analysis tools, such as passwords, personal data (e.g., Social Security numbers), and the contents of e-mails. Legal advisors should carefully review all data analysis policy and high-level procedures.

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

IT professionals throughout an organization, especially incident handlers and other first responders to incidents, should understand their roles and responsibilities for data analysis, receive training and education on data analysis-related policies and procedures, and be prepared to cooperate with and assist others when the technologies that they are responsible for are part of an incident or other event. IT professionals should also consult closely with legal counsel both in general preparation for data analysis activities, such as determining which actions IT professionals should and should not perform, and also on an as-needed basis to discuss specific data analysis situations. Also, management should be responsible for supporting data analysis capabilities, reviewing and approving data analysis policy, and approving certain data analysis actions, such as taking mission-critical systems off-line.

1. Introduction

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,” as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

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.

1.2 Purpose and Scope

This publication seeks to assist organizations in investigating computer security incidents and troubleshooting some information technology (IT) operational problems by providing practical guidance on analyzing data from computers and networks. Specifically, the document includes a description of the processes for performing effective data analysis, and also provides advice regarding different data sources, including files, operating systems, network traffic, and applications.

The publication is not to be used as a guide for executing a digital forensic investigation, construed as legal advice, or used as the basis for performing investigations of criminal activity. Its purpose is to inform readers of various technologies and potential ways to use them when performing incident response or troubleshooting activities. Readers are advised to apply the recommended practices only after consultation with management and legal counsel for compliance with laws and regulations (i.e., local, state, federal, and international) that pertain to their situation.

1.3 Audience

This document has been created for incident response teams; system, network, and security administrators; and computer security program managers who are responsible for acquiring and examining data for incident response or troubleshooting purposes. The practices recommended in this guide are designed to highlight key principles associated with the analysis of computer and network data. Because of the constantly changing nature of computer and network data sources and analysis tools, readers are expected to take advantage of other resources, including those listed in this guide, for more current and detailed information than that presented in this guide.

1.4 Document Structure

The remainder of this document is organized into the following seven major sections:

- Section 2 discusses the need for computer and network data analysis, and provides guidance on establishing data analysis capabilities for an organization.
- Section 3 explains the basic steps involved in performing the computer and network data analysis process: data acquisition, examination, utilization, and review.
- Sections 4 through 7 provide details on acquiring and examining data from various data sources, based on the framework in Section 3. The data source categories discussed in Sections 4 through 7 are data files, operating systems, network traffic, and applications, respectively.
- Section 8 presents case studies that illustrate how analysis can correlate events among several data sources.

The document also contains several appendices with supporting material, as follows:

- Appendix A presents the major recommendations made throughout the document.
- Appendix B presents scenarios where data analysis techniques may be useful and asks the reader a series of questions regarding each scenario.
- Appendices C and D contain a glossary and acronym list, respectively.
- Appendix E lists print resources, and Appendix F identifies online tools and resources, that may be useful references for establishing data analysis capabilities or understanding data analysis tools and techniques.

2. Establishing and Organizing a Data Analysis Capability

The term *data* refers to distinct pieces of digital information that have been formatted in a specific way. The expansion of computers[1] for professional and personal use and the pervasiveness of networking have fueled the need for tools that can analyze an ever-increasing amount of data from many sources. For example, data may be stored or transferred by standard computer systems (e.g., desktops, laptops, servers), networking equipment (e.g., firewalls, routers), personal digital assistants (PDA), CDs, DVDs, removable hard drives, backup tapes, flash memory, thumb drives, and jump drives. Many consumer electronics can also be used to store data; examples of such devices include cell phones, video game consoles, digital audio players, and digital video recorders. This increasing variety of data sources has helped to spur the development and refinement of data analysis tools and techniques. This has also been caused by the realization that such tools and techniques can be used for many purposes, such as reconstructing computer security incidents, troubleshooting operational problems, and recovering from accidental system damage.

This section discusses several aspects of organizing a data analysis capability for an organization. It begins by showing the wide variety of potential uses for data analysis, and then gives a high-level overview of the data analysis process. The next part of the section discusses how data analysis services are typically provided and provides guidance on building and maintaining the necessary skills to perform data analysis tasks. The section also explains the need to include various teams throughout the organization, such as legal advisors and physical security staff, in some data analysis activities. The section ends by discussing how policies and procedures should encompass data analysis, such as defining roles and responsibilities, providing guidance on the proper usage of data analysis tools and techniques, and incorporating data analysis into the information system life cycle.

The data analysis techniques and processes presented in this guide are based on principles of digital forensics. *Forensic science* is generally defined as the application of science to the law. Digital forensics, also known as computer and network forensics, has many definitions. Generally, *digital forensics* is considered to be the application of science to the identification, collection, analysis, and examination of digital evidence while preserving the integrity of the information and maintaining a strict chain of custody for the evidence. Computer and network data analysis is similar to digital forensics and uses many of the same techniques and tools, but data analysis does not necessarily include all of the actions necessary for preserving the integrity of all information collected, nor does it include keeping a chain of custody or other evidence preservation actions. **Accordingly, this publication should not be used as a guide for executing a digital forensic investigation, construed as legal advice, or used as the basis for performing investigations of criminal activity.**

2.1 The Need for Data Analysis

Computer and network data analysis tools and techniques are used for purposes such as the following:

- **Operational Troubleshooting.** Many data analysis tools and techniques can be applied to troubleshooting operational issues, such as finding the virtual and physical location of a host with an incorrect network configuration, resolving a functional problem with an application, and recording and reviewing the current operating system and application configuration settings for a host.

[1] In this publication, the term *computer* is used to refer to all computing, storage, and peripheral devices.

- **Log Monitoring.** Various tools and techniques can assist with log monitoring, such as analyzing log entries and correlating log entries across multiple systems. This can assist with incident handling, identifying policy violations, auditing, and other efforts.
- **Data Recovery.** There are dozens of tools that can recover lost data from systems. This includes data that has been accidentally or purposely deleted, overwritten, or otherwise modified. The amount of data that can be recovered varies on a case-by-case basis.
- **Data Acquisition.** Some organizations use tools to acquire data from hosts that are being redeployed or retired. For example, when a user leaves an organization, the data from the user's workstation can be acquired and stored in case the data is needed in the future. The workstation's media can then be sanitized to remove all of the original user's data.

Regardless of the situation, the data analysis process is composed of the following phases:

- **Acquisition.** The first phase is acquiring data from the possible sources of relevant data, following procedures that preserve the integrity of the data. For example, acquisition is usually performed in a timely manner because of the likelihood of losing data such as current network connections.
- **Examination.** Examinations involve using automated methods to sift through large amounts of acquired data and extract and identify data of particular interest.
- **Utilization.** The next phase is reporting the results of the examination, which may include the actions used in the examination and recommendations for improvement. The formality of the utilization step varies greatly depending on the situation.
- **Review.** Performing reviews of processes and practices within the context of the current task can identify policy shortcomings, procedural errors, and other problems that need to be addressed. Lessons learned during the review phase should be incorporated into future data analysis efforts.

Section 3 describes the data analysis process in depth, while Sections 4 through 7 provide additional information on acquiring and examining different types of computer and network data.

2.2 Staffing

Practically every organization needs to have some capability to perform computer and network data analysis. Although the extent of this need varies, the primary users of data analysis tools and techniques within an organization usually can be divided into the following two groups:

- **IT Professionals.** This group includes technical support staff and system, network, and security administrators. They each use a small number of data analysis techniques and tools specific to their area of expertise during their routine work (e.g., monitoring, troubleshooting, data recovery).
- **Incident Handlers.** They respond to a variety of computer security incidents, such as unauthorized data access, inappropriate system usage, malicious code infections, and denial of service attacks. Incident handlers typically use a wide variety of data analysis techniques and tools during their investigations.

Many organizations rely on a combination of their own staff and external parties to perform data analysis tasks. For example, some organizations perform standard data analysis tasks themselves and use outside parties only when specialized assistance is needed. Even organizations that want to perform all data analysis tasks themselves usually outsource the most demanding ones, such as sending physically damaged media to a data recovery firm for reconstruction. Such tasks typically require the use of specialized software, equipment, facilities, and technical expertise that most organizations cannot justify the high expense of acquiring and maintaining.

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

- **Cost.** There are many potential costs involved with data analysis. Software, hardware, and equipment used to acquire and examine data may carry significant costs (e.g., purchase price, software updates and upgrades, maintenance). Other significant costs involve staff training and labor costs. In general, data analysis actions that are needed rarely might be more cost-effective if performed by an external party, whereas actions that are needed frequently might be more cost-effective to perform internally.
- **Response Time.** Personnel located on-site may be able to initiate data analysis activity more quickly than off-site personnel. For organizations with geographically disparate physical locations, off-site outsourcers located near distant facilities might be able to respond more quickly than personnel located at the organization's headquarters.
- **Data Sensitivity.** Because of data sensitivity and privacy concerns, some organizations may be reluctant to allow external parties to image hard drives and perform other actions that provide access to data. For example, a system that contains traces of an incident might also contain health care information, financial records, or other sensitive data; an organization might prefer to keep that system under its own control to safeguard the privacy of the data. On the other hand, if there is a privacy concern within the team—for example, an incident is suspected to involve a member of the incident handling team—use an independent third party to perform data analysis actions.

Incident handlers performing data analysis tasks need to have solid knowledge of data analysis principles, procedures, tools, and techniques, as well as tools and techniques that could conceal or destroy data. It is also beneficial for incident handlers to have expertise in information security and specific technical subjects, such as the most commonly used operating systems, file systems, applications, and network protocols within the organization. Having this type of knowledge facilitates faster and more effective responses to incidents. Incident handlers also need a general, broad understanding of systems and networks so that they can determine quickly which teams and individuals are well-suited to providing technical expertise for particular data analysis efforts, such as acquiring data for an uncommon application.

Individuals performing data analysis may need to perform other types of tasks as well. For example, incident handlers may provide training courses on data analysis to technical support staff, system and network administrators, and other IT professionals. Examples of possible training topics include an overview of data analysis tools and techniques, advice on using a particular tool, and the signs of a new type of attack. Incident handlers may also want to have interactive sessions with groups of IT professionals to hear their thoughts on data analysis tools and identify potential shortcomings in existing data analysis capabilities.

On an incident handling team, more than one team member should be able to perform each typical data analysis activity, so that the absence of any single team member should not severely impact the team's abilities. Incident handlers may train each other on the use of data analysis tools and other technical and procedural topics. Also, hands-on exercises and external IT and data analysis training courses can be helpful in building and maintaining skills. It may also be beneficial to have the team members see demonstrations of new tools and technologies, or try out tools in a lab. This may be particularly helpful for familiarizing incident handlers with acquiring and examining data from devices such as cell phones and PDAs.

2.3 Interactions with Other Teams

It is not feasible for any individual to be well-versed in every technology (including all software) used within an organization, so individuals performing data analysis actions should be able to reach out to other teams and individuals within their organization as needed for additional assistance. For example, an incident involving a particular database server may be handled more efficiently if the database administrator is available to provide background information, answer technical questions, and provide database documentation and other reference material. Accordingly, organizations should ensure that IT professionals throughout the organization, especially incident handlers and other first responders to incidents, understand their roles and responsibilities for data analysis, receive training and education on data analysis-related policies 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 data analysis activities in a less technical capacity. Examples include management, legal advisors, human resources, auditors, and physical security staff. Management is responsible for supporting data analysis capabilities, reviewing and approving data analysis-related policy, and approving certain data analysis actions (e.g., taking a mission-critical system off-line for 12 hours to acquire its hard drives). Legal advisors should carefully review all data analysis policy and high-level procedures, and they can provide additional guidance when needed to ensure that data analysis actions are performed lawfully. Auditors can help determine the economic impact of an incident, including the cost of data analysis activity. Physical security staff can assist with gaining access to systems. The services that these teams provide can be beneficial.

To facilitate inter-team communications, each team should designate one or more points of contact. These individuals are responsible for knowing the expertise of each team member and directing inquiries for assistance to the appropriate person. Organizations should maintain a list of contact information that the appropriate teams can reference as needed. The list should include both standard (e.g., office phone) and emergency (e.g., cell phone) contact methods.

2.4 Policies

Organizations should ensure that their policies contain clear statements that address all major data analysis considerations, such as performing monitoring and conducting regular reviews of data analysis policies and procedures. 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 data analysis roles that provides more detailed rules for appropriate behavior. Such personnel should be familiar with and understand the data analysis policy. Policies may need to be updated frequently, particularly for organizations that span many jurisdictions, because of changes to laws and regulations. Of course, the organization's data analysis policy should also be consistent with the organization's other policies. Sections 2.4.1 through 2.4.3 discuss policy-related topics in more detail.

2.4.1 Defining Roles and Responsibilities

Data analysis policy should clearly define the roles and responsibilities of all people performing or assisting with the organization's data analysis activities. This should include actions performed during both incident handling and routine work activities (e.g., system administration, network troubleshooting). The policy should include all internal teams that may participate in data analysis efforts, such as those listed in Section 2.3, and external organizations such as outsourcers and incident response organizations.

The policy should clearly indicate who should contact which internal teams and external organizations under different circumstances.

2.4.2 Providing Guidance for Data Analysis Tool Use

Incident handlers, IT professionals such as system and network administrators, and others within an organization can use data analysis tools and techniques for a variety of reasons. Although the technologies have many benefits, they can also be misused accidentally or intentionally to provide unauthorized access to information, or to alter or destroy information. Also, the use of certain data analysis tools may not be warranted in some situations—for example, a minor incident probably does not merit hundreds of hours of data acquisition and examination efforts.

To ensure that tools are used reasonably and appropriately, the organization's policies and procedures should clearly explain what data analysis actions should and should not be performed under various circumstances. For example, a network administrator should be able to monitor network communications on a regular basis to solve operational problems, but not to read users' e-mail unless specifically authorized to do so. A help desk agent might be permitted to monitor network communications for a particular user's workstation to troubleshoot an application problem, but not permitted to perform any other network monitoring. Individual users might be forbidden from performing any network monitoring under any circumstances. Policies 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).

Policies and procedures should also address the use of anti-forensic tools and techniques. Described in Sections 4 through 7, anti-forensic software is designed to conceal or destroy data so that others cannot access it. There are many positive uses for anti-forensic software, such as removing data from computers that are to be donated to charity, and removing data cached by Web browsers to preserve a user's privacy. However, like data analysis tools, anti-forensic tools can be used for both benign and malicious reasons, so organizations should specify who is permitted to use them and under what circumstances.

Because data analysis tools may record sensitive information, policies 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 Data Analysis in the Information System Life Cycle

Many incidents can be handled more efficiently and effectively if data analysis 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 operating system and application deployments

- Maintaining records (e.g., baselines) of network and system configurations
- Establishing data retention policies that support historical reviews of system and network activity.

Most of these considerations are extensions of existing provisions in policies and procedures, so they are typically specified within the relevant individual documents instead of a centralized data analysis policy.

2.5 Procedures

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

The procedures should include general methodologies for investigating an incident using data analysis techniques, since it is not feasible to develop comprehensive procedures tailored to every possible situation. However, organizations should consider developing step-by-step procedures for performing routine tasks such as imaging a hard disk and capturing and recording volatile information from systems. The goal is for the procedures to facilitate consistent, effective, and accurate data analysis actions. Of course, the procedures should also be consistent with the organization's policies and all applicable laws. Accordingly, organizations should include technical experts and legal advisors in the development of procedures as a quality assurance measure. Management should also be involved in procedure development, particularly in ensuring that all major decision-making points are documented and the proper course of action is defined, so that decisions should be made consistently.

It is also important to maintain the procedures so that they are accurate. Management should determine how frequently the procedures should be reviewed (generally at least annually). Reviews should also be conducted when significant changes are made to the team's policies and procedures. When a procedure is updated, the previous version should be archived for possible future use in legal proceedings. Procedure reviews should include the same teams that participate in procedure creation. In addition, organizations might also choose to conduct exercises that help to validate the accuracy of certain procedures.

2.6 Recommendations

The key recommendations presented in this section for organizing a data analysis capability are summarized below.

- **Organizations should have some capability to perform computer and network data analysis.** Data analysis can assist with various tasks within an organization, including reconstructing computer security incidents, troubleshooting operational problems, and recovering from accidental system damage.
- **Organizations should determine which parties should handle each aspect of data analysis.** Most organizations rely on a combination of their own staff and external parties to perform data analysis 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 data analysis capabilities.** More than one team member should be able to perform each typical data analysis activity. Hands-on exercises and IT

and data analysis 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 data analysis.** Individuals performing data analysis actions should be able to reach out to other teams and individuals within an organization as needed for additional assistance. Examples of teams include IT professionals, management, legal advisors, auditors, and physical security staff. Members of these teams should understand their roles and responsibilities for data analysis, receive training and education on data analysis-related policies and procedures, and be prepared to cooperate with and assist others on data analysis actions.

+ **Data analysis 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 data analysis policy for incident handlers and others with data analysis roles that provides more detailed rules for appropriate behavior. Everyone who may be called upon to assist with any data analysis efforts should be familiar with and understand the data analysis policy. Additional policy considerations are as follows:

  – Data analysis policy should clearly define the roles and responsibilities of all people performing or assisting with the organization's data analysis activities. The policy should include all internal and external parties that may be involved, and it should clearly indicate who should contact which parties under different circumstances.

  – The organization's policies and procedures should clearly explain what data analysis actions should and should not be performed under normal and special circumstances, and also address the use of anti-forensic tools and techniques. Policies and procedures should also address the handling of inadvertent exposures of sensitive information.

  – Incorporating data analysis considerations in the information system life cycle can lead to more efficient and effective handling of many incidents.

+ **Organizations should create and maintain procedures for performing data analysis tasks.** The procedures should include general methodologies for investigating an incident using data analysis techniques, and possibly step-by-step procedures for performing routine tasks. The procedures should be reviewed regularly and maintained so that they are accurate.

This page has been left blank intentionally.

3. Performing the Data Analysis Process

The most common goal of performing computer and network data analysis is to gain a better understanding of an event of interest by finding and analyzing the facts involving that event. As described in Section 2.1, data analysis may be needed in many different situations, such as effective handling of malware incidents and unusual operational problems. Regardless of the need, data acquisition and examination should be performed using the same four-phase process shown in Figure 3-1. The exact details of these steps may vary based on the specific need.

This section describes the phases of the data analysis process: acquisition, examination, utilization, and review. During the acquisition phase, data related to a specific event is identified, collected, and protected. The second phase, examination, is when tools and techniques appropriate to the types of data collected during the first phase are executed to identify and analyze the relevant information from the acquired data. The second phase may use a combination of automated tools and manual processes. The next phase, utilization, is preparing and presenting the results of the examination processes in a format that can be easily assimilated by the prospective audiences. The results might need to be presented to incident handlers, IT professionals, end users, management, or others. In many cases, the utilization phase is trivial because the person performing the analysis is also the consumer of the results. The final phase involves reviewing processes and practices within the context of the current event to identify policy shortcomings, procedural errors, and other problems that need to be remedied. Lessons learned during the review phase should be incorporated into future data analysis efforts.

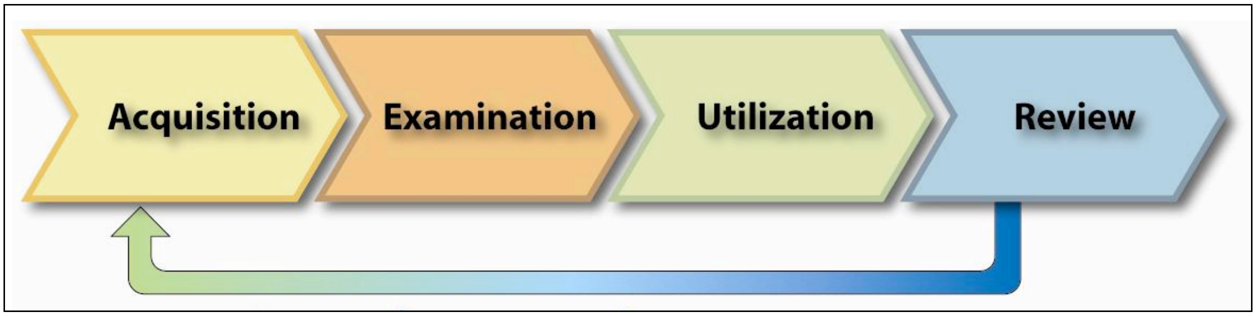


Figure 3-1. Computer and Network Data Analysis Process

3.1 Data Acquisition

The first step in the process is to identify potential sources of data and acquire the data. Section 3.1.1 illustrates the variety of data sources and discusses actions that organizations can take to support the ongoing collection of data. Section 3.1.2 describes the recommended steps for collecting data. Section 3.1.3 discusses incident response considerations, emphasizing the need to weigh the value of acquired data against the costs and impact to the organization of the acquisition process.

3.1.1 Possible Sources of Data

The increasingly widespread use of digital technology for both professional and personal reasons 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. Examples of external storage forms that might be sources of data are thumb drives, memory and flash cards, optical discs, and magnetic disks. Standard computer systems also contain volatile data that is available temporarily (i.e., until the system is shut down or rebooted). In addition to computer-related devices, many types of portable digital devices may also contain data. These devices include PDAs, cell phones, digital cameras, digital recorders, and audio players.

Data sources are often located elsewhere. For example, as described in Sections 6 and 7, there are usually many sources of information within an organization regarding network activity and application usage. Information may also be recorded by other organizations, such as logs of network activity for an Internet Service Provider (ISP). Analysts should also be mindful of the owner of each data source, and the effect that this may have on acquiring data. For example, getting copies of ISP records typically requires a court order. Analysts should also be aware of the organization's policies and legal considerations regarding externally owned property at the organization's facilities, such as an employee's personal laptop or contractor's laptop. The situation can become even more complicated if locations outside the organization's control are involved, such as an incident involving a computer at a telecommuter's home office. Sometimes it is simply not feasible to acquire a primary data source; analysts should be aware of alternate data sources that may contain some or all of the same data, and use those sources instead of the unattainable source.

Organizations can take ongoing proactive measures to collect useful data. For example, as described in Section 5.1.1, most operating systems 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.[2] 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 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. Also, 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.

Another proactive data collecting measure is keystroke monitoring, which records the keyboard usage of a particular system. Although it can provide a valuable record of activity, it can also be a violation of privacy unless users are advised that such monitoring may be performed through organizational policy and login banners. Most organizations do not employ keystroke monitoring except when gathering additional information on a suspected incident. Authority for performing such monitoring should be discussed with legal advisors and documented clearly in the organization's policy.

3.1.2 Collecting 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. Although the following items provide an overview of the three steps, the specific details behind steps 2 and 3 vary based upon the type of data being acquired. Sections 4.2, 5.2, 6.3, and 7.3 provide more detailed explanations of acquiring and

[2] If auditing was not enabled on a system when an event occurred, incident handlers might enable auditing after the event is discovered in an attempt to record evidence of ongoing activity. Because this could alter evidence of the incident and alert an attacker to the presence of the incident handlers, the impact of enabling auditing should be considered, and incident handlers should document their actions.

verifying the integrity of data files, operating system data, network traffic data, and application data, respectively.

1. **Develop a plan to acquire the data.** Developing a plan is an important first step in most cases because there are multiple potential data sources. 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.** Based on the analyst's understanding of the situation and previous experience in similar situations, the analyst should be able to estimate the relative likely value of each potential data source.

   + **Volatility.** *Volatile data* refers to data on a live system that is lost after a computer is powered down. Volatile data may also be lost due to other actions performed on the system. In many cases, acquiring volatile data should be given priority over non-volatile data. However, non-volatile data may also be rather dynamic in nature, such as log files that are overwritten as new events occur.

   + **Amount of Effort Required.** The amount of effort required to acquire different data sources may vary widely. The effort involves not only the time spent by analysts and others within the organization (including legal advisors), but also includes the cost of equipment and services (e.g., outside experts). For example, acquiring data from a network router would probably require much less effort than acquiring data from an ISP.

   By considering these three factors for each potential data source, analysts can make informed decisions regarding the prioritization of data source acquisition, as well as determining which data sources to acquire. In some cases, there are so many possible data sources that it is not practical to acquire them all. Organizations should carefully consider the complexities of prioritizing data source acquisition and develop written plans, guidelines, and procedures that can help analysts in performing prioritization effectively.

2. **Acquire the data.** If the data has not already been acquired by security tools, analysis tools, or other means, the general process for acquiring data involves using a trusted toolkit to collect volatile data and duplicating non-volatile data sources to collect their 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.** After the data has been acquired, its integrity should be verified. Data integrity verification typically consists of using tools to compute the message digest of the original and copied data, and comparing them to make sure that they are the same.

Before the analyst begins to acquire 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 acquire and preserve evidence in a way that supports its use in future legal or internal disciplinary proceedings. In such situations, computer and network forensic procedures should be followed instead of the less formal data analysis procedures discussed in this guide.

To assist the analyst with acquisition, the necessary resources should be prepared beforehand, such as analyst workstations, backup devices, and blank media.

3.1.3 Incident Response Considerations

When performing data analysis during incident response, an important consideration is how and when the incident should be contained. Isolating the pertinent systems from external influences may be necessary to maintain data integrity. In many cases, the analyst should work with the incident response team to make a containment decision (e.g., disconnecting network cables, unplugging power, increasing physical security measures, gracefully shutting down a host). 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 the risk while maintaining data integrity 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. For example, taking a critical system offline for several hours to acquire disk images and other data may adversely affect the ability of the organization to perform its necessary operations. Significant downtime may result in substantial monetary losses to the organization. Therefore, care should be taken to minimize disruptions to an organization's operations.

One step often taken to contain an incident is to secure the perimeter around a computer and limit access to authorized personnel. Also, it is sometimes helpful to create a list of all users who have access to the computer, because they may be able to provide passwords or information on where specific data is located. If the computer is connected to a network, disconnecting network cables attached to the computer can prevent remote users from modifying the computer's data. If the computer uses a wireless network connection, unplugging the external network adapter from the computer or disabling the internal network adapter may be used to sever the network connection. If neither option is possible, then powering off the wireless network access point that the computer is connected to should achieve the same result. However, doing so may prevent users outside the scope of the investigation from performing their daily routines. Also, there could be more than one access point within range of the computer. Some wireless network adapters automatically attempt to connect to other access points when the primary access point is unavailable, so containing the incident in this way could involve disconnecting several access points.

3.2 Examination

After data has been acquired, the next phase is to examine the data, which is identifying, collecting, and organizing the relevant pieces of information from the acquired data. This phase may also involve bypassing or mitigating operating system or application features that obscure data and code, such as data compression, encryption, and access control mechanisms. For example, 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. Additionally, data files of interest may contain extraneous information that should be filtered. For example, yesterday's firewall log may hold millions of records, but only five of the records are related to a particular event of interest.

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 on known files, which can also be used to include or exclude files from further consideration. Specific information on examination tools and techniques is presented in Sections 4.3, 5.3, 6.4, and 7.4.

Once the relevant information has been extracted, the analyst should study the data to draw conclusions from the data. Analysts should follow a methodical approach to draw conclusions based on the available data, or determine that no conclusion can yet be drawn. The analysis should include identifying people, places, items, and events, and determining how they are related so that a conclusion can be reached. Often times, this will include correlating data among multiple sources. For instance, a network IDS log may link an event to a host, the host audit logs may link the event to a specific user account, and the host IDS log may indicate what actions that user performed. Tools such as centralized logging and security event management software can facilitate this process by automatically gathering and correlating the data. Also, comparing system characteristics to known baselines can identify various types of changes made to the system. Section 8 describes the analysis process in more detail.

3.3 Utilization

Data *utilization* is the process of preparing and presenting information that resulted from the examination phase. Many factors affect data utilization, including the following:

+ **Data Reduction.** Reducing the data to present only the necessary facts to the proper people helps to ensure an overall understanding of what has occurred and possibly indicate what needs to be done to correct or modify an issue. If an analyst is identifying which files were e-mailed by a virus to computers that later became infected, the audience wanting to see the names of the files in question will probably not be interested in a display including other e-mails sent to the same accounts.

+ **Alternative Explanations.** When the information regarding an event is incomplete, it may not be possible to identify a definitive explanation as to what happened. When an event has two or more plausible explanations, each should be given due consideration in the data utilization process.

+ **Audience Consideration.** Knowing the audience to which the data or information garnered will be shown is important. A system administrator might want to see network traffic and related statistics in great detail. Senior management might simply want a high-level overview of what happened, such as a simplified visual representation of how the attack occurred, and what needs to be done to prevent similar incidents.

+ **Actionable Information.** Utilization also includes actionable information gained from data that may allow an analyst to acquire new sources of information. For example, a list of contacts may be garnered from the data that may lead to additional information about an incident. Also, information may be obtained that could prevent future events, such as a backdoor on a system that could be used for future attacks, or a worm scheduled to start spreading at a certain time.

3.4 Review

Analysts should continuously review their processes and practices within the context of current tasks to help identify policy shortcomings, procedural errors, and other issues that may need to be remedied. Periodic refreshing of skills through coursework, on-the-job experience, and academic sources helps ensure that people performing data analysis keep pace with rapidly changing technologies and job responsibilities. Periodic review of policies and procedures also helps ensure the organization stays current with trends in technology and changes in law.

Many incident response teams hold formal reviews after each major event. Such reviews tend to include serious consideration of any possible improvements to processes and procedures, and typically at least some minor changes are approved and implemented after each review. For example, many organizations find it resource-intensive to maintain current lists of personnel to contact regarding each different type of incident that may occur. Once changes to processes and procedures are implemented, all team members should be informed of the changes and frequently reminded of the proper procedures to follow. Teams typically have formal mechanisms for tracking changes and identifying the current versions of each process and procedure document. In addition, many teams have posters or other highly visible documents mounted on walls or doors that remind teams of the key steps to take, so that everyone is constantly reminded of how things are supposed to be done.

3.5 Recommendations

The key recommendations presented in this section for the data analysis process are summarized below.

- **Organizations should perform data analysis using a consistent process.** The data analysis process presented in this guide uses a four-phase process: acquisition, examination, utilization, and review. The exact details of the phases may vary based on the need for data analysis.

- **Analysts should be aware of the range of possible data sources.** Analysts should be able to survey a physical area and recognize the 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 acquire data from a primary source.

- **Organizations should be proactive in collecting useful data.** Configuring auditing on operating systems, implementing centralized logging, performing regular system backups, and using security monitoring controls can all generate sources of data for future data analysis efforts.

- **Analysts should perform data acquisition using a standard process.** The recommended steps are developing a plan to acquire the data, acquiring the data, and verifying the integrity of the data. Analysts should create a plan that prioritizes 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.

- **Analysts should use a methodical approach.** The foundation of computer and network data analysis is using a methodical approach to draw conclusions based on the available data, or determine that no conclusion can yet be drawn.

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

4. Using Data from Data Files

A *data file* (also simply 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 examination of computer media is dependent on being able to acquire, extract, and analyze the files that reside on the media. This section begins with an overview of the most common media types and *filesystems*—methods for naming, storing, organizing, and accessing files. It then discusses how files should be acquired and how the integrity of the files should be preserved. This section also discusses various technical issues related to file recovery, such as recovering data from deleted files. The last portion of this section describes the extraction and analysis of files, providing guidance on tools and techniques that can assist analysts.

4.1 File Basics

Before attempting to acquire or examine files, analysts should have at least a basic understanding of files and filesystems. Section 4.1.2 explains how filesystems are used to organize files, and provides an overview of several common filesystems. Section 4.1.3 discusses how data from deleted files may still exist within filesystems. Analysts should also be aware of the variety of media that may contain files; Section 4.1.1 provides several examples of media primarily used in personal computers, and several more examples of media that are commonly used in other types of digital devices.

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. In addition to traditional media types such as hard drives and floppy disks, files are often stored on consumer devices such as PDAs and cell phones, as well as newer media types, such as flash memory cards made popular by digital cameras. Table 4-1 lists media types that are commonly used currently on computers and digital devices. This list does not include every media type available; rather, it is intended to show the variety of media types that an analyst may come across.

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 rewriteable (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 rewriteable (DVD±RW) single and dual layer disks
Hard drive	N/A	20 GB – 300 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 due to 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
ATA flash card	PCMCIA slot	8 MB – 2 GB	PCMCIA flash memory card; measures 85.6 x 54 x 5 mm
Used by Many Types of Digital Devices			
Flash/Jump drive	USB interface	16 MB – 2 GB	Also known as thumb drive because of their size
CompactFlash 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
MultiMediaCard (MMC)	PCMCIA adapter or memory card reader	16 MB – 512 MB	Measure 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
SmartMedia Card	PCMCIA adapter or memory card reader	8 MB – 128 MB	Measure 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; measure 20 x 25 x 1.7 mm

4.1.2 Filesystems

Before media can be used to store files, usually the media must 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 have been formatted with a filesystem. Some media types, such as floppy disks, can contain at most one partition (and consequently, one logical volume). The format of the logical volumes is determined by the selected 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. However, all filesystems share some common traits. First, they use the concepts of directories and files to organize and store data. *Directories* are organizational structures that are used to group files together. In addition to files, directories may contain other directories called *subdirectories*. Second, filesystems use some data structure to point to the location of files on media. Also, they store each data file written to media in one or more *file allocation units*. These are referred to as *clusters* by some filesystems (e.g., File Allocation Table [FAT], NT File System [NTFS]) and *blocks* by other filesystems (e.g., Unix and Linux filesystems). A file allocation unit is simply a group of *sectors*, which are the smallest units that can be accessed on a media.

The following items describe some commonly used filesystems:

+ **FAT12**.[3] FAT12 is used only on floppy disks and FAT volumes smaller than 16 MB. FAT12 uses a 12-bit file allocation table entry to address an entry in the filesystem.

+ **FAT16**. MS-DOS, Windows 95/98/NT/2000/XP, Windows Server 2003, and some UNIX operating systems support FAT16 natively. FAT16 is also commonly used for multimedia devices such as digital cameras and audio players. FAT16 uses a 16-bit file allocation table entry to address an entry in the filesystem. FAT16 volumes are limited to a maximum size of 2 GB in MS-DOS and Windows 95/98. Windows NT and newer operating systems increase the maximum volume size for FAT16 to 4 GB.

+ **FAT32**.[4] Windows 95 OEM Service Release 2 (OSR2), Windows 98/2000/XP, and Windows Server 2003 support FAT32 natively, as do some multimedia devices. FAT32 uses a 32-bit file allocation table entry to address an entry in the filesystem. The maximum FAT32 volume size is 2 terabytes (TB).

+ **NTFS**. Windows NT/2000/XP and Windows Server 2003 support NTFS natively. NTFS is a *recoverable filesystem*, which means that it can automatically restore the consistency of the filesystem when errors occur. In addition, NTFS supports data compression and encryption, and allows user and group-level access permissions to be defined for data files and directories.[5] The maximum NTFS volume size is 2 TB.

+ **High-Performance File System (HPFS)**. HPFS is supported natively by OS/2 and can be read by Windows NT 3.1, 3.5, and 3.51. HPFS builds upon the directory organization of FAT by providing automatic sorting of directories. In addition, HPFS reduces the amount of lost disk space by utilizing smaller units of allocation. The maximum HPFS volume size is 64 GB.

[3] More information on FAT12 and FAT16 is available at http://www.microsoft.com/resources/documentation/windows/xp/all/reskit/en-us/prkc_fil_yksz.asp.

[4] The FAT32 filesystem specification, which provides highly technical details on FAT32, is available for download from http://www.microsoft.com/whdc/system/platform/firmware/fatgen.mspx.

[5] Additional NTFS features are described at http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/prkc_fil_gywp.asp.

- **Second Extended Filesystem (ext2fs).**[6] ext2fs is supported natively by Linux. ext2fs supports standard Unix file types and filesystem checks to ensure filesystem consistency. The maximum ext2fs volume size is 4 TB.
- **Third Extended Filesystem (ext3fs)**. ext3fs is supported natively by Linux. ext3fs is based on the ext2fs filesystem and provides journaling capabilities that allow consistency checks of the filesystem to be performed quickly on large amounts of data. The maximum ext3fs volume size is 4 TB.
- **Hierarchical File System (HFS)**.[7] HFS is supported natively by Mac OS. HFS is mainly used in older versions of Mac OS but is still supported in newer versions. The maximum HFS volume size under Mac OS 6 and 7 is 2 GB. The maximum HFS volume size in Mac OS 7.5 is 4 GB. Mac OS 7.5.2 and newer Mac operating systems increase the maximum HFS volume size to 2 TB.
- **HFS Plus**.[8] HFS Plus is supported natively by Mac OS 8.1 and later, and is a journaling filesystem under Mac OS X. HFS Plus is the successor to HFS and provides numerous enhancements such as long filename support and Unicode filename support for international filenames. The maximum HFS Plus volume size is 2 TB.
- **Unix File System (UFS).**[9] UFS is supported natively by several types of Unix operating systems, including Solaris, FreeBSD, OpenBSD, and Mac OS X. However, most operating systems have added proprietary features, so the details of UFS differ among implementations.
- **Compact Disk File System (CDFS)**. As the name indicates, the CDFS filesystem is used for CDs.
- **International Organization for Standardization (ISO) 9660**. The ISO 9660 filesystem is commonly used on CD-ROMs. Another popular CD-ROM filesystem is Joliet, a variant of ISO 9660. ISO 9660 supports filename lengths of up to 32 characters, while Joliet supports up to 64 characters. Joliet also supports Unicode characters within filenames.
- **Universal Disk Format (UDF)**. UDF is the filesystem used for DVDs, and is also used for some CDs.

4.1.3 Other Data on Media

As described in Section 4.1.2, filesystems are designed to store files on media. However, filesystems may also hold data from deleted files or earlier versions of existing files. This data may provide important information. (Section 4.2 discusses techniques for acquiring the data.) The following items describe how this data may still exist 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 operating system. However, the operating system considers this to be free space and could overwrite any portion of the deleted file at any time.

[6] More information on ext2fs is available at http://e2fsprogs.sourceforge.net/ext2.html.

[7] An overview of HFS is available at http://developer.apple.com/documentation/mac/Files/Files-17.html.

[8] An overview of HFS Plus and technical details on its implementation are available at http://developer.apple.com/technotes/tn/tn1150.html.

[9] An overview of UFS is available at http://en.wikipedia.org/wiki/UFS.

- **Slack Space**. As noted before, filesystems 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 KB and a file is only 7 KB, the entire 32 KB is still allocated for 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 and includes unallocated clusters or blocks. This often includes space on the media where files (and even entire volumes) may have resided at one point but have since been deleted. The free space may still contain pieces of data.

Another way that data might be hidden is with Alternate Data Streams (ADS) within NTFS volumes. NTFS has long supported multiple data streams for files and directories. Each file on an NTFS volume consists of an unnamed stream that is used to store the file's primary data, and optionally one or more named streams (i.e., file.txt:Stream1, file.txt:Stream2) that can be used to store auxiliary information such as file properties and picture thumbnail data.[10] For instance, if a user right-clicks on a file in Windows Explorer, views the file's properties, and then modifies the information displayed in the summary tab, the OS stores the summary information for the file in a named stream.

All data streams within a file share the file's attributes (e.g., timestamps, security attributes). Although named streams do affect the storage quota of a file, they are largely concealed from users because standard Windows file utilities such as Explorer only report the size of a file's unnamed stream. Therefore, a user cannot readily determine if a file contains ADS using the standard Windows file utilities. This allows hidden data to be contained within any NTFS file system. Moving files with ADS to non-NTFS filesystems effectively strips ADS from the file, so ADS can be lost if analysts are not aware of their presence. Software and processes are available to identify ADS.[11]

4.2 Acquiring Files

During data acquisition, the analyst should make one or more copies of the desired files or filesystems. The analyst can then work with a copy of the files without affecting the originals. Section 4.2.1 describes the primary techniques and tools for copying files and residual file data from media. Section 4.2.2 discusses the importance of maintaining the integrity of the files and provides guidance on hardware and software that can assist with preserving and verifying file integrity. It is often important to acquire not only the files, but also significant timestamps for the files, such as when the files were last modified or accessed; Section 4.2.3 describes the timestamps and explains how they can be preserved. Other technical issues related to file acquisition, such as finding hidden files and copying files from Redundant Arrays of Inexpensive Disks (RAID) implementations, are addressed in Section 4.2.4.

4.2.1 Copying Files from Media

Files can be copied from media using two different techniques, as follows:

- **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.

[10] Directories do not have an unnamed stream but may contain named streams.

[11] Additional information on ADS is available at http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prkc_fil_xurt.asp, http://www.infosecwriters.com/texts.php?op=display&id=53, and within http://www.heysoft.de/Frames/f_faq_ads_en.htm.

+ **Physical Backup.** Also known as *disk imaging*, a *physical backup* generates a bit-for-bit copy of the original media, including free space and slack space.[12] Physical backups require more storage space and take longer to perform than logical backups.

When a physical backup is executed, either a disk-to-disk or a disk-to-file copy can be performed. A *disk-to-disk copy*, as its name suggests, copies the contents of the media directly to another media. A *disk-to-file copy* copies the contents of the media to a single logical data file. A disk-to-disk copy is useful since the copied media can be connected directly to a computer and its contents readily viewed. However, a disk-to-disk copy requires a second media similar to the original media.[13] A disk-to-file copy allows the data file image to be moved and backed up easily. However, to view the logical contents of an image file, the analyst has to restore the image to media or open it in an application capable of displaying the logical contents of data file images. Section 4.3 discusses this in more detail.

Numerous hardware and software tools can perform physical and logical backups. Hardware tools are generally portable, provide bit-by-bit images, connect directly to the drive or computer to be imaged, and have built-in hash functions.[14] 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.[15] Some software solutions create logical copies of files or partitions and may ignore free or unallocated drive space, while others create a bit-by-bit image copy of the media. The type of data that is required may determine what software or hardware device is used for imaging data. For example, if only one folder on a particular partition is required, the analyst could use a simple software solution instead of a hardware-based imaging device.

Due to the increasing number of disk imaging tools available 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.[16] Some disk imaging tools can also perform recordkeeping, such as automated audit trails. The use of such tools can support consistency in the examination process and the accuracy and reproducibility of results.

Generally, tools that perform physical backups 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. 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.[17] For example, it is not necessary to take down a critical server used by hundreds of people just to acquire files from a single user's home directory. For logical backups of live systems, analysts can use standard

[12] A physical backup is also known as a *bitstream image*.

[13] The destination medium may need to be wiped before the copy occurs, so that any existing data on the medium is eliminated.

[14] Examples of hardware-based disk imaging tools are Image MASSter's SOLO Forensics (http://www.ics-iq.com/) and Logicube's Solitaire (http://www.logicube.com/). Additional products are referenced in Web sites listed in Appendix F, including The Ultimate Collection of Forensics Software (TUCOFS) (http://www.tucofs.com/tucofs/tucofs.asp?mode=filelist&catid=10&oskey=12).

[15] Examples of software-based disk imaging tools are Linux dd, SafeBack (http://www.forensics-intl.com/safeback.html), EnCase (http://www.encase.com/), Norton Ghost (http://www.symantec.com/sabu/ghost/ghost_personal/), and ILook (http://www.ilook-forensics.org/). Additional products are referenced in Web sites listed in Appendix F.

[16] The test results can be found at http://www.cftt.nist.gov/disk_imaging.htm.

[17] Analysts should also consider the possible need to acquire volatile data from the system. If the system is live, its volatile data is likely to change more quickly and be more challenging to preserve.

system backup software. However, performing a backup could impact the performance of the system and consume significant amounts of network bandwidth, depending on whether the backup is performed locally or remotely.

Organizations should have policy and procedures that indicate the circumstances under which physical and logical backups (including those from live systems) may be performed and which personnel may perform them.[18] It is typically most effective to establish policy and procedures based on categories of systems (i.e., low, medium, or high impact) and the nature of the event of interest; some organizations might also choose to create separate policy statements and procedures for particularly important systems. The policy or procedures should identify the individuals or groups with authority to make decisions regarding backups; these people should be capable of weighing the risks and making sound decisions. The policy or procedures should also identify which individuals or groups have the authority to perform the backup for each type of system; access to some systems might be restricted because of the sensitivity of the operations or data in the system.

4.2.2 Data File Integrity

During backups, the integrity of the original media should be maintained. To ensure that the backup process does not alter data on the original media, analysts can use a write-blocker while backing up the media. A *write-blocker* is a hardware or software-based tool that prevents a computer from writing to computer storage media connected to it. Hardware write-blockers are physically connected to the computer and the storage media being processed to prevent any writes to that media.[19] Software write-blockers are installed on the analyzing system and currently are available only for MS-DOS and Windows systems. (Some operating systems (e.g., Mac OS X) may not require software write-blockers because they can be set to boot with secondary devices not mounted. However, attaching a hardware write-blocking device will ensure integrity is maintained.) MS-DOS-based software write-blockers work by trapping Interrupt 13 and extended Interrupt 13 disk writes. Windows-based software write-blockers use filters to sort interrupts sent to devices to prevent any writes to storage media.[20]

In general, when using a hardware write-blocker, the media or device used to read the media should be connected directly to the write-blocker, and the write-blocker should be connected to the computer or device used to perform the backup. When using a software write-blocker, the software should be loaded onto a computer before the media or device used to read the media is connected to the computer. Write-blockers may also allow write-blocking to be toggled on or off for a particular device. It is important that when write-blocking is used, that it be toggled on for all connected devices.[21] Write-blockers should be tested routinely to ensure they support newer devices. For example, a new device might make use of reserved or previously unused functions or placeholders to implement device-specific functions that may ultimately write to a device and alter its contents.

[18] The intention of this is not to restrict users from performing backups of their own data and local workstations, but to prevent people from acquiring backups of others' systems and data without appropriate cause for doing so.

[19] Examples of hardware write-blockers are FastBloc (http://www.guidancesoftware.com/lawenforcement/ef_index.asp), NoWrite (http://www.mykeytech.com/nowrite.html), and SCSIBlock (http://www.digitalintelligence.com/products/scsiblock/). Additional tools are referenced in the Web sites listed in Appendix F.

[20] Examples of software write-blockers are PDBlock (http://www.digitalintelligence.com/software/disoftware/pdblock/) and WriteBlocker XP (https://www.acesle.com/). Additional tools are referenced in the Web sites listed in Appendix F.

[21] These are only general guidelines for using write-blockers. Analysts should refer to the operating procedures for a specific write-blocker product for instructions on proper usage.

After a backup is performed, it is important to verify that the copied data is an exact duplicate of the original data.[22] Computing the message digest of the copied data can be used to verify and ensure data integrity. A *message digest* is a digital signature that uniquely identifies data and has the property that changing a single bit in the data will cause a completely different message digest to be generated. There are many algorithms for computing the message digest of data, but the two most commonly used are MD5 and Secure Hash Algorithm 1 (SHA-1). These algorithms take as input data of arbitrary length and produce as output 128-bit message digests. Because SHA-1 is a FIPS-approved algorithm and MD5 is not, Federal agencies should use SHA-1 instead of MD5 for message digests.[23]

When a physical backup is performed, the message digest of the original media should be computed and recorded before the backup is performed. After the backup is performed, the message digest of the copied media should be computed and compared with the original message digest to verify that data integrity has been preserved. Additionally, the message digest of the original media should be computed again to verify that the backup process did not alter the original media, and all results should be documented. The process should be used for logical backups, except that message digests should be computed and compared for each data file.

4.2.3 File Modification, Access, and Creation Times

It is often important to know when a file was used or manipulated, and most operating systems 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.** This is the last time a file was changed in any way. This includes when a file is written to and when it is changed by another program.

+ **Access Time.** This is the last time any access was performed on a file (e.g., viewed, opened, printed).

+ **Creation Time.** This is generally the time and date the file was created. However, when a file is copied to a system, the creation time will become the time the file was copied to the new system. The modification time will remain intact.

Each type of filesystem may store different types of times. For example, Windows systems retain the last modified time, the last access time, and the creation time of files. UNIX systems retain the last modification, last inode[24] change, and last access times. However, some UNIX systems (including versions of BSD and SunOS) do not update the last access time of executable files when they are run. Some UNIX systems record the time when the metadata for a file was most recently altered. Metadata is data that provides information about a file's contents.

If an analyst wants to establish an accurate timeline of events, then the file times should be preserved to obtain reliable results. Analysts should be aware that not all methods for acquiring data files can preserve file times. For instance, performing a logical backup may cause file creation times to be altered when a

[22] If a backup is performed on a live system, it is likely that some files will change between the time the backup is initiated and it is completed and verified.

[23] Federal agencies must use FIPS-approved encryption algorithms contained in validated cryptographic modules. The Cryptographic Module Validation Program (CMVP) at NIST coordinates FIPS testing; the CMVP Web site is located at http://csrc.nist.gov/cryptval/. FIPS 180-2, *Secure Hash Standard*, is available at http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf. NIST has announced that Federal agencies should plan on transitioning from SHA-1 to stronger forms of SHA (e.g., SHA-224, SHA-256) by 2010. For more information, see NIST comments from August 2004 posted at http://csrc.nist.gov/hash_standards_comments.pdf, as well as http://www.nsrl.nist.gov/collision.html.

[24] An *inode* is a set of data regarding certain characteristics of a file, such as the privileges set for the file and the file's owner.

data file is copied, but physical backups can preserve file times because a bit-for-bit copy is generated. Therefore, whenever file times are essential, a physical backup should be used to acquire data.

Analysts should be aware that file times may not be accurate for various reasons, including the following:

- The computer's clock does not have the correct time. For example, the clock may not be synced regularly with an authoritative time source.
- 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

There are several technical issues that may arise in acquiring data files. As described in Section 4.2.1, the primary issue is acquiring deleted files and remnants of files existing in free and slack space on media. Individuals can use a variety of techniques to hinder the acquisition of such data. For example, there are many utilities available that perform *wiping*—overwriting media (or portions of the media, such as particular files) with random or constant values (e.g., all 0's). Such utilities vary in services and reliability, but most are effective in preventing easy acquisition of files, especially if several wipes are performed. Individuals can also use physical means to prevent data acquisition, such as demagnetizing a hard drive (also known as *degaussing*) or physically damaging or destroying media. Both physical and software-based techniques can make it very difficult, or even impossible, to recover all of the data using software. Recovery attempts in these cases necessitate the use of highly specialized forensic experts with advanced facilities, hardware, and techniques, but the cost and effort in doing so is prohibitive for general use.[25] In some cases, the data is simply not recoverable.

Another common issue is acquiring hidden data. Many operating systems permit users to tag certain files, directories, or even partitions as hidden, which means that by default they are not displayed in directory listings.[26] Some applications and operating systems hide configuration files to reduce the chance that users will accidentally modify or delete them. Also, on some operating systems, 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 operating system and many data files.[27] Users may create hidden partitions by altering the partition table to disrupt disk management and prevent applications from seeing that the data area exists. Hidden data could also be found within ADSs on NTFS volumes and the end-of-file slack space and free space on a medium. Many acquisition tools can recognize some or all of these methods of hiding data and recover the associated data.

Yet another issue that may arise is acquiring data from RAID arrays that use striping (e.g., RAID-0, RAID-5).[28] 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. This can be problematic because all partitions of a striped volume must be present to examine its contents, but the partitions reside on separate physical disk drives. Therefore, to examine a striped volume, each disk drive in the RAID array needs to be imaged and the RAID

[25] Companies that specialize in such recovery efforts include Data Recovery Services, DriveSavers, and Ontrack Data Recovery.

[26] On UNIX systems, files or folders beginning with a '.' are considered hidden and are not displayed when listing files unless the –a flag is used.

[27] An example of a freely available tool that can be used to locate hidden partitions is the FDISK utility built into DOS. Information on additional tools is available from the Web sites listed in Appendix F.

[28] An overview of RAID is available at http://www.adaptec.com/worldwide/product/markeditorial.html?prodkey=quick_explanation_of_raid.

configuration has to be recreated on the examination system.[29] Some tools can acquire striped volumes and are also able to preserve unused data areas of the volume, such as free space and slack space.[30]

4.3 Examining Data Files

After a logical or physical backup has been performed, the backup may have to be restored to another media before the data can be examined. This is dependent on the tools that will be used to perform the analysis. Some tools can analyze data directly from an image file, while others require that the backup image be restored to a medium first.[31] Regardless of whether a backup image file or restored image is used in the examination, it should only be accessed as read-only to ensure that the data being examined is not modified and that the data will provide consistent results on successive runs. As noted in Section 4.2.2, write-blockers can be used during this process to prevent writes from occurring to the restored image. After restoring the backup (if needed), the analyst begins to examine the acquired data and performs an assessment of the relevant files and data by locating all files, including deleted files, remnants of files in slack and free space, and hidden files. Next, the analyst may need to access the data within the files, which may be complicated through such measures as encryption and passwords. This section describes these processes as well as techniques that can expedite the examination of data and files.

4.3.1 Locating the Files

The first step in the examination is to locate the files. A disk image may capture many gigabytes of slack space and free space, which could contain thousands of files and file fragments. Manually extracting data from unused space can be a time-consuming and difficult process, as it requires knowledge of the underlying filesystem format. Fortunately, several tools are available that can automate the process of extracting unused space and saving it to data files, as well as recovering deleted files and files within a recycling bin. Analysts can also display the contents of slack space with hex editors or special slack recovery tools.

4.3.2 Accessing the Data

The next step in the examination process is to gain access to the data within the files. To make sense of the contents of a file, an analyst needs to know what type of data the file contains. The intended purpose of file extensions is to denote the nature of the file's contents; for example, a jpg extension indicates a graphic file, and an mp3 extension indicates a music file. However, users can assign any file extension to any type of file, such as naming a text file **mysong.mp3** or omitting a file extension. Also, some file extensions might be hidden or unsupported on other operating systems. Therefore, 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. As shown in Figure 4-1, the file header contains a file signature that identifies the type of data that particular file contains.[32] The header is indicative of the file contents regardless of the file extension. The example in Figure 4-1 has a file header of FF D8, which indicates that this is a JPEG file. A file header could be located in a file separate from the actual file data. Another

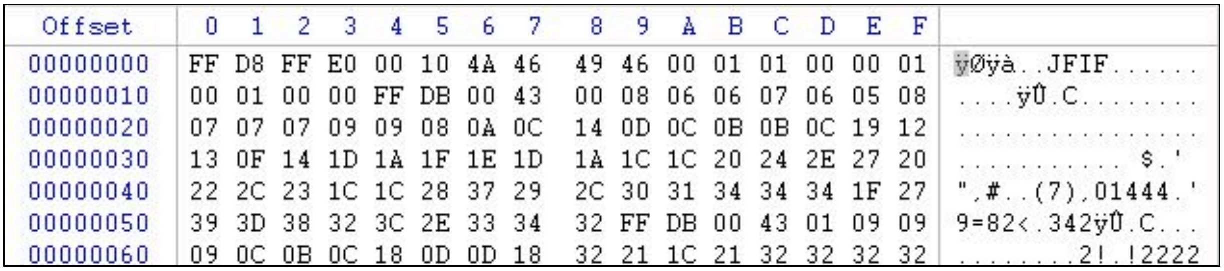
[29] Because RAID-5 stores parity information on one disk, it is possible to examine a RAID-5 volume by imaging all of the disks but one.

[30] For more information regarding RAIDs, visit http://www.anandtech.com/storage/showdoc.html?i=1491.

[31] At one time, when tools had more limited capabilities, it was often recommended to restore data files to a system using the same OS or a related one. Current tools are more advanced and are designed to be used with many types of data files, regardless of the underlying OS, so it is no longer necessary in most cases to restore data files to a particular OS.

[32] File headers can also indicate whether a file is encrypted.

effective technique for identifying the type of data in a file is a simple histogram showing the distribution of ASCII values as a percentage of total characters in a file. For example, a spike in the ‘space’, ‘a’, and ‘e’ lines generally indicates a text file, while consistency across the histogram indicates a compressed file. Other patterns are indicative of files that are encrypted or that were modified through steganography.



```
 Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  FF D8 FF E0 00 10 4A 46  49 46 00 01 01 00 00 01  ÿØÿà..JFIF......
00000010  00 01 00 00 FF DB 00 43  00 08 06 06 07 06 05 08  ....ÿÛ.C........
00000020  07 07 07 09 09 08 0A 0C  14 0D 0C 0B 0B 0C 19 12  ................
00000030  13 0F 14 1D 1A 1F 1E 1D  1A 1C 1C 20 24 2E 27 20  ........... $.' 
00000040  22 2C 23 1C 1C 28 37 29  2C 30 31 34 34 34 1F 27  ",#..(7),01444.'
00000050  39 3D 38 32 3C 2E 33 34  32 FF DB 00 43 01 09 09  9=82<.342ÿÛ.C...
00000060  09 0C 0B 0C 18 0D 0D 18  32 21 1C 21 32 32 32 32  ........2!.!2222
```

Figure 4-1. File Header Information

Encryption often presents challenges for analysts. Users might encrypt individual files, folders, volumes, or partitions so that others cannot access their contents without a decryption key or passphrase.[33] The encryption may be performed by the operating system or a third-party program. Although it is relatively easy to identify an encrypted file, it is usually not so easy to decrypt it. The analyst may be able to identify the encryption method by examining the file header, identifying encryption programs installed on the system, or finding encryption keys (which are often stored on other media). Once the encryption method is known, the analyst can better determine the feasibility of decrypting the file.

Although an analyst can detect the presence of encrypted data rather easily, the use of steganography is more difficult to detect. *Steganography*, also known as *steg*, is the embedding of data within other data. Digital watermarks and hiding words and information within images are examples of steganography. Some techniques an analyst may use to locate stegged data include looking for multiple versions of the same image, identifying the presence of grayscale images, searching metadata and registries, using histograms, and using hash sets to search for known steganography software. Once certain stegged data exists, analysts might be able to extract the embedded data by determining what software created the data and then finding the stego key, or using brute force and cryptographic attacks to determine a password.[34] However, these efforts are often unsuccessful and can be extremely time-consuming, particularly if the analyst does not find the presence of known steganography software on the media being reviewed. Also, some software programs can analyze files and estimate the probability that the files were altered with steganography.

Analysts may also need to access non-stegged files that are protected by passwords. Passwords are often stored on the same system as the files they protect, but in an encoded or encrypted format. Various utilities are available that can crack passwords placed on individual files, as well as operating system passwords.[35] Most cracking utilities can attempt to guess passwords, as well as performing brute force

[33] Although volumes and partitions can be encrypted on some operating systems, this is not common due to corruption and other functional problems that may result in a complete loss of data if only a sector of data is corrupted. Encryption of individual files and folders is far more common, and is supported by many newer operating systems.

[34] Further discussion regarding steganography is outside the scope of this document. For more information, see the article *An Overview of Steganography for the Computer Forensics Examiner* by Gary Kessler, available at http://www.fbi.gov/hq/lab/fsc/backissu/july2004/research/2004_03_research01.htm.

[35] An example of an open source password cracking utility is John the Ripper, which supports multiple OS’s and file types. Additional password cracking utilities are listed on several Web sites listed in Appendix F, including Computer Forensics Tools (http://www.forensix.org/tools/) and The Ultimate Collection of Forensic Software (TUCOFS) (http://www.tucofs.com/tucofs.htm).

attempts that try every possible password. The time needed for a brute force attack on an encoded or encrypted password can vary greatly depending on the type of encryption used and the sophistication of the password itself. Another approach in some instances is to bypass a password. For example, an analyst could boot a system and disable its screensaver password, or bypass a BIOS password by pulling the BIOS jumper from the system's motherboard or using a manufacturer's backdoor password.[36] Of course, bypassing a password may mean rebooting the system, which may be undesirable. Another possibility is to attempt to capture the password through network or host-based controls (e.g., packet sniffer, keystroke logger), with proper management and legal approval.

4.3.3 Analyzing the Data

An analyst's toolkit should contain various tools that provide the ability to perform quick reviews of data as well as in-depth analysis. Many products allow the analyst to perform a wide range of processes to analyze files and applications, as well as acquiring 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.

Although such products are invaluable in performing analysis, understanding what processes should be run to answer particular questions about the data is an important first step. An analyst may need to provide a quick response or just answer a simple question about the acquired data. In these cases, a complete analysis may not be necessary or feasible. As a result, an analysis toolkit should contain applications that can accomplish the data analysis in many ways and can be run quickly and efficiently from floppy disks, CDs, or an analyst workstation. The following list mentions several types of processes 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 before it is collected, and more efficient (e.g., not needing native applications for viewing each type of file). Various tools are available to view common types of files, and there are also specialized tools solely for viewing graphics. If available file viewers do not support a particular file format, then the original source application should be used; if it is not available, then it may be necessary to research the file's format and manually extract the data from the file.[37]

+ **Uncompressing Files.** Compressed files may contain files with useful information, as well as other compressed files. Therefore, it is important that the analyst locates and extracts compressed files. Uncompressing files should be performed early in the process to ensure the contents of compressed files are included in searches and other actions. Analysts should keep in mind that compressed files might contain malicious content, such as compression bombs, which are files that have been repeatedly compressed, typically dozens or hundreds of times. Compression bombs can cause examination tools to fail or consume considerable resources; they may also contain malware and other malicious payloads. Although there is no definite way to detect these prior to uncompressing, their impact may be minimized. For instance, the examination system should use up-to-date antivirus software and should be standalone to limit the effects to just that system. Also, an image of the examination system should be created so that if needed, the system can be restored.

+ **Graphically Displaying Directory Structures.** This makes it easier and faster for analysts to gather general information about the contents of media, such as the type of software installed and

[36] See the article *How to Bypass BIOS Passwords* (available at http://labmice.techtarget.com/articles/BIOS_hack.htm) for more information and examples of known BIOS backdoor passwords.

[37] Web sites such as Wotsit's Format (http://www.wotsit.org/) contain file format information for hundreds of file types.

the likely technical aptitude of the user(s) that created the data. Most products can display Windows, Linux, and Unix directory structures, while other products are specific to Macintosh directory structures.

- **Identifying Known Files.** The benefit of finding files of interest is obvious, but it is often beneficial to eliminate unimportant files from consideration, such as known good OS and application files. Analysts can use hash sets created by NIST's National Software Reference Library (NSRL) project[38] or personally created hash sets[39] as a basis for identifying known benign and malicious files. Hash sets typically use the SHA-1 and MD5 algorithms to establish message digest values for each known file.

- **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 aid the analyst in reducing the volume of information to review. In addition to proprietary file formats that cannot be string searched without additional tools, 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 may cause problems with string searches; some searching tools attempt to overcome this by providing language translation functions. Another possible issue is 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 resides in one cluster, and the rest of the string resides in a non-adjacent cluster. Similarly, some search tools may report a false match if part of a search string resides in one cluster and the remainder of the string resides in another cluster that is not part of the same file that contains the first cluster.

- **Accessing File Metadata.** File *metadata* provides details about any given file. For example, acquiring the metadata on a graphic file might provide the graphic's creation date, copyright information, description, and the creator's identity.[40] Metadata for graphics generated by a digital camera might include the make and model of the digital camera used to take the image, as well as F-stop, flash, and aperture settings. For word processing files, metadata could specify the author, the organization that licensed the software, when and by whom edits were last performed, and user-defined comments. Special utilities can extract metadata from files.

Another important aspect of analyzing the data is examining 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 data analysis. For example, such information can be used to reconstruct a timeline of activities. While this may seem like a simple task, it is often complicated by unintentional or intentional discrepancies in time settings among systems. Knowing the time, date, and time zone settings for a computer whose data will be analyzed can greatly assist an analyst; Section 5 describes this in more detail.

It is usually beneficial to analysts if an organization maintains its systems with accurate timestamping. The Network Time Protocol (NTP) synchronizes the time on a computer with an atomic clock run by

[38] The NSRL home page is located at http://www.nsrl.nist.gov/.

[39] Analysts may also create hashes of system files periodically; these hash sets are then available when an event occurs so that the analyst can quickly eliminate known benign files from examination. Analysts should rely on standard hash sets such as those from the NSRL project whenever possible, and create custom hash sets primarily for organization-specific files.

[40] Only certain types of graphics files can include metadata; for example, JPEG-format graphics might have metadata, but bitmap-format graphics cannot.

NIST or other organizations. Synchronization helps to ensure that each system maintains a reasonably accurate measurement of time.

If multiple tools are used to complete the 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 operating system. Write-blockers may be used to prevent these tools from modifying the MAC times. However, although write-blockers can prevent the MAC times from being modified on the media, they cannot prevent the operating system from caching the changes in memory (i.e., storing the changes in RAM). Consequently, the operating system may report the cached MAC times rather than the actual MAC times, thereby returning inaccurate results. The analyst should be aware that the last access time of data files and directories might change between queries, depending upon the tool used to perform the query. Because of these issues, analysts should take care to choose a MAC viewing method and record the details of that method.

In many cases, analysis involves not only data from files, but also data from other sources, such as the operating system state, network traffic, or applications. Section 8 provides examples of how data from files and data from other sources can be correlated through analysis.

4.4 Recommendations

The key recommendations presented in this section for using data from data files are summarized below.

- **Analysts should work with copies of files, not the original files.** During the acquisition phase, the analyst should make one or more copies of the desired files or filesystems. The analyst can then work with a copy of the files without affecting the originals. A physical backup should be performed if preserving file times is important. A logical backup is sufficient for informal file acquisition from live systems.

- **Analysts should preserve and verify file integrity.** Using a write-blocker during a backup 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 should be accessed as read-only whenever possible; write-blockers can also be used to prevent writes to the image file or restored image.

- **Analysts should rely on file headers 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 definitively identify the type of data stored in many files by looking at their file headers.

- **Analysts should have a toolkit for data examination.** It should contain various tools that provide the ability 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 an analysis workstation.

5. Using Data from Operating Systems

An *operating system* (OS) is a program that runs on a computer and provides a software platform on which other programs can run. In addition, an OS is responsible for processing input commands from a user, sending output to a display, interacting with storage devices to store and retrieve data, and controlling peripheral devices such as printers and modems. Some common OSs for workstations or servers include various versions of Windows, Linux, Unix, and Mac OS. Some network devices, such as routers, have their own proprietary OSs (e.g., Cisco Internetwork Operating System [IOS]). Personal digital assistants (PDA) often run specialized operating systems, including PalmOS and Windows CE.[41] Many embedded systems such as cellular phones, digital cameras, and audio players also use operating systems.[42] This section discusses the components of an OS that may be relevant to data analysis, and provides guidance for doing so with common workstation and server operating systems.[43]

5.1 OS Basics

OS data exist 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. From an analysis perspective, many types of non-volatile and volatile data may be of interest. This section discusses commonly used types of OS data.

5.1.1 Non-Volatile Data

The primary source of non-volatile data within an OS is the filesystem.[44] The filesystem is also usually the largest and richest source of data within the OS, containing most of the information recovered during a typical examination. The filesystem provides storage for the OS on one or more media.[45] A filesystem typically contains many different types of files, each of which may be of value to analysts in different situations. Also, as noted in Section 4.1.2, important residual data also can be recovered from unused filesystem space. The following list describes several types of data that are commonly found within OS filesystems:

+ **Configuration Files**. The OS may use configuration files to store OS and application settings.[46] For example, configuration files could list the services to be started automatically after system boot, and specify the location of log files and temporary files. Users may also have individual OS and application configuration files that contain user-specific information and preferences, such as hardware-related settings (e.g., screen resolution, printer settings) and file associations. Configuration files of particular interest are as follows:

[41] For more information on PDA data analysis, see NIST SP 800-72, *Guidelines on PDA Forensics*, available at http://csrc.nist.gov/publications/nistpubs/index.html.

[42] A discussion of the types of information that can be found on these types of devices and the methods for acquiring and examining the information is beyond the scope of this document. Because of the wide variety of devices and the knowledge and equipment needed for many cases, most organizations may find it best to secure such a device and transfer it to an appropriate party, such as a law enforcement agency, that is experienced in acquiring and examining data from such devices.

[43] Guidance specific to data from proprietary and specialized operating systems is outside the scope of this document; however, many of the concepts described in this section should also apply to them.

[44] This may not be true for some devices, such as consumer electronics that do not use standard filesystems.

[45] In some cases, the filesystem may be "stored" in dynamic memory. The term *memory filesystems* refers to filesystems that reside only in a system's memory. Such filesystems are considered volatile data. Filesystems, including an entire bootable OS implementation, may also reside on removable media such as flash drives.

[46] On Windows systems, many configuration settings reside in a set of special files known as the *registry*. For more information on the registry, see Microsoft Knowledge Base article 256986, *Description of the Microsoft Windows Registry*, available at http://support.microsoft.com/?id=256986.

  - **Users and Groups.** The OS keeps a record of its user accounts and groups. Account information may include group membership, account name and description, account permissions, account status (e.g., active, disabled), and the path to the account's home directory.

  - **Password Files.** The OS may store password hashes in data files. Various password-cracking utilities may be used to convert a password hash to its clear text equivalent for certain OSs.

  - **Scheduled Jobs.** The OS maintains a list of scheduled tasks that are to be performed automatically at a certain time (e.g., perform a virus scan every week). Information that can be gleaned from this include the task name, the program used to perform the task, command line switches and arguments, and the days and times when the task is to be performed.

- **Logs.** OS log files contain information about various operating system events, and may also hold application-specific event information. Depending on the OS, logs may be stored in text files, proprietary-format binary files, or databases. Also, some OSs write log entries to two or more separate files. 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. Typically, failed events and the most significant successful events are logged, but many operating systems permit system administrators to specify which types of events will be logged. The details logged for each event also vary widely; each event is usually timestamped, and other supporting information could include event codes, status codes, and username.

  - **Audit Records.** Audit records contain security event information such as successful and failed authentication attempts and security policy changes. OSs typically permit system administrators to specify which types of events should be audited. Administrators also can configure some OSs to log successful, failed, or all attempts to perform certain actions.

  - **Application Events.** Application events are significant operational actions performed by applications, such as application startup and shutdown, application failures, and major application configuration changes. Section 7 contains more information on application event logging.

  - **Command History.** Some operating systems have separate log files (typically for each user) that contain a history of the OS commands performed by each user.

  - **Recently Accessed Files.** An operating system might log the most recent file accesses or other usage, creating a list of the most recently accessed files.

- **Application Files**. Applications may be composed of many types of files, including executables, scripts, documentation, configuration files, log files, history files, graphics, sounds, and icons. Section 7 provides an in-depth discussion of application files.

- **Data Files**. Data files store information for applications; examples of common data files include text files, word processing documents, spreadsheets, databases, audio files, and graphics files. Also, when data is printed, most operating systems create one or more temporary print files that contain the print-ready version of the data. Sections 4 and 7 discuss application data files in more depth.

+ **Swap Files**. Most OSs use swap files in conjunction with random access memory (RAM) to provide temporary storage for data often used by applications. Swap files essentially extend the amount of memory available to a program by allowing pages (or segments) of data to be swapped in and out of RAM. Swap files may contain a broad range of OS and application information such as login IDs, password hashes, and contact information. Section 5.1.2 discusses the contents of memory in more detail.

+ **Dump File**s. Some OSs offer 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. When the system is next turned on, the state of the system is restored.

+ **Temporary Files**. During the installation of an operating system, 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. Temporary files are also created when many applications are run; again, such files should be deleted when the application is terminated, but this does not always happen. Temporary files could contain copies of other files on the system, application data, or other information.

Although filesystems are the primary source of non-volatile data, another data source of interest is the Basic Input/Output System (BIOS). The BIOS contains many types of hardware-related information, such as the attached devices (e.g., CD-ROM drives, hard drives), the types of connections and interrupt request line (IRQ) assignments (e.g., serial, USB, network card), motherboard components (e.g., processor type and speed, cache size, memory information), system security settings, and hot keys. The BIOS also communicates with RAID drivers and displays the information provided by the drivers. For example, the BIOS views a hardware RAID as a single drive and a software RAID as multiple drives. The BIOS typically permits the user to set passwords, which restrict access to the BIOS settings and may prevent the system from booting without supplying the password. The BIOS also holds the system date and time.

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 may contain many types of data and information that may be of interest. For example, RAM often contains frequently and recently accessed data, such as data files, password hashes, and recent commands. Also, similar to filesystems, RAM may also 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 in these units. Sometimes an application may not request an entire unit, but it is given one anyway. Thus, it is possible that residual data could reside in the unit of memory that was allocated to an application, although it may not be addressable by the application. For performance and efficiency, some operating systems vary the size of the units they allocate, which tends to result in smaller memory slack sizes.

+ **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.

The following list includes some of the other most significant types of volatile data that may exist within an OS:

+ **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. For example, many hosts are assigned IP addresses dynamically by another host, meaning that their IP addresses are not part of the stored configuration. Many hosts also have multiple network interfaces defined, such as wired, wireless, VPN, and modem; the current network configuration indicates which interfaces are currently in use. Also, users 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. Section 6 provides an in-depth examination of the significance of network connections.

+ **Running Processes**. *Processes* are the programs that are currently executing on a computer. The processes include services offered by the OS and applications run by administrators and users. Most OSs offer ways to view a list of the currently running processes. This list can be examined to determine the services that are active on the system, such as a Web server, and the programs that individual users are running (e.g., encryption utility, word processor, e-mail client). Process lists may also indicate which command options were used, as described in Section 7. 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.[47] However, login session information may be available only if the computer has been configured to audit logon attempts. Logon records can help to determine a user's computer usage habits and confirm whether a user account was active when a certain 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 operating system might differ from the BIOS due to OS-specific settings such as time zone.

[47] Impersonation can allow a regular system user to have higher system privileges in order to accomplish certain tasks. For example, a particular program may require administrator access in order to run. Through impersonation, a regular user may be given those permissions to run the application and then be reverted back to the usual privileges.

5.2 Acquiring OS Data

As described in Section 5.1, OS data exist in both non-volatile and volatile states. Non-volatile OS data such as filesystem data can be acquired using the approaches discussed in Section 4 for performing logical and physical backups. Volatile OS data should be collected before the computer is powered down. Sections 5.2.1 and 5.2.2 provide recommendations for acquiring volatile and non-volatile OS data, respectively. Section 5.2.3 discusses technical issues that may impede the acquisition of data.

5.2.1 Acquiring Volatile OS Data

Volatile OS data involving an event can be acquired 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 if the volatile OS data needs to be preserved. Ideally, the criteria for making this decision should have been documented in advance so that the analyst can make the best decision immediately. The importance of this decision cannot be stressed enough, because powering off the system or even disconnecting it from a network may eliminate the opportunity to acquire potentially important information. For example, if a user recently ran encryption tools to secure data, the computer's RAM may contain password hashes, which could be used to determine the passwords.

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 may change and other volatile OS data may be altered. In addition, a malicious party may have installed rootkits that are designed to return false information, delete files, or perform other malicious acts. Therefore, the risks associated with collecting volatile OS data should be weighed against the potential for recovering important information to determine if the effort is warranted. If a live system is in sleep mode or has visible password protection, analysts need to decide whether or not to alter the state of the system by waking it from sleep mode or attempting to crack or bypass the password protection, so that analysts can attempt to collect volatile data. The effort needed to collect the volatile data might not be merited in some cases, so analysts might instead decide to perform a shutdown, as described in Section 5.2.2.

When collecting volatile OS data, all tools that might be used should be placed on a floppy disk, CD-ROM, or USB flash drive, from which the tools should be executed. Doing so allows OS data to be collected while causing the least amount of disturbance to the system. Also, only trusted tools should be used since a user may have replaced system commands with malicious programs, such as one to format a hard disk or return false information. However, 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 all user-level tools, even if trusted tools are used.

When creating a collection of trusted 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 DLLs and other supporting files are not needed. This eliminates the need to place the appropriate versions of DLLs on the trusted tool media and increases the reliability of the tools. The analyst should know how each tool affects or alters the system before acquiring the volatile data. The message digest of each tool should be computed and stored safely to verify file integrity. 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 each command was executed, and what the output of each command was.

The media containing the tools should protect them from changes. Floppy disks should be write-protected to ensure that no changes are made to the tools. When using a CD-ROM, only a write-once CD (i.e., CD-R) should be used to store the tools since the contents of a rewriteable CD could be altered by

CD-burning utilities on the user's computer. After the tools have been burned to a write-once CD, the disc should be finalized, which ensures that no additional data can be written to it.[48] Some CD-burning utilities also allow the current session to be closed. However, *closing* a session simply indicates to the CD-burning utility that no additional data will be written to the disc in the current session; it does not prevent additional data from being written to the disc in a different session (often referred to as a multi-session disc). Therefore, the session should be finalized, not closed, when creating a toolkit CD.

Because the media containing the tools should be write-protected, the results produced by the tools cannot be placed onto the tool media. Analysts often direct tool output to a floppy disk, but the prevalence of floppy disk drives on computing devices is decreasing. As a result, alternative methods of collecting output have been developed. Specially prepared CDs and USB flash drives containing a Windows or Linux-based environment can be used to gather output without changing the state of the system, and typically direct the output to another USB flash drive, external hard drive, or other writable media.

The following lists several types of volatile OS data and explains how tools may be useful in collecting each type of data:[49]

+ **Contents of Memory.** There are several utilities that can copy the contents of RAM to a data file and assist with subsequent analysis of the data. On most systems, it is unavoidable to alter RAM while running a utility that attempts to make a copy of RAM. Therefore, the goal is to do so with as small a footprint as possible so as to minimize the disruption of RAM.

+ **Network Configuration.** Most operating systems include a utility that displays the current network configuration, such as **ifconfig** on Unix systems and **ipconfig** on Windows systems. Information that can be provided through network configuration utilities may include the hostname, the physical and logical network interfaces, and configuration information for each interface (e.g., IP address, MAC address, current status).

+ **Network Connections.** Operating systems typically provide a method for displaying a list of the current network connections. Both Windows and Unix-based systems usually include the **netstat** program, which lists network connections by source and destination IP addresses and ports, and also lists which ports are open on each interface.[50] Third-party utilities are available that can display port assignments for each program. Most operating systems also can display a list of remotely mounted filesystems, which provides more detailed information than a network connection list. Section 6.2.7 provides additional information on gathering network connection information.[51]

+ **Running Processes.** All Unix-based systems offer the **ps** command for displaying the currently running processes. Although Windows offers a GUI-based process list utility, the Task Manager, it is usually preferable to have a text-based listing. Third-party utilities can be used to generate a text list of running processes for Windows systems.

[48] Finalizing a CD is a typical feature of CD burning utilities.

[49] Many resources are available that list the hundreds of tools available for analysts. Appendix F lists several Web sites that contain more information on computer data analysis tools.

[50] Another way of identifying the open ports is by running port scanning software from another system. Port scanning software sends network traffic to various ports and analyzes the responses, as well as missing responses, to determine which ports are open. However, port scanning might produce inaccurate results due to security controls, such as host-based firewalls that block the scans; also, the scans could change the state of the system. Accordingly, port scans are best suited for informal data acquisition and for information collection when access to the operating system is not available.

[51] More information on fport is available at http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/proddesc/fport.htm.

- **Open Files.** All Unix-based systems offer the **lsof** command for displaying a list of open files. Third-party utilities can be used to generate text lists of open files for Windows systems.
- **Login Sessions.** Some operating systems have built-in commands for listing the currently logged on users, such as the **w** command for Unix systems, which also lists the source address of each user and when the user logged into the system. Third-party utilities are available that can list currently connected users on Windows systems.
- **Operating System Time.** There are several utilities that can be used to retrieve the current system time, time zone information, and daylight savings time settings. On Unix systems, the **date** command can be used to retrieve this information. On Windows systems, the **date**, **time**, and **nlsinfo** commands can be used collectively to retrieve this information.

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

- **OS Command Prompt**. This is a utility that provides an OS command prompt through which the other tools in the toolkit can be executed, such as **cmd** on Windows systems.
- **SHA-1 Checksum**. A utility that can compute the SHA-1 message digest of data files is helpful in file verification. It may also be useful to include in the toolkit a list of SHA-1 message digests for system data files associated with the target OS to assist with file verification. Utilities are available for various OSs for this purpose.[52]
- **Directory List**. A utility for listing the contents of directories should be included for navigating a filesystem and seeing its contents. Practically all operating systems include such a utility; for example, the **ls** command is used on Unix systems, while on Windows systems the **dir** command is used.
- **String Search**. A utility for performing a text string search may be useful in identifying data files of interest. Unix systems offer the **grep** command for performing text string searches, and a third-party **grep** utility is also available on Windows systems.[53]
- **Text Editor**. A simple text editor may be useful for viewing text files or composing notes. Numerous text editors are available, such as **Notepad** on Windows systems and **vi** on Unix systems.

The types of volatile data that should be acquired with the toolkit depend on the specific need. For instance, if a network intrusion is suspected, then it may be useful to collect network configuration information, network connections, login sessions, and running processes to determine how someone gained access to a system. If an investigation concerns spyware, then the contents of RAM, the list of running processes, the list of open files, network configuration information, and network connections may reveal social security and credit card numbers, programs used to obtain or encrypt the data, password hashes, and methods that may have been used to transfer the information over a network. When in doubt, it is normally a good idea to collect as much volatile data as possible since all opportunities to acquire the data will be lost once the computer is powered down. A determination can be made afterwards as to which acquired volatile data should be examined. An automated script on a toolkit CD can be used for consistency in collecting volatile data. The script can include ways to transfer the collected information to local storage media, such as a thumb drive, and to networked drive locations.

[52] See http://lists.gpick.com/pages/Checksum_Tools.htm for more information on checksum utilities.

[53] One Windows version of grep is available at http://unxutils.sourceforge.net/.

Since volatile data has a propensity to change over time, the order and timeliness in which volatile data should be acquired is important. In most cases, analysts first should collect information on network connections and login sessions, since 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 acquired later. The recommended order in which volatile data generally should be collected, listed 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 Acquiring Non-Volatile OS Data

After obtaining volatile OS data, analysts often need to acquire non-volatile OS data as well. To do so, the analyst first needs to decide whether the system should be shut down or not. This affects the ability to perform physical backups and many logical backups, but can also change which OS data is preserved. Most systems can be shut down through two methods, as follows:

- **Perform a Graceful OS Shutdown.** Nearly every OS offers a shutdown option.[54] This causes the OS to perform cleanup activities, such as closing open files, deleting temporary files, and possibly clearing the swap file, before shutting down the system. A graceful shutdown can also trigger the removal of malicious material; for example, memory-resident rootkits may disappear, and Trojan horses may remove evidence of their malicious activity. The OS is typically shut down from the account of the administrator or the current user of the system (if the current user has sufficient privileges).
- **Remove Power from the System**. Disconnecting the power cord from the back of the computer (and removing the batteries on a laptop or other portable device) can preserve swap files, temporary data files, and other information that might be altered or deleted during a graceful shutdown.[55] Unfortunately, a sudden loss of power can cause some OSs to corrupt data, such as open files. Also, for some consumer devices, such as PDAs and cell phones, removing battery power can cause a loss of data.[56]

Analysts should be aware of the characteristics of each operating system and choose a shutdown method based on the typical behavior of the OS and the types of data that need to be preserved. For example,

[54] For example, on Windows systems, analysts could use the Shut Down feature on the Start menu.

[55] Disconnecting a power cord from the electrical wall outlet is not recommended because the computer's power cord may be plugged into an Uninterruptible Power Supply (UPS) unit.

[56] Maintaining power for such devices often needs to be performed on an ongoing basis. If a device does not have regular power, typically its memory will be sustained by battery power only in the short term (weeks at most, minutes at worst), even if the device is powered off. For long-term storage of consumer devices that contain important data, power should be maintained to preserve the memory of the devices.

DOS and Windows 95/98 systems generally do not corrupt data when power is removed suddenly, so removing power should preserve data. Other operating systems may corrupt data, such as open files or files that were being accessed at the time, from a loss of power, so a graceful shutdown is generally best unless swap files or temporary data files are of particular interest, or if the system may contain rootkits, Trojan horses, or other malicious programs that might be triggered by a graceful shutdown. After performing a shutdown, the analyst should then acquire filesystem data from the system's storage media using the methods discussed in Section 4.

Once the filesystem data has been acquired, tools can be used to acquire specific types of data from the filesystem. Acquiring regular files, such as data, application, and configuration files, is relatively straightforward, and is described in more detail in Section 4. The following lists several other types of non-volatile operating system data and explains how tools may be useful in acquiring each type from the filesystem:[57]

- **Users and Groups**. Operating systems maintain a list of users and groups that have access to the system. On Unix systems, users and groups are listed in **/etc/passwd** and **/etc/groups**, respectively. In addition, the **groups** and **users** commands can be used to identify users who have logged onto the system and the groups to which they belong. On Windows systems, the **net user** and **net group** commands can be used to enumerate the users and groups on a system.
- **Passwords.** Most operating systems maintain password hashes for users' passwords on disk. On Windows systems, third-party utilities can be used to dump password hashes from the Security Account Manager (SAM) database. On Unix systems, password hashes are usually in the **/etc/passwd** or **/etc/shadow** file. As described in Section 4.3.2, password cracking programs can then be used to attempt to extract passwords from their hashes.
- **Network Shares.** A system may enable local resources to be shared across a network. On Windows systems, the **SrvCheck** utility can be used to enumerate network shares.[58] Third-party utilities can provide similar information for other operating systems.
- **Logs.** Logs that are not stored in text files may necessitate the use of log extraction utilities. For example, specialized utilities can retrieve information about recent successful and failed logon attempts on Windows systems. Most log entries on Unix systems are stored in text files by syslog or in the **/var/log** directory, so special utilities are not needed to acquire information from the logs.[59] Searching for filenames ending in **.log** should identify most log files.

Occasionally, analysts may need to acquire data from the BIOS, such as system date and time or processor type and speed.[60] Since the BIOS primarily contains information related to the system's hardware configuration, BIOS data acquisition is most likely to be needed when a system administrator is troubleshooting operational issues. Typically, analysts that need BIOS data first acquire any needed volatile data and filesystems, then reboot the system and hit the appropriate function key (generally specified in the initial screen during boot) to display the BIOS settings. If the BIOS password is set, the analyst may not be able to gain access to the BIOS settings easily, and may have to attempt to guess default passwords or circumvent the password protection. There are a variety of methods used to bypass BIOS passwords, including finding the appropriate manufacturer backdoor password, using a password

[57] Some of the tools described in this section could also be used to collect data from a live system.

[58] SrvCheck is available from the Windows Server 2003 Resource Kit.

[59] More information on the syslog protocol is available in RFC 3164, *The BSD Syslog Protocol*, available at http://www.ietf.org/rfc/rfc3164.txt.

[60] Hard drive information presented in the BIOS may be inaccurate for larger hard drives. Many drives now use Logical Block Addressing, which causes the BIOS to display incorrect drive geometry information. Analysts should be able to acquire the correct information by examining the physical label on the hard drive itself.

cracker, moving the appropriate jumper on the motherboard, or removing the CMOS battery (if possible). Each system might be different, so analysts should first research the system's characteristics as described in motherboard documentation to avoid harming a system unnecessarily.[61]

5.2.3 Technical Issues with Acquiring Data

There are potential technical issues that may impede the acquisition of OS data. Section 4 describes several filesystem-related issues; this section focuses on additional acquisition 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 overview of possible issues, but rather to provide information on common ones.

+ **OS Access.** Acquiring volatile data may be difficult because the analyst cannot readily gain access to the operating system. For instance, a user may run a password-protected screensaver or have the system locked; the analyst needs to circumvent this protection or find another way to gain access to volatile OS data.[62] If a password-protected screensaver is active, restarting the system might allow the analyst to bypass the screensaver, but would also cause all volatile OS data to be lost. Another possibility is that a host may use biometric-based authentication, such as a fingerprint reader, or another add-on authentication service; this could cause similar issues in accessing volatile OS data. There are third-party utilities for some OSs that claim to crack screen saver passwords without rebooting the system. These utilities generally rely on the CD drive's autorun feature; the utility automatically runs in the background, then locates the encrypted password and attempts to decrypt it.

+ **Log Modification.** The user may 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. One way of reducing the impact of logging changes is to configure systems to archive their log entries on a centralized server.

+ **Hard Drives with Flash Memory.** Occasionally, an analyst may 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 needs to find, guess, or crack the password to gain access to the drive.

+ **Key Remapping.** On some computers, individual keys or combinations of keystrokes can be remapped to perform a different function from their initial purpose. For example, a person could map the Ctrl, Alt, and Del keys so that they wipe the computer's hard drive instead of the expected action, rebooting the system. An analyst who uses the keyboard of a computer of interest could enter keystrokes that cause unexpected actions to be performed. Accordingly, the best way of avoiding key remapping problems is by acquiring data from the computer without using its keyboard. For example, an analyst could attach an analysis workstation to a computer of interest using a crossover network cable and run scripts from the workstation.

5.3 Examining OS Data

Various tools and techniques can be used to support the examination process. Many of the same ones discussed in Section 4.3 for examining acquired data files also can be used with acquired OS data. Also,

[61] More information on bypassing BIOS passwords is available at http://www.freelabs.com/~whitis/security/backdoor.html and http://labmice.techtarget.com/articles/BIOS_hack.htm.

[62] Several Web sites indicate ways to circumvent specific screen savers, such as taking advantage of known OS vulnerabilities. However, the screen saver bypass methods are of little use if the OS is unknown or the user has eliminated the vulnerabilities. General information regarding passwords is available from the Microsoft Knowledge Base article titled, "Information About Unlocking a Workstation" (http://support.microsoft.com/kb/q281250/) and the article titled "Password Information in Windows XP" (http://www.kellys-korner-xp.com/win_xp_passwords.htm).

as described in Section 7, security applications such as file integrity checkers and host-based intrusion detection systems can be very helpful in identifying malicious activity against operating systems. For instance, file integrity checkers can be used to compute the message digests of OS files and compare them against databases of known message digests to determine if any files have been compromised. If intrusion detection software is installed on the computer, it may contain logs that indicate the actions that were performed against the OS.

Another issue that analysts face is the examination of swap files and RAM dumps, which are large binary data files containing unstructured data. Hex editors can be used to open these files and examine their contents; however, manually trying to locate intelligible data using a hex editor on large files can be a time-consuming process. Filtering tools automate the process of examining swap and RAM dump files by identifying text patterns and numerical values that can potentially represent phone numbers, names of people, e-mail addresses, Web addresses, and other types of critical information.

Analysts often want to gather additional information on a particular program running on a system. After obtaining a list of the processes currently running on a system, analysts can look up the process name to obtain additional information, such as the process's purpose and manufacturer. 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 dynamic link libraries (DLL) on Windows systems, to determine which libraries are loaded and what their typical purposes are.

As Section 5.2 describes, analysts may acquire many different types of OS data, including multiple filesystems. Trying to sift through each type of data to find relevant information can be a time-intensive process. Analysts generally find it useful to identify a few data sources to review initially, and then find other likely sources of important information based on that review. Also, in many cases, analysis can involve data from other types of sources, such as network traffic or applications. Section 8 provides examples of how data from operating systems and other sources can be correlated through analysis.

5.4 Recommendations

The key recommendations presented in this section for using data from operating systems are summarized below.

+ **Analysts should act appropriately to preserve volatile OS data.** The criteria for determining whether volatile OS data needs to be preserved should be documented in advance so that analysts can make informed decisions as quickly as possible. The risks associated with collecting volatile OS data should be weighed against the potential for recovering important information to determine if the effort is warranted.

+ **Analysts should use a trusted toolkit for acquiring volatile OS data.** Doing so allows accurate OS data to be collected while causing the least amount of disturbance to the system and protecting the tools from changes. The analyst should know how each tool should affect or alter the system when acquiring data.

+ **Analysts should choose the appropriate shutdown method for each system.** Each way of shutting down a particular operating system can cause different types of data to be preserved or corrupted, so 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 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.[63] This section provides an introduction to network traffic, including descriptions of major sources of network traffic data (e.g., intrusion detection software, firewalls). It next discusses techniques for acquiring data from these sources and points out the potential legal and technical issues in data acquisition. The rest of the section focuses on the techniques and tools for examining data from network traffic. Because a basic knowledge of Transmission Control Protocol/Internet Protocol (TCP/IP) is necessary to understand the data, tools, and methodologies presented in this section, it begins with an overview of TCP/IP.

6.1 TCP/IP Basics

TCP/IP is widely used throughout the world to provide network communications. TCP/IP communications are composed of four layers that work together. When a user wants to transfer data across networks, the data is passed from the highest layer through intermediate layers to the lowest layer, with each layer adding additional information. The lowest layer sends the accumulated data through the physical network; the data is then passed up through the layers to its destination. Essentially, the data produced by a layer is encapsulated in a larger container by the layer below it. The four TCP/IP layers, from highest to lowest, are shown in Figure 6-1.

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 assure 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. Internet Protocol (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.

Figure 6-1. TCP/IP Layers

The four layers work together to transfer data between hosts. As shown in Figure 6-2, each layer encapsulates the previous layers. The following items describe each of the layers and point out the characteristics that are most pertinent to network data analysis. Section 6.1.5 explains how the layers relate to each other.

[63] Because nearly all network traffic of interest to organizations uses the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite, this section addresses only TCP/IP-based communications. However, most of the principles discussed in this section can be applied to other types of network traffic as well.

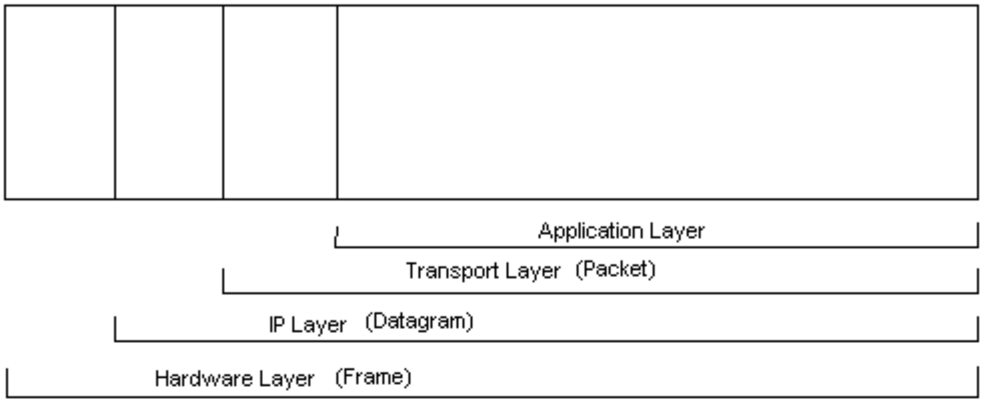


Figure 6-2. TCP/IP Encapsulation

6.1.1 Application Layer

The application layer enables applications to transfer data between an application server and client. An example of an application layer protocol is HyperText Transfer Protocol (HTTP), which transfers data between a Web server and a Web browser. Other common application layer protocols include Domain Name System (DNS), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and Simple Network Management Protocol (SNMP). There are hundreds of unique application layer protocols in common use, and many more that are not so common.[64] Regardless of the protocol in use, application data is generated and then passed to the transport layer for further processing. Section 7 focuses on application-related data acquisition and examination.

6.1.2 Transport Layer

The transport layer is responsible for packaging data so it can be transmitted between hosts. After the transport layer has encapsulated application data, the resulting logical units are referred to as *packets*. (A packet can also be created without application data—for example, when first negotiating a connection.) Each packet contains a *header*, which is composed of various *fields* that specify characteristics of the transport protocol in use, and optionally contains a *payload*, which holds the application data.

Most applications that communicate over networks rely on the transport layer to attempt to ensure reliable delivery of data. Generally, this is done by using the Transmission Control Protocol (TCP) transport layer protocol, which establishes a connection between two hosts and then makes a best effort to ensure the reliable transfer of data over that connection. Each TCP packet includes a source port and a destination port. One of the ports is associated with a server application on one system, and the other port is associated with a corresponding client application on the other system. Client systems typically select any available port number for application use, while server systems normally have a static port number dedicated to each application. Although many server ports are normally used by particular applications (e.g., FTP servers at port 21, HTTP servers at port 80), many server applications can be run from any port number, so it is unwise to assume that network traffic contains data from a certain application based solely on the server port number.

[64] Because of this, it is outside the scope of this document to discuss individual application layer protocols in detail.

When loss of some application data is not a concern (e.g., streaming audio, video), the User Datagram Protocol (UDP) is typically used. UDP involves less overhead and latency than TCP because UDP is connectionless; one host simply sends data to another host without any preliminary negotiations. UDP is also used for applications that are willing to take responsibility for ensuring reliable delivery of data, such as DNS, and applications that are intended for use only on local area networks, such as Dynamic Host Configuration Protocol (DHCP) and SNMP. Like TCP, each UDP packet contains a source port and a destination port. Although UDP and TCP ports are very similar, they are distinct from each other and not interchangeable. Some applications (such as DNS) can use both TCP and UDP ports; although such applications typically use the same number for the TCP port and the UDP port, this is not required.

6.1.3 IP Layer

The IP layer can also be called the network layer, because it is responsible for handling the addressing and routing of data that it receives from the transport layer. The IP header contains a field called IP Version, which indicates which version of the IP protocol is in use. Typically this is set to 4 for IPv4, but the use of IPv6 is increasing, so this field may be set to 6 instead.[65] Besides the IP Version field, other significant IP header fields are as follows:

- **Source and Destination IP Addresses.** These are the “from” and “to” addresses that are intended to indicate the endpoints of the communication.[66] Examples of IP addresses are 10.3.1.70 (IP version 4) and 1000:0:0:2F:8A:400:0427:9BD1 (IP version 6).
- **IP Protocol Number.** This indicates which transport layer protocol the IP payload contains.[67] Commonly used IP protocol numbers include 1 (ICMP), 6 (TCP), 17 (UDP), and 50 (Encapsulating Security Payload [ESP]).

The IP layer is also responsible for providing error and status information involving the addressing and routing of data; it does this with the Internet Control Message Protocol (ICMP). ICMP is a connectionless protocol that makes no attempt to guarantee that its error and status messages are delivered. Because it is designed to transfer limited information, not application data, ICMP does not have ports; instead, it has message types, which indicate the purpose of each ICMP message.[68] Some message types also have message codes, which can be thought of as sub-types. For example, the ICMP message type Destination Unreachable has several possible message codes that indicate what is unreachable (e.g., network, host, protocol). Most ICMP messages are not intended to elicit a response.[69]

IP addresses are often used through a layer of indirection. When people need to access a resource on a network, such as a Web server or e-mail server, they typically enter the server’s name, such as www.nist.gov, rather than the server’s IP address. The name, also known as a *domain name*, is mapped to the IP address through the Domain Name System (DNS) application layer protocol. The primary reason for entering a domain name instead of an IP address is that it is generally easier for people to remember a name than an IP address. Also, a host’s IP address might change over time; by referencing a

[65] There are other possible IP version numbers as well, although none are commonly used. The official list of valid IP Version field values is available at http://www.iana.org/assignments/version-numbers. This document assumes the use of IPv4, but the techniques described can easily be adapted for use with IPv6 (assuming that comparable tools that support IPv6 are available).

[66] IP addresses are often inaccurate or misleading for identifying the actual endpoints of communication. Section 6.3 discusses this topic in more detail.

[67] The official list of valid IP Protocol Number values is available at http://www.iana.org/assignments/protocol-numbers.

[68] The current list of valid ICMP types is available at http://www.iana.org/assignments/icmp-parameters.

[69] ICMP is designed to limit responses, particularly to error messages. If ICMP had not been designed this way, message loops could occur. For example, if host A received an ICMP error message from host B and responded with an error message, and host B responded to that error message with an error message, the two hosts could continue sending error messages regarding the error messages.

host by domain name, which is then mapped to the host's IP address, users can reach the host using the same name no matter what IP address the host is currently using.

6.1.4 Hardware Layer

As the name implies, the hardware layer involves the physical components of the network, including cables, routers, switches, and network interface cards (NIC). The hardware layer also includes various hardware layer protocols; Ethernet is the most widely used. Ethernet relies on the concept of a Media Access Control (MAC) address, which is a unique six-byte value (such as 00-02-B4-DA-92-2C) that is permanently assigned to a particular network interface card.[70] Each frame contains two MAC addresses, which indicate the MAC address of the NIC that just routed the frame and the MAC address of the next NIC that the frame is being sent to. As a frame passes through networking equipment (such as routers and firewalls) on its way between the original source host and the final destination host, the MAC addresses are updated to refer to the local source and destination. There may be several separate hardware layer transmissions linked together within a single IP layer transmission.

Besides the MAC addresses, each frame also contains an EtherType value, which indicates the protocol that the frame's payload contains (typically IP or Address Resolution Protocol [ARP]).[71] When IP is used, each IP address maps to a particular MAC address. (Multiple IP addresses can map to a single MAC address, so a MAC address does not necessarily uniquely identify an IP address.)

6.1.5 Layers' Significance in Network Data Analysis

Each of the four layers of the TCP/IP protocol suite contains important information. The hardware layer provides information on physical components, while other layers describe logical aspects. For events within a network, an analyst can map an IP address (logical identifier at the IP layer) to the MAC address of a particular network interface card (physical identifier at the physical layer), thereby identifying a host of interest. The combination of the IP protocol number (IP layer field) and port numbers (transport layer fields) can tell an analyst which application was most likely being used or targeted. This can be verified by examining the application layer data.

Network data analysis relies on all 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. Still, 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, 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 data analysis provides important support to the analysis of application-layer activities.

[70] The first three bytes of each MAC address indicate the vendor of the NIC; a list of mappings is available at http://standards.ieee.org/regauth/oui/oui.txt. However, various software utilities are publicly available that allow people to configure systems to spoof other MAC addresses. There have also been cases where manufacturers have accidentally created NICs with duplicate MAC addresses.

[71] EtherType value 0x0800 is IP, while 0x0806 is ARP. See http://www.iana.org/assignments/ethernet-numbers for more information on EtherType values.

6.2 Network Traffic Data Sources

Organizations typically have several types of sources of information on network traffic that may be useful. These sources collectively capture important data from all four TCP/IP layers. The following sections highlight the major categories of network traffic data sources—firewalls and routers, packet sniffers and protocol analyzers, intrusion detection systems, remote access, security event management software, and network forensic analysis tools—as well as several other types of data sources. For each likely source, this section explains its purpose and describes the type of data that is typically collected and that can potentially be collected.

6.2.1 Firewalls and Routers

Network-based devices such as firewalls and routers, and host-based devices such as personal firewalls, examine network traffic and permit or deny it based on a set of rules. Firewalls and routers are usually configured to log basic information for most or all denied connection attempts and connectionless packets; some log every packet.[72] Information logged typically includes the date and time the packet was processed, the source and destination IP addresses, and 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.

Network-based firewalls and routers that perform network address translation (NAT) may contain additional valuable data regarding network traffic. *NAT* is the process of mapping addresses on one network to addresses on another network; this is most often done by mapping private addresses from an internal network to one or more public addresses on a network that is connected to the Internet. NAT differentiates multiple internal addresses that are mapped to a single external address by assigning a different source port number to the external address for each internal address. The NAT device typically records each NAT address and port mapping.

Some firewalls also act as proxies. A *proxy* receives a request from a client, and then sends a request on the client's behalf to the desired destination. When a proxy is used, each successful connection attempt actually results in the creation of two separate connections: one between the client and the proxy server, and another between the proxy server and the true destination. Proxy servers may log basic information on each connection. Many proxies are application-specific, and some actually perform some analysis and validation of application protocols such as HTTP. The proxy may reject client requests that appear to be invalid and log information regarding these requests.[73]

Besides NAT and proxying services, firewalls and routers may also perform other functions, such as intrusion detection and virtual private networking (VPN). These functions are discussed in more detail in Section 6.2.3 and 6.2.4, respectively.

6.2.2 Packet Sniffers and Protocol Analyzers

Packet sniffers are designed to monitor network traffic on wired or wireless networks and capture packets. Normally, a network interface card (NIC) only accepts incoming packets that are specifically intended for it. When a NIC is placed in *promiscuous mode*, it accepts all incoming packets that it sees, regardless of

[72] Although logging all packets records more information about recent network activity than logging information for connections and connection attempts, space limitations might permit packets to be kept for a short time only. Also, the overhead required to record all packets might cause the system's performance to degrade.

[73] Some organizations configure their networks and network security so that all network traffic passing through the network perimeter for common applications is proxied, preventing individual users from bypassing the proxies. In such an environment, the proxy server logs can be particularly valuable for network data analysis.

their intended destinations. Packet sniffers generally work by placing the NIC in promiscuous mode; the user then configures the sniffer to capture all packets or only those with particular characteristics (e.g., certain TCP ports, certain source or destination IP addresses). Packet sniffers are commonly used to capture a particular type of traffic for troubleshooting or investigative purposes. For example, if IDS alerts indicate unusual network activity between two hosts, a packet sniffer could record all the packets between the hosts, potentially providing additional information for analysts.

Most packet sniffers are also *protocol analyzers*, which means that they can reassemble streams from individual packets and decode communications that use any of hundreds or thousands of different protocols.[74] Protocol analyzers usually can process not only live network traffic, but also packets that have been recorded previously in *capture files* by packet sniffers. Protocol analyzers are extremely valuable in displaying raw packet data in an understandable format. Protocol analyzers are discussed in more depth in Section 6.4 and Section 7.

6.2.3 Intrusion Detection Systems

Network-based intrusion detection systems (IDS) perform packet sniffing and analyze network traffic to identify suspicious activity and record relevant information.[75] *Host-based IDS* monitor characteristics of a system and events occurring within the system, which can include network traffic.[76] Unlike network-based IDS sensors, which can monitor all network traffic on a particular network segment, host-based IDS software is intended to monitor network traffic only for the host on which it is installed.[77] 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 on the attack.[78]

Some IDSs can be configured to capture packets related to suspicious activity. This can range from recording only the packet that triggered the IDS to label the activity suspicious, to recording the rest of the session. Some IDSs even have the ability to store all sessions for a short period of time, so that if something suspicious is detected, the previous activity in the same session can be preserved. The packets are captured primarily so that intrusion detection analysts can review them when validating IDS alerts and

[74] Examples of open source packet sniffer and protocol analyzer software for wired networks include Ethereal (http://www.ethereal.com/), TCPDump (http://www.tcpdump.org/), and WinDump (http://www.winpcap.org/windump/). Open source software is also available for wireless networks, including Ethereal and Kismet (http://www.kismetwireless.net/). Additional packet sniffer and protocol analyzer software products are listed on various Web sites, including the Talisker Security Wizardry Portal (http://www.networkintrusion.co.uk/protanalyzers.htm), Softpedia (http://www.softpedia.com/get/Network-Tools/Protocol-Analyzers-Sniffers/), Packet Storm (http://packetstormsecurity.org/defense/sniff/), and other Web sites listed in Appendix F.

[75] Some network-based IDS allow administrators to identify misuse as well as attacks. For example, an administrator (with proper approval) could configure the IDS with character strings of interest, such as an acronym or phrase associated with a sensitive project. The IDS can then search network traffic for file transfers, e-mails, and other forms of communication that use one of the character strings of interest.

[76] Examples of open source network-based IDS products include Bro (http://www.bro-ids.org/) and Snort (http://www.snort.org/). For more information on network-based and host-based IDS products, see the Talisker Security Wizardry Portal (http://www.networkintrusion.co.uk/ids.htm), Honeypots.net (http://www.honeypots.net/ids/products/), the Common Vulnerabilities and Exposures Web site (http://www.cve.mitre.org/compatible/product.html), and other Web sites listed in Appendix F.

[77] See NIST SP 800-31, *Intrusion Detection Systems*, for more information on network-based and host-based IDS. It is available at http://csrc.nist.gov/publications/nistpubs/index.html.

[78] Many IDS vendors provide help files that contain detailed information on each type of activity. IDS vendors also typically provide pointers to external sources of information, such as CERT®/CC advisories, Common Vulnerabilities and Exposures (CVE) numbers, and software vendor vulnerability announcements.

investigating suspicious activity. Some IDSs also have *intrusion prevention* capabilities, which means that they actively attempt to stop attacks in progress. Any use of intrusion prevention features should be indicated in the IDS logs.

6.2.4 Remote Access

Remote access servers are devices such as VPN gateways and modem servers that facilitate connections between networks. This often involves external systems connecting to internal systems through the remote access server, but could also include internal systems connecting to external or internal systems. Remote access servers typically record the origin of each connection, and may also indicate which user account was authenticated for each session. If the remote access server assigns an IP address to the remote user, this is also likely to be logged. Some remote access servers also provide packet filtering functions; this typically involves logging similar to that for firewalls and routers, as described in Section 6.2.1. Remote access servers typically work at a network level, supporting the use of many different applications. Because the servers have no understanding of the applications' functions, they usually do not record any application-specific data.

In addition to remote access servers, organizations typically use multiple applications that are specifically designed to provide remote access to a particular host's operating system. Examples include secure shell (SSH), telnet, terminal servers,[79] and remote control software. Such applications can typically be configured to log basic information for each connection, including source IP address and user account. Organizations also typically use many applications that are accessed remotely, such as client/server applications. Some of these applications may also log basic information for connections.

Although most remote access-related logging occurs on the remote access server or application server, in some cases the client also logs information related to the connection.

6.2.5 Security Event Management Software

Security event management (SEM)[80] 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.[81] 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 metaevents based on imported event data. Many SEM products not only can identify malicious activity, such as attacks and virus infections, but they can also 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.

Because SEM products can handle nearly any security event data source, such as OS logs, antivirus software alerts, and physical security device logs, SEM products may contain a wide variety of information regarding events. However, it is typical for only some data fields to be brought over; for example, if an IDS records packets, the packets may not be transferred to the SEM because of bandwidth and storage limitations. Also, because most data sources record information in different formats, SEM products typically *normalize* data—converting each data field to a standard format and labeling data consistently. Although this is beneficial for analysis (as described in Section 6.4), the normalization

[79] In this context, *terminal server* refers to products such as Microsoft Windows Terminal Services and Citrix Metaframe that provide graphical remote access to operating systems and applications.

[80] SEM software is listed on several Web sites listed in Appendix F, including the Common Vulnerabilities and Exposures (CVE) Web site (http://www.cve.mitre.org/compatible/product.html).

[81] Another common term for security event management is security information management (SIM).

process may occasionally introduce errors in the data or cause some data to be lost. Fortunately, SEM products typically do not alter the original data sources, so analysts can verify the accuracy of the data if needed.

6.2.6 Network Forensic Analysis Tools

Network forensic analysis tools (NFAT)[82] typically provide the same functionality as packet sniffers, protocol analyzers, and SEM software in a single product. While SEM software concentrates on correlating events among existing data sources (which typically include multiple network traffic-related sources), NFAT software is primarily focused on collecting and analyzing network traffic. NFAT software also offers additional features that further facilitate network data analysis, such as the following:

- Reconstructing events by replaying network traffic within the tool, ranging from an individual session (e.g., instant messaging between two users) to all sessions during a particular time period. The speed of the replaying can typically be adjusted as needed.
- Visualizing the traffic flows and the relationships among hosts. Some tools can even tie IP addresses, domain names, or other data to physical locations and produce a geographic map of the activity.
- 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 may be of use for data analysis in some capacity, including the following:

- **Dynamic Host Configuration Protocol (DHCP) Servers.** The DHCP service assigns IP addresses to hosts on a network as needed. Some hosts might have static IP addresses, which means that they always receive the same IP address assignment; however, most hosts typically receive dynamic assignments. This means that the hosts are required to renew their IP address assignments regularly, and that there is no guarantee that they will be assigned the same addresses. DHCP servers may contain assignment logs that include the MAC address, the IP address assigned to that MAC address, and the time the assignment occurred.
- **Network Monitoring Software.** Network monitoring software is designed to observe network traffic and gather statistics on it.[83] For example, it may record high-level information on traffic flows for a particular network segment, such as the amount of bandwidth typically consumed by various protocols. Network monitoring software may also collect more detailed information on network activity, such as the payload size and the source and destination IP addresses and ports for each packet. Some managed switches and other network devices offer basic network monitoring capabilities, such as collecting statistics on bandwidth usage.
- **Internet Service Provider Records.** Internet service providers (ISP) may collect network traffic-related data as part of their normal operations and when investigating unusual activity, such as extremely high volumes of traffic or an apparent attack. Normal ISP records often might

[82] Listings of NFAT software are available from Web sites listed in Appendix F, such as the Talisker Security Wizardry Portal (http://www.networkintrusion.co.uk/fornettools.htm).

[83] Open source network monitoring software includes EtherApe (http://etherape.sourceforge.net/) and IPaudit (http://ipaudit.sourceforge.net/). Packet sniffers, protocol analyzers, and IDS software may also perform basic network monitoring functions. See the Web sites listed in Appendix F for additional product names.

be kept only for days or hours. Section 6.3.1 discusses legal considerations with acquiring network traffic data from ISPs and other third parties.

+ **Client/Server Applications.** Some client/server applications used over networks may record information regarding successful and failed usage attempts, which could include connection-related data such as the client's IP address and port. The data fields recorded (if any) vary widely among applications.

+ **Hosts' Network Configurations and Connections.** Section 5.1.2 and 5.2.1 describe the types of network information that can be acquired from individual hosts, including the TCP and UDP ports at which a host is listening.

6.3 Acquiring Network Traffic Data

As described in Section 6.2, organizations typically have network traffic data recorded in many places during normal operations. Organizations also use the same data recording mechanisms to acquire additional data on an as-needed basis when investigating incidents or troubleshooting problems. For example, a network administrator or incident handler might deploy a packet sniffer to examine unusual packets sent by a host.

Network traffic data is usually recorded to a log or stored in a packet capture file. In most cases, acquiring the data is as simple as acquiring copies of the logs and packet capture files. Section 4 describes how to acquire files. If data is not stored in a file (e.g., traffic flow map displayed graphically, data displayed on the console screen only), screen captures or photographs of the screen may be needed. Although acquiring network traffic data is typically straightforward, there are several important legal and technical issues that can make data acquisition more complicated.

6.3.1 Legal Considerations

Acquiring network traffic can pose some legal issues, such as the capture (intentional or incidental) of information with privacy or security implications, such as passwords or the contents of e-mails. This could expose the information to staff members that are analyzing data or administering the recording systems (e.g., IDS sensors). 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 may violate an organization's data retention policy. It is also important to have policies regarding the monitoring of networks, as well as warning banners on systems that indicate that activity may be monitored.

Although most network traffic data acquisition occurs as part of regular operations, it may also be acquired as part of troubleshooting or incident handling. In the latter case, it is important to follow consistent processes and document all actions performed. For example, recording all packets sent and received by a particular user should be performed only after a formal request and approval process has been completed successfully. Organizations should have policies that clearly explain what types of monitoring can and cannot be performed without approval, and describe or reference the procedures that detail the request and approval process.

As privacy has become a greater concern to organizations, many have become less willing to share information with each other, including network 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 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

There are several potential technical issues that may impede the acquisition of data on network traffic. This section describes several major issues and provides guidance on what, if anything, can be done to mitigate each issue.

- **Data Storage.** When large volumes of network activity occur, 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.[84]

- **Encrypted Traffic.** When protocols such as IPsec, SSH, and 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. To acquire data on the decrypted traffic, a data source needs to be positioned where it can see the decrypted activity. For example, placing an IDS sensor immediately behind a VPN gateway can be effective at identifying anomalous activity in the decrypted communications. If communications are encrypted all the way to the internal host (e.g., an SSL-encrypted Web session), then devices monitoring network traffic cannot see the decrypted packets.

- **Services Running on Unexpected Ports.** Applications such as intrusion detection systems and protocol analyzers often rely on port numbers to identify which service is in use for a given connection. Unfortunately, as described in Section 6.1.2, most services can be run on any port number. Traffic involving services running on unexpected port numbers may not be captured, monitored, or analyzed properly, causing usage of unauthorized services (e.g., providing Web services on an atypical port) not to be detected. Another motivation is to slip traffic through perimeter devices that filter based on port numbers. 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. A classic example of an alternate access point is a modem in a user's workstation. If an attacker can dial into the workstation and gain access, then attacks can be

[84] Organizations should also provide sufficient data storage to keep logs associated with computer security incidents for a substantially longer time than other logs as needed. For example, General Records Schedule (GRS) 24, *Information Technology Operations and Management Records*, specifies that "computer security incident handling, reporting and follow-up records" should be destroyed "3 years after all necessary follow-up actions have been completed." GRS 24 is available from the National Archives and Records Administration at http://www.archives.gov/records-mgmt/ardor/.

launched from that workstation against other hosts. In such cases, little or no information regarding the network activity may be logged because the activity does not pass through firewalls, IDS-monitored network segments, and other common data collection points. Organizations typically address this 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 experience failures or outages occasionally for various reasons (e.g., system maintenance, software failures, attacks). In the case of dedicated monitoring systems, such as IDS sensors, using redundant equipment (e.g., two sensors monitoring the same activity) can lessen the impact of monitoring failures.[85] Another strategy is to perform multiple levels of monitoring, such as configuring network-based and host-based firewalls to log connections.

6.4 Examining Network Traffic Data

When an event of interest has been identified, analysts 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 may 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 reviewing 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 only be used effectively by well-trained, experienced analysts. In addition to understanding the tools, analysts also need to have solid knowledge of networking principles, common network and application protocols, network and application security products, and network-based threats and attack methods.[86] It is also very important that analysts have good 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 information supporting the applications and operating systems 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, their work should be easier and faster to perform. 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 need to understand the characteristics and relative value of each data source so that they can locate the relevant data quickly.

Because the analysis process is often complex, and analysts need extensive knowledge of networking and several areas of information security to analyze network traffic data effectively and develop sound conclusions, it is outside the scope of this guide to describe techniques for analyzing data and drawing conclusions in complex situations. Accordingly, this section focuses on the basic steps of the examination process, and also highlights some significant technical issues that analysts should consider.

[85] In most organizations, the cost of redundant monitoring makes it feasible only for the highest risk areas.

[86] Helpful references for analysts include lists of commonly used protocols and their typical port numbers, and Request for Comment (RFC) documents that explain the standards for various network and application protocols.

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 happens through one of two methods, as follows:

- Someone within the organization (e.g., help desk agent, system administrator, security administrator) has received an indication, such as an automated alert or a user complaint, that there may be a security or operational-related issue. The analyst has been asked to research the corresponding activity.
- The analyst's regular duties include reviewing security event data (e.g., IDS monitoring, network monitoring, firewall log review). During a review, 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 regarding the event as a basis for research. In most cases, the event was detected through a network traffic data source, such as an IDS sensor or a firewall, so the analyst can simply be pointed to that data source for more information. However, in some cases, such as a user complaint, it may not be apparent which data sources (if any) might contain relevant information or which hosts or networks may be involved. Analysts may need to rely on more general information, such as reports that several systems on the 4th floor have been rebooting themselves. Although data examination is easier if the event information is specific (e.g., IP addresses of affected systems), even general information provides the analyst with a starting point for finding the relevant data sources.

6.4.2 Examine Data Sources

As described in Section 6.2, organizations may have many sources of network traffic-related data. A single event of interest could be noted by many data sources, but it is inefficient or impractical to check each source individually. 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 was identified by an alert from one of the primary data sources. Accordingly, analysts typically consult one or a few primary data sources first.

For each data source that is examined, analysts should consider its fidelity. In general, analysts should have more confidence in original data sources than data sources that receive normalized (modified) data from other sources. Also, analysts should validate any data that is based on analyzing or interpreting data, 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).[87] Tools such as NFAT and IDS may also produce inaccurate alerts if they do not process all packets within a connection.[88] Validation should be based on an examination 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.

[87] From an analyst's perspective, the concept of false negatives is important because it means that security devices sometimes fail to report attacks that they have observed. Analysts should not assume that an activity is benign if security devices have not reported it as malicious.

[88] There are several possible causes for not processing all packets, including security device failure (e.g., outage, software bug), security device overload (e.g., unusually high volume of packets to process), and asynchronous routing. In asynchronous routing, the incoming packets and outgoing packets for a connection take different routes. If only one of these routes is monitored by a device such as an IDS sensor, the device can only see part of the connection.

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. For example, an attack may have occurred between two hosts on an internal network segment that is not monitored or controlled by network security devices. Analysts should then identify other likely data sources and examine them for evidence.
- **Insufficient or Unvalidated Data on Primary Sources.** Analysts may 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 then query the appropriate secondary data sources based on the pertinent data from the primary data sources. For example, if IDS records indicate an attack against the system at IP address 10.20.30.40 with an apparent origin of IP address 10.3.0.1, then querying other data sources using one or both IP addresses might find additional data regarding the activity. Analysts also use timestamps,[89] protocols, port numbers, and other common data fields to narrow the search as necessary.
- **Best Source of Data Elsewhere.** Occasionally, the best sources of network traffic data may be 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 may need to perform more data acquisition 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. See Section 6.2 for additional guidance on tools that can acquire data. Acquiring additional data may be helpful if the activity is ongoing or intermittent; if the activity has ended, there is no opportunity to acquire 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 recorded by each source may vary widely, the value of each source may vary in general and for specific cases. The following items describe the typical value of the most common data sources in network data analysis:

- **IDS Software.** IDS data is often the starting point for examining suspicious activity. Not only do IDS typically attempt to identify malicious network traffic at all TCP/IP layers, but they also log many data fields (and sometimes raw packets) that can be useful in validating events and correlating them with other data sources. As described earlier, IDS software produces false positives, so IDS alerts need to be validated. The extent to which this can be done depends on the amount of data recorded related to the alert and the information available to the analyst on the signature characteristics or anomaly detection method that triggered the alert.

[89] As mentioned earlier in this document, organizations should use time synchronization to keep systems' clocks consistent. Correlating an event among multiple network traffic sources is easier and more effective if the clocks are in sync. If event data sources are on separate devices, timestamps may be helpful in confirming the path that packets used. (When packets traverse a network, it takes them some amount of time to get from one device to the next.)

+ **SEM Software.** Ideally, SEM can be extremely useful for data analysis 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 is dependent upon 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 typically valuable for reviewing events that it has monitored. NFAT software usually offers features that support analysis, such as traffic reconstruction and visualization; Section 6.2.6 describes these in more depth.

+ **Firewalls, Routers, Proxy Servers, and Remote Access Servers.** By itself, data from these sources is usually of little value. Examining the data over time may indicate overall trends, such as an increase in blocked connection attempts. However, because these sources typically record little information about each event, the data provides little insight as to the nature of the events. Also, there may be many events logged each day, so the sheer volume of data can be overwhelming. The primary value of the data is to correlate events recorded by other sources. For example, if a host is compromised and a network IDS sensor detected the attack, querying the firewall logs for events involving the apparent attacking IP address may confirm where the attack entered the network and may indicate other hosts that the attacker may have attempted to compromise. Address mapping (e.g., NAT) performed by these devices is important for network data analysis because the apparent IP address of an attacker or a victim may actually be used by hundreds or thousands of hosts. Fortunately, analysts usually can review the logs to determine which internal address was in use.

+ **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 activity using a particular IP address. However, analysts should be mindful of the possibility that attackers on an organization's internal networks have falsified their MAC addresses or IP addresses, which is known as *spoofing*.

+ **Packet Sniffers.** Of all the network traffic data sources, packet sniffers can collect the most information on network activity. However, sniffers may capture huge volumes of benign data as well—millions or billions of packets—and typically they provide no indication 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 have identified as possibly malicious. Some organizations record most or all packets for some period of time so that when an incident occurs, the raw network data is available for analysis.[90] Packet sniffer data is best reviewed with a protocol analyzer, which interprets the data for the analyst based on knowledge of protocol standards and common implementations.

+ **Network Monitoring.** Network monitoring software is helpful in identifying significant deviations from normal traffic flows, such as those caused by distributed denial of service (DDoS) attacks. During these attacks, hundreds or thousands of systems launch simultaneous attacks against particular hosts or networks. Network monitoring software can document the impact of these attacks to network bandwidth and availability, as well as provide information on the apparent targets. Traffic flow data may also be helpful in investigating suspicious activity identified by other sources. For example, it may indicate whether a particular communications pattern has occurred in the preceding days or weeks.

[90] Many NFAT programs, as described in Section 6.2.6, provide this function, as well as additional capabilities.

+ **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. Section 6.4.4 discusses this in more depth.

6.4.2.2 Examination Tools

Because network data analysis 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 for certain situations. Analysts should be aware of the possible approaches to examining 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; for example, a particular protocol analyzer may not be able to translate a certain protocol or handle unexpected protocol data (e.g., illegal data field value). It can be helpful to have an alternate tool available that may not have the same flaw or omission. Analysts should review unexpected or unusual results produced by tools to confirm that they are valid.

Tools are often helpful in filtering data. For example, an analyst may need to search data without any concrete information that could narrow the search. This is most likely to occur when the analyst is responsible for performing periodic or ongoing reviews of security event data logs and alerts. If the volume of log entries and alerts is low, then reviewing the data is relatively easy—but in some cases, there may be many thousands of events listed per day. When a manual data review is not possible or practical, analysts should use an automated solution that filters the events and presents the analysts with only the events that are most likely to be of interest.

One effective technique is to import the logs into a database and run queries against them, either eliminating types of activity highly likely to be benign and reviewing the rest, or focusing on the types of activity most likely to be malicious. For example, if the initial suspicion is that the server was compromised through HTTP activity, then log filtering might start by eliminating everything except HTTP activity from consideration. An analyst who is very familiar with a particular data source can generally perform a blind search on it relatively quickly, but blind searches can take an extremely long time on unfamiliar data sources, because there may be little or no basis for eliminating certain types of activity from consideration.

Another option is to use a visualization tool, which presents security event data in a graphical format. This is most often used to represent network traffic flows visually, which can be very helpful in troubleshooting operational issues and identifying misuse. For example, attackers may use *covert channels*—using protocols in unintended ways to secretly communicate information (e.g., setting certain values in network protocol headers or application payloads). The use of covert channels is generally hard to detect, but one useful method is identifying deviations in expected network traffic flows.

Visualization tools are often included in NFAT software, as described in Section 6.2.6. Some visualization tools can perform traffic reconstruction—by examining timestamp and sequential data fields, the tools can determine the sequence of events and graphically display how the packets traversed the organization's networks. Some visualization tools can also be used to display other types of security event data. For example, an analyst could import intrusion detection records into a visualization tool, which would then display the data according to several different characteristics, such as source or destination IP address or port. An analyst can then suppress the display of known good activity so that only unknown events are shown.

Although visualization tools can be very effective for analyzing certain types of data, analysts typically experience a steep learning curve with such tools. Importing data into the tool and displaying it is usually relatively straightforward, but learning how to use the tool efficiently to reduce large datasets down to a few events of interest can take considerable effort. Traffic reconstruction may also be performed by protocol analyzers; although they generally lack visualization capabilities, they can turn individual packets into data streams and provide sequential context for activities.

6.4.3 Draw Conclusions

One of the most challenging aspects of network data analysis is that the available data is typically not comprehensive. In many cases, if not most, some of the network traffic data was not recorded and consequently has been lost. Generally, analysts should think of the analysis process in terms of a methodical approach that develops conclusions based on the data that is available and assumptions regarding the missing data (which should be based on technical knowledge and expertise). Although analysts should strive to locate and examine all available data regarding an event, this is not practical in some cases, particularly when there are many redundant data sources. During the examination process, the analyst should eventually have located, validated, and analyzed enough data to be able to reconstruct the event, understand its significance, and determine its impact. In many cases, additional data is available from other types of sources, such as data files or host operating systems. Section 8 provides examples of how data from network traffic and other sources can be correlated through analysis to get a more accurate and comprehensive view of what occurred.

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 may 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.

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 for identifying 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 is the focus. If an attack is ongoing, such as an extended denial of service attack, organizations may 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 attacks use spoofed IP addresses. Spoofing is far more difficult to perform successfully for attacks that require connections to be established, so it is most commonly used in cases where connections are not needed.[91] When packets are spoofed, usually the attacker has no interest in seeing the response. This is not always true—attackers could spoof an address from a subnet that they monitor, so that when the response goes to that system, they can sniff it from the network. Sometimes spoofing occurs by accident, such as an attacker

[91] Connectionless protocols such as ICMP and UDP are the most likely to be spoofed.

misconfiguring a tool and accidentally using internal NAT addresses. Sometimes an attacker spoofs a particular address on purpose—for example, the spoofed address may be the actual intended target of the attack, and the organization seeing the activity is simply a middleman.

+ **Many Source IP Addresses.** Some attacks appear to use hundreds or thousands of different source IP addresses. Sometimes this is accurate—for example, distributed denial of service attacks typically rely on large numbers of compromised machines performing a coordinated attack. Sometimes this is bogus—an attack may not require the use of real source IP addresses, so the attacker generates many different fake IP addresses to add confusion. Sometimes attackers will use one real IP address and many fake ones; in that case, it may be possible to identify the real IP address by looking for other network activity occurring before or after the attack that uses any of the same IP addresses. Finding a match does not confirm that it was the attacker's address; the attacker could have inadvertently or purposely spoofed a legitimate IP address that happened to be interacting with the organization.

+ **Validity of the IP Address.** Because IP addresses are often assigned dynamically, the system currently at a particular IP address may not be the same system that was there when the attack occurred. Also, many IP addresses do not belong to end-user systems, but instead to network infrastructure components that substitute their IP address for the actual source address, such as a firewall performing NAT. Some attackers use *anonymizers*, which are intermediate servers that perform activity on a user's behalf to preserve the user's privacy.

The following describes several possible ways of attempting to validate the identity of a suspicious host:

+ **Contact the IP Address Owner.** The Regional Internet Registries, such as the American Registry for Internet Numbers (ARIN),[92] 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 may 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, the analyst should provide information on the owner to the management and legal advisors for the analyst's organization, who can initiate contact with the organization or give the analyst approval to do so if needed. This is due primarily to concerns involving sharing information with external organizations; also, the owner of an IP address could be the person attacking the organization.

+ **Send Network Traffic to the IP Address.** Organizations should not send network traffic to an apparent attacking IP address to validate its identity. Any response that is generated cannot conclusively confirm the identity of the attacking host. If the IP address is for the attacker's system, the attacker may see the traffic and react by destroying evidence or attacking the host sending the traffic. 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.** As mentioned in Section 6.3.1, ISPs generally require a court order before providing any information to an organization on suspicious network activity. Accordingly, ISP assistance is generally only an option during the most serious network-based attacks, particularly those that involve IP address spoofing. ISPs have the ability to trace ongoing attacks back to their source, whether the IP addresses are spoofed or not.

[92] ARIN's web site is at http://www.arin.net/. The other registries are the Asia Pacific Network Information Centre (APNIC), located at http://www.apnic.net/; Latin American and Caribbean IP Address Regional Registry (LACNIC), located at http://lacnic.net/; and Réseaux IP Européens Network Coordination Centre (RIPE NCC), located at http://www.ripe.net/.

- **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 may show previous activity. Possible external sources include Internet search engines and online incident databases that allow searches by IP address.[93]

- **Look for Clues in Application Content.** Application data packets related to an attack may contain clues to the attacker's identity. Besides IP addresses, other 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

The 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 data analysis tools and techniques might inadvertently disclose sensitive information to analysts and others involved in data analysis activities. Also, long-term storage of sensitive information inadvertently captured by data analysis 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, 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.

- **Organizations should configure data sources to improve the collection of information.** Over time, operational experience should be used to improve the capabilities for data analysis. Organizations should periodically review and adjust the configuration settings of data sources to optimize the capture of relevant information.

- **Analysts should have solid technical knowledge.** Because current tools have rather limited analysis abilities, analysts need to 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 data sources that receive normalized data from other sources. Analysts should validate any unusual or unexpected data that is based on analyzing or interpreting 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, but it should be weighed against other important goals.

[93] One publicly available incident database is DShield, located at http://www.dshield.org/.

7. Using Data from Applications

The popularity of computers is largely due to the breadth and depth of applications that they can provide for users. By itself, an operating system is of little use; the applications running on the operating system, such as e-mail, Web browsers, and word processors, make the computer valuable to users. The same is true for networks—they are primarily used to send application-related data between systems. Files provide a storage mechanism for application data, configuration settings, and logs. From a data analysis perspective, applications bring together files, operating systems, 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 data analysis. The section also provides guidance on acquiring and examining 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. Besides code, many applications also have one or more of the following additional components: configuration settings, authentication, logs, data, and supporting files. Sections 7.1.1 through 7.1.5 describe these components in detail, and Section 7.1.6 discusses the major types of application architectures, which relate to the intended distribution of the major components.

7.1.1 Configuration Settings

Most applications allow users or administrators to customize certain aspects of the application's behavior by altering configuration settings. From a data analysis perspective, many settings are usually trivial (e.g., specifying background colors), but others may 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.** Applications may store settings in a text file or a file with a proprietary binary format.[94] Some applications require the configuration file to be on the same host as the application, while other applications allow configuration files to be located on other hosts. For example, an application may be installed on a workstation, but the configuration file for a particular user could be stored on the user's home directory on a file server.

+ **Runtime Options.** Some applications permit certain configuration settings to be specified at runtime through the use of command-line options. For example, the Unix e-mail client **mutt** has options for specifying the location of the mailbox to open and the location of the configuration file. Identifying which options are being used for an active session is OS and application-specific; possible methods include reviewing the list of active OS processes, examining an OS history file, and reviewing an application log. Runtime options could also be specified in icons, startup files, batch files, and other ways.

+ **Added to Source Code.** Some applications that make source code available (e.g., open source applications, scripts) actually place user or administrator-specified configuration settings directly into the source code. If the application is then compiled (e.g., converted from human-readable code to a binary, machine-readable format), the configuration settings may actually be contained

[94] For example, on Windows systems, many configuration settings are stored in the Windows registry, which is essentially a set of large configuration files.

within executable files, potentially making the settings far more difficult to access than if they were specified in configuration files or runtime options.

7.1.2 Authentication

Some applications verify the identity of each user attempting to run the application. Although this is usually done to prevent unauthorized access to the application, it may also be done when access is not a concern so that the application can be customized based on the user's identity. 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 operating system.

+ **Pass-Through Authentication.** Pass-through authentication refers to passing operating system credentials (typically, username and password) unencrypted from the operating system 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. For example, if all hosts using an application are part of the same Windows domain, and each user has already been authenticated by the domain, then the application could extract the OS-authenticated identity from each workstation's environment. The application could restrict access to the application by tracking which users are permitted access and comparing the OS-authenticated identity to the authorized user list. This technique is only effective if users cannot alter the user identity in the workstation environment.

Authentication implementations vary widely among environments and applications, and it is beyond the scope of this document to discuss the details. Analysts should be aware that there are many different ways in which users can be authenticated; accordingly, the sources of user authentication records may vary greatly among applications and application implementations. Analysts should also know that some applications use access control (typically enforced by the operating system) to restrict access to certain types of information or application functions. This can be helpful in determining what a particular application user could have done. Some applications record information related to access control, such as failed attempts to perform sensitive actions or access restricted data.

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. Examples of actions that might be recorded are establishing a connection to another system and issuing administrator-level commands. Event log entries might also include supporting information, such as what username was used to perform each action and what status code was returned (which provides more information on the result than a simple successful/failed status).

- **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.[95] Applications may use audit capabilities built into the operating system or provide their own auditing capabilities.
- **Error.** Some applications create error logs, which record information regarding application errors, typically with timestamps. Error logs are helpful in troubleshooting both operational issues and attacks. Error messages can be helpful in determining when an event of interest occurred and identifying some characteristics of the event.
- **Installation.** Applications may create a separate installation log file that records information pertinent to the initial installation and subsequent updates of an application. Information recorded in an installation log varies widely, but is likely to include the status of various phases of the installation, and 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. Debugging records are often very cryptic and may only have meaning to the software's creator, who can decipher error codes and other facets of the records. If an application offers a debugging capability, typically it is only enabled if administrators or developers need to resolve a specific operational problem.

7.1.4 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. For example, an e-mail client allows a user to create an e-mail message and send it to someone, and to receive, view, and delete an e-mail message from someone else. 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. Data may also be stored in databases, which are highly structured collections of files and data specifications. 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 operating systems 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 they are not of value for data analysis. Types of supporting files include the following:

- **Documentation.** This may include administrator and user manuals, help files, and licensing information. Documentation can be helpful to analysts in many ways, such as explaining what the application does, how the application works, and what components the application has. Documentation also typically contains contact information for the vendor of the application; the

[95] Some applications record logon attempts to a separate authentication log. Section 7.1.2 contains additional information on authentication.

vendor may be able to answer questions and provide other assistance in understanding the application.

- **Links.** Also known as *shortcuts*, *links* are simply a pointer to something else, such as an executable. Links are most frequently used on Windows systems; for example, the items listed on the Start menu are really links to programs. By examining the properties of a link, an analyst can determine what program the link runs, where the program is, and what options (if any) are set.
- **Graphics.** This may include standalone graphics used by the application, as well as graphics for icons. Although application graphics are typically of little interest to an analyst, icon graphics may be of interest in attempting to identify which executable was running.

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. Local applications are unlikely to perform authentication. Application data may be contained on the user's system or another system (e.g., file server) and usually cannot be modified simultaneously by multiple users. Examples of local applications are text editors, graphics editors, and office productivity suites (e.g., word processor, spreadsheet).
- **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. Logs are most likely stored on the workstations only. A three-tiered client/server application separates the user interface from the rest of the application, and also separates the data from the other components. The classic three-tier model places the user interface code on the client workstation (along with some supporting files), the rest of the application code on an application server, and the data on a database server. Many Web-based applications use four-tier models: Web browser, Web server, application server, and database server. Each tier only interacts with the adjacent tiers, so in three and four-tier models, the client does not directly interact with the database server. Examples of typical client/server applications are medical records systems, e-commerce applications, and inventory systems.
- **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 on other clients; this information is then used to establish direct connections that do not need to go through the centralized server. Examples of peer-to-peer applications are certain file sharing, chat, and instant messaging programs. Some of these programs are popularly referred to as peer-to-peer but are actually client/server, because the clients communicate with a centralized server, instead of communicating directly with each other.

Most applications are quite flexible in terms of architecture. For example, many client/server applications can have multiple tiers installed on a single system. Especially during application demos or testing, all tiers might be installed on one system. On the other hand, some local applications can be split among systems, with some components on local systems and some on remote systems. Applications often make it easy to specify where different components should be installed and where data and configuration files should be stored. For many applications, there can be a great deal of variety among deployments.

Applications that are designed to split their code among multiple hosts typically use application protocols for communications between hosts.[96] Ubiquitous types of applications such as e-mail and Web use well-known, standardized application protocols to facilitate interoperability among different components. For example, nearly every e-mail client program is compatible with nearly every e-mail server program because they are based on the same application protocol standards. However, a program based on a standard may add proprietary features or violate the standard in some way, especially if the standard is not exhaustively detailed. If interoperability with other applications is not a concern (or not desirable) and the same party is creating all application components, non-standard protocols are often used.

As described throughout Section 7.1, applications may have many different components that operate together. In addition, an application may be dependent on one or more other applications. For example, many e-commerce application clients run within Web browsers. Many applications also rely on OS services, such as printing and DNS lookups (to find the IP addresses of application servers and other devices). Applications vary widely in complexity, from a simple utility program such as a calculator to a large e-commerce application that may involve many thousands of components and have millions of users.

7.2 Types of Applications

Applications exist for nearly every purpose imaginable. Although data analysis techniques can be applied to any application, certain types of applications are more likely to be the focus of 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. The following sections describe each of these types of applications in more detail.

7.2.1 E-mail

E-mail has become the predominant means for people to communicate electronically. 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 memo 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 only display a few header fields for each message: the sender 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:[97]

+ 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.

E-mail client applications are used to receive, store, read, compose, and send e-mails. Most e-mail clients also provide an address book that can hold contact information, such as e-mail addresses, names, and

[96] Applications that are designed to keep all code on a single host typically do not need to use any application protocols.
[97] Most e-mail clients have a configuration setting that specifies whether full or partial e-mail headers should be displayed.

phone numbers. Encryption programs are sometimes used in conjunction with e-mail clients to encrypt an e-mail's body and/or attachments.

When a user sends an e-mail, it is transferred from the e-mail client to the server using SMTP. If the sender and recipient of the e-mail use different e-mail servers, the e-mail is then routed using SMTP through additional e-mail servers until it reaches the recipient's server. Typically, the recipient uses an e-mail client on a separate system to retrieve the e-mail using Post Office Protocol 3 (POP3) or Internet Message Access Protocol (IMAP); in some cases, the e-mail client may be on the destination server (e.g., a multi-user Unix system). The destination server often performs checks on the e-mails before making them available for retrieval, such as blocking messages with inappropriate content (e.g., spam, virus). 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.[98]

7.2.2 Web Usage

Through Web browsers, people access Web servers that contain nearly any type of data imaginable. Many applications also offer Web-based interfaces, which are also accessed through Web browsers. Because they can be used for so many purposes, Web browsers are one of the most commonly used applications. The basic standard for Web communications is the HyperText Transfer Protocol (HTTP); however, HTTP can contain many types of data in a variety of standard and proprietary formats. HTTP is essentially just the mechanism for transferring data between the Web browsers and Web servers.[99]

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 that they receive. Data often logged by Web servers for each request include 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 success, the status code explains what action was performed; for failure, the status code explains why the request failed.

Besides Web browsers and servers, several other types of devices and software might also log related information. For example, Web proxy servers and application proxying firewalls might perform detailed logging of HTTP activity, with a similar level of detail to Web server logs.[100] 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 may find useful data in the services' logs, particularly regarding denied Web requests.

[98] For more information on e-mail services, see NIST SP 800-45, *Guidelines on Electronic Mail Security*, available for download from http://csrc.nist.gov/publications/nistpubs/index.html.

[99] For a detailed description of the current HTTP standard, see RFC 2616, *Hypertext Transfer Protocol—HTTP/1.1*, available at http://www.ietf.org/rfc/rfc2616.txt. Also, see NIST SP 800-44, *Guidelines on Securing Public Web Servers*, for additional information on Web services; it is available for download from http://csrc.nist.gov/publications/nistpubs/index.html.

[100] Typically, proxies cannot log the details of SSL or TLS-protected HTTP requests, because the requests and the corresponding responses pass through the proxy encrypted, which conceals their contents.

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**. Group chat applications provide virtual meeting spaces where many users can share messages at once. Group chat applications typically use a client/server architecture. The most popular group chat protocol, Internet Relay Chat (IRC), is a standard protocol that uses relatively simple text-based communications.[101] IRC also provides a mechanism for users to send and receive files.

+ **Instant Messaging Applications**. Instant Messaging (IM) applications are either peer-to-peer, allowing users to send text messages and files directly to each other, or client/server, passing messages and files through a centralized server. IM application configuration settings may contain user information, lists of users that the user has communicated with, file transfer information, and archived messages or chat sessions. There are several major Internet-based IM services, each of which uses its own proprietary communications protocols. Several companies also offer enterprise IM products that are run within an organization. Such products are often integrated to some extent with the organization's e-mail services and can be used only by authenticated e-mail users.

+ **Audio and Video.** As the capacity of networks continues to increase, conducting real-time video and audio communications across computer networks also becomes more common. Technologies such as Voice over IP (VoIP) permit people to conduct telephone conversations over networks such as the Internet.[102] Some audio implementations provide computer-based service from end to end, while others are only partially computer-based, with an intermediate server converting the communications between computer networks and standard phone networks. Many audio technologies are primarily peer-to-peer applications. Video technologies can be used to hold teleconferences or have "video phone" communications between two individuals. Commonly used protocols for audio and video communications include H.323 and Session Initiation Protocol (SIP).[103]

7.2.4 File Sharing

Users can share files using many different programs. As described earlier in this section, e-mail, group chat programs, and IM software all offer the ability to send and receive particular files. However, these programs generally do not allow a recipient to browse through files and choose the files to transfer. Full-fledged file sharing programs and protocols are needed for this level of functionality. 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 (if required), reviewing the list of available files (if needed), then transferring files to or from the server. Examples of commonly used client/server file sharing services are File Transfer Protocol (FTP), Network File Sharing (NFS), Apple Filing Protocol (AFP), and Server

[101] The original standard for IRC is documented in RFC 1459, *Internet Relay Chat Protocol*, available at http://www.ietf.org/rfc/rfc1459.txt. RFCs 2810 through 2813 contain additional information that supplements RFC 1459.

[102] For more information on VoIP, see NIST SP 800-58, *Security Considerations for Voice Over IP Systems*, available at http://csrc.nist.gov/publications/nistpubs/index.html.

[103] The standard for SIP is available as RFC 3261, *SIP: Session Initiation Protocol*, located at http://www.ietf.org/rfc/rfc3261.txt.

Message Block (SMB).[104] These are standardized protocols that do not protect the confidentiality of the data in transit, including any supplied authentication credentials such as passwords. Secure alternatives, such as Secure FTP (SFTP) and Secure Copy (scp), encrypt their network communications. Most operating systems have built-in file sharing clients (e.g., FTP, SMB), but users can also install various third-party programs that provide similar functionality.

+ **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, where 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). Users typically can choose from several client programs when using a particular service. Although most services allow each user to control which files are shared on their system, services known as encrypted peer-to-peer work by storing others' files on an encrypted portion of each user's hard drive, and giving users no control over or knowledge of what is stored in that area of their own systems. Anonymous peer-to-peer services send requested files through multiple intermediate hosts instead of simply sending them from source to destination, with the goal of making it very difficult to identify the true source or destination of any given file.

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. The class of software used for creating, viewing, and editing such documents is known as office productivity applications. This includes word processor, spreadsheet, presentation, and personal database software. Documents often have user or system information embedded into 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.[105]

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. Examples of commonly used security applications include 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 may 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

[104] More information on SMB is available from http://samba.anu.edu.au/cifs/docs/what-is-smb.html.

[105] One example of an office productivity application that might capture user or system information within document is Microsoft Office. More information on this is available from Microsoft Knowledge Base article 834427, located at http://support.microsoft.com/default.aspx?scid=kb;en-us;834427.

include file encryption utilities, steganographic tools, and system cleanup tools. System cleanup tools are special-purpose software that remove data pertaining to particular applications, such as Web browsers, as well as data in general locations, such as temporary directories. 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 effects of the tools.

7.3 Acquiring Application Data

As described in Section 7.1, application-related data may be located within filesystems, volatile OS data, and network traffic. Sections 4.2, 5.2, and 6.3 contain specific information on acquiring data from these respective sources. The following lists the types of application data that each source may contain:

+ **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 on 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 users connecting to a remote application, and application components on different systems communicating with each other. 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 acquired. In many cases, the analyst must first decide which application is of interest. For example, it is common to have multiple Web browsers and e-mail clients installed on a single system. If analysts are asked to acquire data regarding an individual's use of the organization's e-mail services, they need to be mindful of all the ways in which the individual could have accessed those services. The user's computer could contain three different e-mail clients, plus two Web browsers that could be used to access a Web-based e-mail client provided by the organization. For the user's computer, analysts can simply acquire all data from it and then determine during the examination process which clients were actually used for e-mail. However, there are many potential data sources besides the user's computer, and these sources may vary based on the client that was used. For example, use of the Web-based client may have been recorded in Web server, firewall, IDS, and content monitoring software logs, as well as Web browser history files, Web browser caches, cookies, and personal firewalls. In some situations, acquiring the necessary data may involve identifying all components of the application, deciding which are most likely to be of interest (based on the details of the situation and the need), finding the location of each component, and acquiring data from those components. Section 8 contains examples that illustrate the complexity of identifying components and prioritizing data acquisition for applications.

7.4 Examining Application Data

Examining 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, 5.3, and 6.4, respectively. The examination may be hindered if the application is custom, such as a program written by the user; the analyst is unlikely to have any knowledge of such an application. Another possible issue during examination involves the use of application-based security controls, such as data encryption and passwords. Many applications use such security controls to thwart unauthorized access to sensitive data by authorized users.

In some cases, analysts need to bring together the 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 requires a skilled and knowledgeable analyst who understands the information presented by all the sources. The analyst can review the results of the examination of individual application data sources and see how the information fits together. Tools that may be helpful to analysts include security event management software (as described in Section 6.2.5), which could correlate some application-related events among multiple data sources, and log analysis software (including some types of host-based intrusion detection software), which could be run against certain types of logs to identify suspicious activity. Section 8 provides examples of how data from multiple types of sources can be correlated through analysis to get a more accurate and comprehensive view of what occurred.

7.5 Recommendations

The key recommendations presented in this section for using data from applications are summarized below.

+ **Analysts should consider all the possible application data sources.** Application events may be recorded by many different data sources. Also, 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 bring together application data from various sources.** The analyst should review the results of the examination 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

Sections 4 through 6 describe the acquisition and examination of data from three data source categories: data files, operating systems, and network traffic. The techniques and processes for acquiring and examining the data in these categories are fundamentally different. Section 7 describes the acquisition and examination of application data, which brings together the three data source categories. For example, many applications use data files, alter the configuration of operating systems, and generate network traffic. Many situations, such as computer security incidents, can be handled most effectively by analyzing multiple types of data sources and correlating events across the sources.

This section of the guide presents two examples of how multiple data sources can be used together during an analysis. Each example describes a scenario, indicates a specific need for data analysis, and presents an explanation of how the analysis process might be performed. The explanations also illustrate how complex the analysis 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. Their network intrusion detection systems have recently reported several unusual alerts involving the server, and an analyst who reviews 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. Accordingly, 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 identify the distinguishing characteristics of the worm. This information is critical to the incident response team, so that they can act effectively to perform containment, eradication, and recovery activities, as well as preventing 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, then the characteristics identified by the handler 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 may 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-Based IDS.** Because the initial reports of the incident were generated by network IDS sensors, it is very likely that the network IDS data contains information on the basic characteristics of the network activity. At a minimum, the data should indicate which server was attacked and on what port number, which indicates which network service was targeted. Identifying the service is very important for identifying the exploited vulnerability and developing a mitigation strategy to prevent similar incidents from occurring on other systems. From an analysis standpoint, knowing the targeted service and port number is also important because the information can be used to identify other likely data sources and query them for relevant information. Some network IDS deployments may record additional useful information, such as application data (e.g., Web requests and responses, e-mail headers and file attachment names). The application data may contain words, phrases, or other character sequences that are associated with a particular worm.

- **Network-Based Firewall.** Firewalls are typically configured to log blocked connection attempts, which includes the intended destination IP address and port. Accordingly, firewalls may have records of worm activity that they blocked. Some worms attempt to exploit multiple services or service ports; firewall records may show that a worm actually tries to establish connections to at least four different port numbers, but that the firewall blocks connections using three of the ports. This information could be useful in identifying the worm. If firewalls are configured to record permitted connections, then their logs may show which hosts within the organization have received worm traffic or been infected and generated their own worm traffic. This is particularly useful for situations where network IDS sensors do not monitor all traffic that reaches the firewall. Other perimeter devices that the worm traffic may have passed through, such as routers, VPN gateways, and remote access servers, may record information similar to that logged by network-based firewalls.

- **Host-Based IDS and Firewall.** IDS and firewall products running on the infected system may contain more detailed information than network-based IDS and firewall products. For example, a host-based IDS can identify changes to files or configuration settings on the host that were performed by a worm. This information is helpful not only in planning containment, eradication, and recovery activities by determining how the worm has affected the host, but also in identifying which worm infected the system. However, because many worms disable host-based security controls and destroy log entries, data from host-based IDS and firewall software may be limited or missing. If the software was configured to forward copies of its logs to centralized log servers, then queries to those servers may provide some information.

- **Antivirus Software.** Because the threat reached the server and successfully breached it, it is unlikely that network or host-based antivirus software contains any record of it. If antivirus software had detected the worm, it should have stopped it. However, it is possible that the antivirus software saw the worm but somehow failed to stop it, or that the antivirus software has been updated since the infection with new signatures that can recognize the worm. The incident handler could also scan the server for worms using a current version of antivirus software from a trusted toolkit.

- **Application Logs.** If the worm uses a common application protocol, such as HTTP or SMTP, information regarding it may be recorded in several places, such as application server logs, proxy servers, and application-specific security controls. Less common application protocols probably have information only in the application server logs. Application logs may record extensive details on the application-specific characteristics of the activity, and are particularly helpful at identifying attack characteristics from less common applications.

The goal in the initial information gathering effort is to identify enough characteristics so that the worm can be identified positively. This can be challenging, particularly for worms that have dozens of variants; these variants often have similar characteristics but cause different effects to 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.[106] Virtually any instance of malware, other than the latest threats (e.g., released in the past several hours), should 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 to prevent infections on other systems.

If a search of malware databases does not lead to the identification of the worm, then the incident handler may need to perform additional research and analysis to determine the information normally 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 timeframe for the vendor's response is unknown. To gather more information, the analyst can examine the infection through the following two methods:

+ **Current State of the Host.** The analyst can look at several different aspects of the host's current state. In this case, it is probably most effective to start by examining 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 could 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 that the analyst can then 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. Also, worms often install backdoors and other tools onto systems that permit attackers to gain remote access to infected systems, which could lead to additional damage. Accordingly, organizations may choose to disconnect infected systems from networks immediately, instead of performing an analysis of the host first. This may make it considerably more difficult for analysts to identify a worm and determine its effects on systems—for example, network activity and certain aspects of the host state would not be available. The analyst may need to perform a more detailed analysis of the server, such as acquiring 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 operating system, 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 so that the incident response team can act effectively to contain, eradicate, and recover from the incident.

[106] Malware databases are maintained by several vendors, including Computer Associates (http://www3.ca.com/securityadvisor/virusinfo/default.aspx), F-Secure (http://www.europe.f-secure.com/virus-info/), Network Associates (http://vil.nai.com/vil/default.asp), Sophos (http://www.sophos.com/virusinfo/analyses/), Symantec (http://securityresponse.symantec.com/avcenter/vinfodb.html), and Trend Micro (http://www.trendmicro.com/vinfo/virusencyclo/).

8.2 Threatening E-mail

An incident handler responds to a request for assistance with a malware incident. An employee's system has become infected, apparently through an e-mail that claims to have been sent from another employee's e-mail account 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 in determining the true source of 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.

The first piece of information to review is the e-mail's header. It should contain the domain name and 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.[107] 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. 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 may still be stored in both locations.

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

- **Local E-mail Client.** In that case, the incident handler should be able to use network records, such as DHCP logs, to identify the desktop, laptop, PDA, or other device that was used to send the e-mail. The incident handler can examine the device to look for malware and for records related to the e-mail. For example, the e-mail client might be configured to keep a copy of each e-mail that it sends, or the user may have saved drafts of the e-mail message. If the message cannot be found intact on the system, acquiring data from the device's memory and filesystems, including deleted and temporary files, might lead to the identification of fragments of the e-mail. Also, 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 also examine the authentication records on the host to determine which user account was in use at the time 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 could 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 could then determine which system was assigned that IP address at the time and examine the identified system for the malware and the e-mail. For example, temporary files from the Web browser could contain a copy of the e-mail's contents.

[107] Within the e-mail header, these records are stored in reverse order, so the most recent record occurs first and the least recent record occurs last. If an e-mail has been forged, the false records are the least recent, so they should appear last in the header.

- **Spoofed.** If the IP address was fabricated—for example, it is not a valid address within the organization's networks—then the incident handler needs to rely on other data sources to attempt 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 earlier, it is possible that the final e-mail server in the list contains a copy of the e-mail. Backups of that server might contain a copy of the e-mail only if it had been held there for delivery for several hours or more. Other services associated with e-mail, such as antivirus software and spam filters, might 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, recreate the sequence of events from a cyber perspective. For example, a user logged on to a particular desktop computer at 8:37 a.m. At 10:02 a.m., the malware 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 at 11:23 a.m., including the malware e-mail. This information can then be used to identify and eradicate all copies of the malware e-mail, as well as determine which hosts may need to have additional recovery procedures performed.

8.3 Recommendations

The key recommendations presented in this section for using data from multiple sources are summarized below.

- **Analysts can handle many situations most effectively by analyzing individual data sources and then correlating events among them.** The techniques and processes for acquiring and examining different types of data sources are fundamentally different. Many applications have data captured in data files, operating systems, 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 together from many data sources in a single place.

This page has been left blank intentionally.

Appendix A—Recommendations

Appendix A lists the major recommendations presented in Sections 2 through 8 of this document. The first group of recommendations applies to organizing a data analysis capability. The remaining recommendations have been grouped by the phases of the data analysis process: acquisition, examination, utilization, and review.

A.1 Organizing a Data Analysis Capability

+ **Organizations should have some capability to perform computer and network data analysis.** Data analysis can assist with various tasks within an organization, including reconstructing computer security incidents, troubleshooting operational problems, and recovering from accidental system damage.

A.1.1 Data Analysis Participants

+ **Organizations should determine which parties should handle each aspect of data analysis.** Most organizations rely on a combination of their own staff and external parties to perform data analysis tasks. Organizations should decide which parties should take care of which tasks based on skills and abilities, cost, response time, and data sensitivity.

+ **Data analysts should have solid technical knowledge.** Because current tools have rather limited analysis abilities, analysts need to 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 data analysis capabilities.** More than one team member should be able to perform each typical data analysis activity. Hands-on exercises and IT and data analysis 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 data analysis.** Individuals performing data analysis actions should be able to reach out to other teams and individuals within an organization as needed for additional assistance. Examples of teams include IT professionals, management, legal advisors, auditors, and physical security staff. Members of these teams should understand their roles and responsibilities for data analysis, receive training and education on data analysis-related policies and procedures, and be prepared to cooperate with and assist others on data analysis actions.

A.1.2 Data Analysis Policies and Procedures

+ **Data analysis considerations should be clearly addressed in policies.** At a high level, policies should allow authorized personnel to monitor user activities and perform investigations for legitimate reasons under appropriate circumstances. Organizations may also have a separate data analysis policy for incident handlers and others with data analysis roles that provides more detailed rules for appropriate behavior. Everyone who may be called upon to assist with any data analysis efforts should be familiar with and understand the data analysis policy. Additional policy considerations are as follows:

  – Data analysis policy should clearly define the roles and responsibilities of all people performing or assisting with the organization's data analysis activities. The policy should include all internal and external parties that may be involved, and it should clearly indicate who should contact which parties under different circumstances.

– The organization's policies and procedures should clearly explain what data analysis actions should and should not be performed under normal and special circumstances, and also address the use of anti-forensic tools and techniques. Policies and procedures should also address the handling of inadvertent exposures of sensitive information.

– Incorporating data analysis considerations in the information system life cycle can lead to more efficient and effective handling of many incidents.

– The organization's policies should address the long-term storage of sensitive information captured by data analysis tools, and should ensure that this does not violate the organization's data retention policies.

– The organization's policies should also address the monitoring of networks, as well as requiring warning banners on systems that indicate activity may be monitored.

+ **Organizations should create and maintain procedures for performing data analysis tasks.** The procedures should include general methodologies for investigating an incident using data analysis techniques, and possibly step-by-step procedures for performing routine tasks. The procedures should be reviewed regularly and maintained so that they are accurate.

A.1.3 Technical Preparation

+ **Analysts should have a trusted toolkit for data acquisition and examination.** It should contain various tools that provide the ability to acquire 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 an analysis 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, 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.

A.2 Performing the Data Analysis Process

+ **Organizations should perform data analysis using a consistent process.** The data analysis process presented in this guide uses a four-phase process: acquisition, examination, utilization, and review. The exact details of the phases may vary based on the need for data analysis.

A.2.1 Data Acquisition

+ **Organizations should be proactive in collecting useful data.** Configuring auditing on operating systems, implementing centralized logging, performing regular system backups, and using security monitoring controls can all generate sources of data for future data analysis efforts.

+ **Analysts should be aware of the range of possible data sources.** Analysts should be able to survey a physical area and recognize the 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 acquire data from a primary source.

- **Analysts should consider all the possible application data sources.** Application events may be recorded by many different data sources. Also, 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 acquisition using a standard process.** The recommended steps are developing a plan to acquire the data, acquiring the data, and verifying the integrity of the data. Analysts should create a plan that prioritizes 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.

- **Analysts should act appropriately to preserve volatile OS data.** The criteria for determining whether volatile OS data needs to be preserved should be documented in advance so that analysts can make informed decisions as quickly as possible. The risks associated with collecting volatile OS data should be weighed against the potential for recovering important information to determine if the effort is warranted.

- **Analysts should use a trusted toolkit for acquiring volatile OS data.** Doing so allows accurate OS data to be collected while causing the least amount of disturbance to the system and protecting the tools from changes. The analyst should know how each tool should affect or alter the system when acquiring data.

- **Analysts should choose the appropriate shutdown method for each system.** Each way of shutting down a particular operating system can cause different types of data to be preserved or corrupted, so analysts should be aware of the typical shutdown behavior of each OS.

- **Analysts should preserve and verify file integrity.** Using a write-blocker during a backup 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 should be accessed as read-only whenever possible; write-blockers can also be used to prevent writes to the image file or restored image.

A.2.2 Examination

- **Analysts should use a methodical approach.** The foundation of computer and network data analysis is using a methodical approach to draw conclusions based on the available data, or determine that no conclusion can yet be drawn.

- **Analysts should work with copies of files, not the original files.** During the acquisition phase, the analyst should make one or more copies of the desired files or filesystems. The analyst can then work with a copy of the files without affecting the originals. A physical backup should be performed if preserving file times is important. A logical backup is sufficient for informal file acquisition from live systems.

- **Analysts should consider the fidelity and value of each data source.** Analysts should have more confidence in original data sources than data sources that receive normalized data from other sources. Analysts should validate any unusual or unexpected data that is based on analyzing or interpreting data, such as IDS and SEM alerts.

- **Analysts should rely on file headers 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 definitively identify the type of data stored in many files by looking at their file headers.
- **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, 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 together from many data sources in a single place.

A.2.3 Utilization

- **Analysts should bring together application data from various sources.** The analyst should review the results of the examination of individual application data sources and determine how the information fits together, to perform a detailed analysis of application-related events and event reconstruction.
- **Analysts can handle many situations most effectively by analyzing individual data sources and then correlating events among them.** The techniques and processes for acquiring and examining different types of data sources are fundamentally different. Many applications have data captured in data files, operating systems, and network traffic.

A.2.4 Review

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

Appendix B—Scenarios

Tabletop exercises that focus on how data analysis tools and techniques can be used in various scenarios provide an inexpensive and effective way to build and maintain skills and identify problems with procedures and policies. The exercise participants review a brief scenario and are then asked several questions related to the scenario. The participants discuss each question and formulate an answer based on what they would really do in the situation. The response is then compared with the organization's policies, procedures, and guidelines to identify any discrepancies or deficiencies. For example, the answer to a question might indicate that actions would be delayed because a participant lacks a particular piece of software and a particular team within the organization does not provide off-hours support.

Section C.1 contains a list of general questions that could be applied to almost any scenario. Section C.2 contains several sample scenarios, some of which are followed by additional scenario-specific questions. Organizations are encouraged to adapt these questions and scenarios for use in their own exercises.

B.1 Scenario Questions

1. What are the potential sources of data?

2. Of the potential sources of data, which are the most likely to contain helpful information, and why?

3. Which data source would be checked first, and why?

4. Which tools and techniques would most likely be used? Which other tools and techniques might also be used?

5. Which groups and individuals within the organization would probably be involved in the data analysis activities?

6. What communications with external parties might occur, if any?

7. From a data analysis standpoint, what would be done differently if the scenario had occurred at a different day and time (on-hours versus off-hours)?

8. From a data analysis standpoint, what would be done differently if the scenario had occurred at a different physical location (onsite versus offsite)?

B.2 Scenarios

Scenario 1: Possible DDoS Attack

On a Saturday afternoon, external users start having problems accessing the organization's public Web sites. Over the next hour, the problem worsens to the point where nearly every attempt to access any of the organization's public Web sites fails. Meanwhile, a member of the organization's networking staff responds to automatically generated alerts from an Internet border router and determines that much of the organization's Internet bandwidth is being consumed by an unusually large volume of User Datagram Protocol (UDP) packets to and from both the organization's public Domain Name System (DNS) servers.

Scenario 2: Online Payment Problems

Over the course of a week, the number of phone calls coming into the organization's help line for online bill presentment and payment increases by 400%. Most callers complain of having to re-submit payment information multiple times, and many cannot complete their payment.

The following are additional questions for this scenario:

1. The problems could be caused by a non-technical reason, such as a lack of clear instructions for new users. How should the technical and non-technical aspects of the investigation be coordinated and balanced?

2. How might privacy considerations impact the use of data analysis tools and techniques?

3. How would tools and techniques be used if application developers were confident that an operational problem was causing the issues?

Scenario 3: Unknown Wireless Access Point

On a Monday morning, the organization's help desk receives calls from five users on the same floor of a building who state that they are having problems with their wireless access. A network administrator who is asked to assist in resolving the problem brings a laptop with wireless capability to the users' floor. As he views his wireless networking configuration, he notices that there is a new wireless access point listed as available. Based on the insecure configuration settings transmitted by the access point, the administrator does not believe that his team deployed it.

The following are additional questions for this scenario:

1. What types of tools might be used to locate the access point overtly? Covertly?

2. How would the analyst's actions change if it was determined that the access point was deployed for a legitimate business purpose, such as a contractor from outside the organization temporarily working at this office?

3. How would the analyst's actions change if it was determined that an unknown individual was seen deploying the access point?

Scenario 4: Reinfected Host

During the past two weeks, a user has needed to have the same virus removed from a laptop computer twice, and the user is now reporting similar symptoms again. The technical support staff that handled the previous infections confirmed that the antivirus software on the computer was enabled and up to date, and were unable to determine how the virus is reinfecting the computer.

The following are additional questions for this scenario:

1. What other sources of data might be spotted by visually surveying the user's office?

2. What are the most likely possible data sources outside of the user's office?

3. What legal considerations should the analysts be aware of if they want to access and use a data source that the organization does not own?

Scenario 5: Mistaken Identity

Within the past 24 hours, two employees in the organization have reported fraudulent purchases charged to their organization-issued credit cards. The organization frequently buys items from the companies that sold the items in the questioned transactions. A follow-on assessment shows that charges to the organization's credit cards across the organization have increased by 30% in the past three days.

The following are additional questions for this scenario:

1. How could data analysis tools and techniques help to determine what has happened (e.g., individual employees in the organization are victims of identity theft, financial resources have been corrupted)?

2. What privacy concerns should be considered in investigating employees' financial transactions?

Scenario 6: Unwanted Screen Saver

The organization's help desk has just received several calls from users who complain that a screen saver depicting pastoral scenery is activating while they are working at their computers. The screen saver requires each user to submit his or her password to unlock the screen saver and continue work. At the same time, the organization's network intrusion detection systems report several unusual alerts involving a Web server. The data in the alerts indicates that some suspicious activity was directed at the server, and the server is now generating similar activity directed at other systems. The intrusion detection analyst's initial hypothesis is that a worm may have attacked a vulnerable network service on the Web server.

The following are additional questions for this scenario:

1. Given the time-sensitive nature of this scenario, how should analysts prioritize their actions?

2. How would the use of data analysis tools and techniques change if the worm disrupted network communications?

3. How would the use of data analysis tools and techniques change if the infected desktop systems were used to process sensitive information that the organization is required to safeguard?

Scenario 7: Phishing Attempts

In the past 24 hours, several employees have called the help desk to question the validity of e-mails from the organization's official credit card provider. The e-mails cite a possible security breach in the financial institution's records and ask recipients to follow a link to the institution's Web site and create a new password, after identifying themselves by providing their existing passwords and account information.

The following are additional questions for this scenario:

1. Given the time-sensitive nature of this scenario, how should analysts prioritize their actions?

2. What other organization(s) should be contacted to mitigate potential cases of identity theft?

Scenario 8: Encrypted Files

An employee leaves an organization unexpectedly, and the employee's manager is granted access to the former employee's desktop computer to retrieve important project information that should be stored on it.

The manager finds some filenames that appear to be related to the project, but the manager cannot access the contents of the files. A system administrator looks at the system and concludes that the former employee probably encrypted the files.

The following are additional questions for this scenario:

1. Who should determine how much effort should be put into attempting to recover the encrypted data? How would this be determined?

2. What changes could be made to the organization's policies and procedures to reduce the impact of similar future incidents?

Appendix C—Glossary

Selected terms used in the *Guide to Computer and Network Data Analysis: Applying Forensic Techniques to Incident Response* are defined below.

Acquisition: The first phase of the computer and network data analysis process, which involves identifying, collecting, and protecting data related to an event.

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

Cluster: A group of contiguous sectors.

Computer and Network Data Analysis: The identification, collection, analysis, and examination of data from computers and networks without preserving the integrity of all information and maintaining a strict chain of custody.

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

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

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 physical backup.

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

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

Examination: The second phase of the computer and network data analysis process, which involves using data analysis tools and techniques to identify and analyze relevant information from acquired 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.

Filename: A unique name used to reference a file.

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

Forensic Science: The application of science to the law.

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

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 have been formatted with a filesystem.

Message Digest: A digital signature that uniquely identifies data and has the property that changing a single bit in a data stream will yield a completely different message digest.

Metadata: Data within a file header that provides information about the file's contents.

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

Network-Based Intrusion Detection System (IDS): 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 (OS): 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.

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

Process: An executing program.

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, and 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.

Review: The final phase of the computer and network data analysis process, which involves examining recent data analysis activity and identifying policy shortcomings, procedural errors, and other areas for improvement.

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

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

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

Steganography: Embedding data within other data to conceal it.

Subdirectory: A directory contained within another directory.

Utilization: The third phase of the computer and network data analysis process, which involves the preparation and presentation of the results of the examination phase.

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

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

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

This page has been left blank intentionally.

Appendix D—Acronyms

Selected acronyms used in the *Guide to Computer and Network Data Analysis: Applying Forensic Techniques to Incident Response* are defined below.

ADS	Alternate Data Stream
ARIN	American Registry for Internet Numbers
ARP	Address Resolution Protocol
ATA	Advanced Technology Attachment
BIOS	Basic Input/Output System
BMP	Bitmap
CCIPS	Computer Crime and Intellectual Property Section
CD	Compact Disc
CD-R	CD Readable
CD-ROM	CD Read Only Memory
CD-RW	CD Rewriteable
CDFS	CD File System
CFI	Computer & Financial Investigations
CFRDC	Computer Forensics Research and Development Center
CFTT	Computer Forensics Tool Testing
CNF	Computer and Network Forensics
CSD	Computer Security Division
CVE	Common Vulnerabilities and Exposures
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DLL	Dynamic Link Library
DNS	Domain Name System
DoD	Department of Defense
DoS	Denial of Service
DVD	Digital Video Disc or Digital Versatile Disc
DVD-R	DVD Recordable
DVD-ROM	DVD Read Only Memory
DVD-RW	DVD Rewritable
E-mail	Electronic Mail
ESP	Encapsulating Security Payload
ext2fs	Second Extended Filesystem
ext3fs	Third Extended Filesystem
FAT	File Allocation Table
FBI	Federal Bureau of Investigation
F.I.R.E.	Forensic and Incident Response Environment
FISMA	Federal Information Security Management Act
FLETC	Federal Law Enforcement Training Center
FTK	Forensic Toolkit
FTP	File Transfer Protocol

GB	Gigabyte
GUI	Graphic User Interface
HFS	Hierarchical File System
HPFS	High-Performance File System
HTCIA	High Technology Crime Investigation Association
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IACIS	International Association of Computer Investigative Specialists
IATAC	Information Assurance Technology Analysis Center
ICMP	Internet Control Message Protocol
ID	Identification
IDE	Integrated Drive Electronics
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMAP	Internet Message Access Protocol
IOS	Internetwork Operating System
IP	Internet Protocol
IPsec	Internet Protocol Security
IR	Interagency Report
IRC	Internet Relay Chat
IRQ	Interrupt Request Line
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Information Technology
ITL	Information Technology Laboratory
JPEG	Joint Photographic Experts Group
KB	Kilobyte
LACNIC	Latin American and Caribbean IP Address Regional Registry
MAC	Media Access Control
MAC	Modification, Access, and Creation
MB	Megabyte
MD	Message Digest
MDET	MetaData Extraction Tool
MISTI	MIS Training Institute
MMC	Multimedia Card
MO	Magneto Optical
MS-DOS	Microsoft Disk Operating System
NAT	Network Address Translation
NFAT	Network Forensic Analysis Tool
NFS	Network File Sharing
NIC	Network Interface Card
NIJ	National Institute of Justice
NIST	National Institute of Standards and Technology

NSRL	National Software Reference Library
NTFS	Windows NT File System
NTI	New Technologies Inc.
NTP	Network Time Protocol
NW3C	National White Collar Crime Center
OEM	Original Equipment Manufacturer
OIG	Office of Inspector General
OMB	Office of Management and Budget
OS	Operating System
OSR2	OEM Service Release 2
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association
PDA	Personal Digital Assistant
POP	Post Office Protocol
RAID	Redundant Arrays of Inexpensive Disks
RAM	Random Access Memory
RAS	Remote Access Server
RCFL	Regional Computer Forensics Laboratory
RFC	Request for Comment
RIPE NCC	Réseaux IP Européens Network Coordination Centre
SAM	Security Account Manager
SCP	Secure Copy
SCSI	Small Computer System Interface
SD	Secure Digital
SDMI	Secure Digital Music Initiative
SEM	Security Event Management
SFTP	Secure FTP
SIP	Session Initiation Protocol
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SP	Special Publication
SSH	Secure Shell
SSL	Secure Sockets Layer
TB	Terabytes
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TCT	The Coroner’s Toolkit
TUCOFS	The Ultimate Collection of Forensic Software
UDF	Universal Disk Format
UDP	User Datagram Protocol
UFS	Unix File System
UPS	Uninterruptible Power Supply
URL	Uniform Resource Locator
USB	Universal Serial Bus

VoIP	Voice Over IP
VPN	Virtual Private Network

Appendix E—Print Resources

Bejtlich, Richard, *The Tao of Network Security Monitoring: Beyond Intrusion Detection*, Addison-Wesley, 2004.

Carrier, Brian, *File System Forensic Analysis*, Addison-Wesley, 2005.

Casey, Eoghan, *Digital Evidence and Computer Crime*, Academic Press, 2004.

Casey, Eoghan, *Handbook of Computer Crime Investigation: Forensic Tools & Technology*, Academic Press, 2001.

Davis, Chris et al, *Hacking Exposed: Computer Forensics Secrets & Solutions*, McGraw-Hill Osborne Media, 2004.

Farmer, Dan and Venema, Wietse, *Forensic Discovery*, Addison-Wesley, 2004.

Kruse II, Warren G. and Heiser, Jay G., *Computer Forensics: Incident Response Essentials*, Addison-Wesley, 2001.

Lucas, Julie and Moeller, Brian, *The Effective Incident Response Team*, Addison-Wesley, 2004.

Orebaugh, Angela, *Ethereal Packet Sniffing*, Syngress, 2004.

Oseles, Lisa, “Computer Forensics: The Key to Solving the Crime”*,* October 2001. http://faculty.ed.umuc.edu/~meinkej/inss690/oseles_2.pdf

Prosise, Chris et al , *Incident Response and Computer Forensics*, *Second Edition*, McGraw-Hill Osborne Media, 2003.

Schiffman, Mike et al, *Hacker's Challenge 2: Test Your Network Security & Forensic Skills*, McGraw-Hill Osborne Media, 2002.

Schweitzer, Douglas, *Incident Response: Computer Forensics Toolkit*, Wiley, 2003.

Zalewski, Michal, *Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks*, No Starch, 2005.

This page has been left blank intentionally.

Appendix F—Online Tools and Resources

The lists below provide examples of online tools (particularly free/open source) and resources that may be helpful in performing computer and network data analysis.

Organizations

Organization	URL
Computer Crime and Intellectual Property Section (CCIPS), U.S. Department of Justice	http://www.cybercrime.gov/
Computer Forensics Research and Development Center (CFRDC)	http://www.ecii.edu/edu_center.html
Department of Defense (DoD) Computer Forensics Laboratory	http://www.dcfl.gov/dcfl/dcfl.htm
Federal Bureau of Investigation (FBI)	http://www.fbi.gov/
Florida Association of Computer Crime Investigators	http://www.facci.org/
High Technology Crime Investigation Association (HTCIA)	http://www.htcia.org/
International Association of Computer Investigative Specialists (IACIS)	http://www.cops.org/
National White Collar Crime Center (NW3C)	http://www.cybercrime.org/
Regional Computer Forensics Laboratory (RCFL)	http://www.rcfl.gov/

Technical Resource Sites

Resource Name	URL
Computer Crime Research Center	http://www.crime-research.org/
Computer Forensics Links (compiled by Dave Dittrich)	http://staff.washington.edu/dittrich/
Computer Forensics Links and Whitepapers	http://www.forensics.nl/links/
Computer Forensics Tool Testing (CFTT) Project	http://www.cftt.nist.gov/
Digital Mountain Technical and Legal Resources	http://www.digitalmountain.com/TechnicalResources.htm
The Electronic Evidence Information Center	http://www.e-evidence.info/
Forensic Focus – Billboard and Links	http://www.forensicfocus.com/
National Institute of Justice (NIJ) Electronic Crime Program	http://www.ojp.usdoj.gov/nij/topics/ecrime/welcome.html
National Software Reference Library (NSRL)	http://www.nsrl.nist.gov/
Technology Pathways Resource Center	http://www.techpathways.com/DesktopDefault.aspx?tabindex=8&tabid=14
Wotsit's Format	http://www.wotsit.org/

Training Resources

Training Resource Name	URL
CompuForensics	http://www.compuforensics.com/training.htm
Computer Forensic Services	http://www.computer-forensic.com/training.html
Computer Forensics Training Center Online	http://www.cftco.com/
Federal Law Enforcement Training Center (FLETC), Computer & Financial Investigations (CFI) Division	http://www.fletc.gov/cfi/index.htm
Foundstone	http://www.foundstone.com/
IACIS	http://www.cops.org/html/training.htm
InfoSec Institute	http://www.infosecinstitute.com/courses/computer_forensics_training.html
MIS Training Institute (MISTI)	http://www.misti.com/
New Technologies Inc. (NTI)	http://www.forensics-intl.com/training.html
NW3C	http://www.cybercrime.org/courses.html
SANS Institute	http://www.sans.org/

Other Technical Resource Documents

Resource Name	URL
Basic Steps in Forensic Analysis of Unix Systems, by Dave Dittrich	http://staff.washington.edu/dittrich/misc/forensics/
Computer Forensics: Introduction to Incident Response and Investigation of Windows NT/2000, by Norman Haase	http://www.sans.org/rr/whitepapers/incident/647.php
Electronic Crime Scene Investigation: A Guide for First Responders	http://www.ojp.usdoj.gov/nij/pubs-sum/187736.htm
Evidence Seizure Methodology for Computer Forensics, by Thomas Rude	http://www.crazytrain.com/seizure.html
Forensic Analysis of a Live Linux System, by Mariusz Burdach	http://www.securityfocus.com/infocus/1769 (part one), http://www.securityfocus.com/infocus/1773 (part two)
How to Bypass BIOS Passwords	http://labmice.techtarget.com/articles/BIOS_hack.htm
NIST Interagency Report (IR) 7100, *PDA Forensic Tools: An Overview and Analysis*	http://csrc.nist.gov/publications/nistir/index.html
NIST SP 800-31, *Intrusion Detection Systems*	http://csrc.nist.gov/publications/nistpubs/index.html
NIST SP 800-44, *Guidelines on Securing Public Web Servers*	http://csrc.nist.gov/publications/nistpubs/index.html
NIST SP 800-45, *Guidelines on Electronic Mail Security*	http://csrc.nist.gov/publications/nistpubs/index.html
NIST SP 800-61, *Computer Security Incident Handling Guide*	http://csrc.nist.gov/publications/nistpubs/index.html
NIST SP 800-72, *Guidelines on PDA Forensics*	http://csrc.nist.gov/publications/nistpubs/index.html
NIST SP 800-83 (DRAFT), *Guide to Malware Incident Prevention and Handling*	http://csrc.nist.gov/publications/drafts.html
An Overview of Steganography for the Computer Forensic Examiner, by Gary Kessler	http://www.fbi.gov/hq/lab/fsc/backissu/july2004/research/2004_03_research01.htm
RFC 3164: *The BSD Syslog Protocol*	http://www.ietf.org/rfc/rfc3164.txt

Resource Name	URL
RFC 3227: *Guidelines for Evidence Collection and Archiving*	http://www.ietf.org/rfc/rfc3227.txt

Web Sites with Data Analysis Software Listings

Software Type	Web Site Name	URL
Intrusion detection and prevention systems	Honeypots.net	http://www.honeypots.net/ids/products/
Network packet sniffers and protocol analyzers	Packet Storm	http://packetstormsecurity.org/defense/sniff/
Network protocol analyzers	Softpedia	http://www.softpedia.com/get/Network-Tools/Protocol-Analyzers-Sniffers/
Various computer and network ools	Forensic and Incident Response Environment (F.I.R.E.)	http://fire.dmzs.com/?section=tools
Various computer and network tools	Foundstone	http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/freetools.htm
Various computer and network tools	Freshmeat	http://freshmeat.net/search/?q=forensic&section=projects
Various computer and network tools	Knoppix Security Tools Distribution	http://www.knoppix-std.org/tools.html
Various computer and network tools	Open Source Digital Forensics Analysis Tool Categories	http://www.opensourceforensics.org/tools/categories.html
Various computer and network tools	Penguin Sleuth Kit	http://www.linux-forensics.com/forensics/pensleuth.html
Various computer and network tools	Talisker Security Wizardry Portal	http://www.networkintrusion.co.uk/
Various computer and network tools	The Sleuth Kit	http://www.sleuthkit.org/sleuthkit/tools.php
Various computer and network tools	The Ultimate Collection of Forensic Software (TUCOFS)	http://www.tucofs.com/tucofs.htm
Various computer and network tools	Top 75 Security Tools	http://www.insecure.org/tools.html
Various computer tools	Checksum Tools	http://lists.gpick.com/pages/Checksum_Tools.htm
Various computer tools	Computer Forensics Tools, Software, Utilities	http://www.forensix.org/tools/
Various computer tools	Funduc Software	http://www.funduc.com/
Various network tools	Common Vulnerabilities and Exposures (CVE)	http://www.cve.mitre.org/compatible/product.html