Computer Forensics: Tracking an Offender

By Jay G. Heiser,Warren G. Kruse II

Date: Nov 30, 2001

Sample Chapter is provided courtesy of Addison Wesley.

Return to the article

Learn to collect and analyze evidence found in a compromised computer system. The goal of computer forensics is to conduct the investigation in a manner that will hold up to legal scrutiny.

In this sample chapter from *Computer Forensics: Incident Response Essentials*, Kruse and Heiser explain how to track an offender across the digital matrix.

In this age of pervasive connectivity, it is unrealistic to expect cyber crime incidents to be isolated to a single system. Like characters in a William Gibson novel, cyber sleuths often have to track offenders across the digital matrix. While the techniques of network forensics are still largely undeveloped, it would be a disservice to devote an entire book to *computer* forensics without any discussion of Internet methods that you can use to find leads to suspect computers.

When tracking cyber offenders across the Internet, you use many of the same software tools that system and network administrators use to monitor and test network connectivity. Many of these programs are included in modern operating systems, and you may already be familiar with them. Even if you are already comfortable with the tools we discuss in this chapter, you may not have considered their use during an investigation. Unfortunately, many of our most common Internet application protocols make no provisions for strongly authenticating the transmitter of a communication. Services like email and Usenet are based on simple text-based initiation protocols and basically use the honor system. This complicates investigations because you cannot necessarily trust the identification information contained within Internet messages. The better you understand the underlying protocols and processes, the better you can evaluate the validity of the names and Internet addresses associated with Internet communications.

Internet Fundamentals

This book is intended to be an introduction to computer investigations, not to TCP/IP. If you want to be an effective network tracker, you need a thorough understanding of the Internet protocol suite. Many books are available on this subject. W. Richard Stevens' three-volume set, *TCP/IP Illustrated*, published by Addison-Wesley (1993, 1995, 1996), is considered one of the definitive references. The more comprehensive and detailed your understanding of Internet technology, the greater your skill at investigating network-enabled crime.

The Internet and many private networks run a set of protocols commonly referred to as TCP/IP, which stands for Transmission Control Protocol/Internet Protocol. The label "TCP/IP" is a convenient abbreviation for a set of related network protocols, the development of which effectively started in the late 1960s and is ongoing today. More precisely referred to as "the Internet protocol suite," it is a set of communication conventions that a device must implement in order to participate on the Internet. TCP/IP is not specific to any operating system, programming language, or network hardware. It is an equal opportunity set of standards that enables Macs, Windows, Unix, routers, switches, and a variety of mainframe environments to communicate with each other. It is not specific to network topology, meaning that Ethernet, token ring, and wireless networks can also interoperate. This universal interoperability is a prerequisite to both modern computer crime and investigations.

Plenty of books and essays exhaustively discuss the Open Systems Interconnection (OSI) seven-layer Network Reference model, so we won't spend a great deal of time on it. The model is illustrated in Figure 2-1. The original seven-layer model was conceived as an abstraction that didn't apply to any currently existing technology—especially not the burgeoning suite of Internet protocols—and the exact labeling of Internet services and protocols within this model continues to be a matter of tremendous debate (especially the session and presentation layers). But it is a debate of no consequence because after all, the Internet still functions whatever abstract labels are assigned to its protocols. The important lesson to learn from this model is that certain infrastructural services provide the foundation for the actual file sharing and distributed applications that are the reason the network exists in the first place. These services are stacked on top of each other like Lego building blocks. Its relevance to forensic investigations is that you cannot interpret evidence without understanding its place within the hierarchy of stacked services. Let's look at a concrete example to see how this layering works.

Figure 2-1 OSI seven-layer model

You might not have realized that when you send and receive email, you are dealing with three different addresses, each within a different network layer. Every network interface has a unique hardware address burned into it at the factory. This address is called the MAC (media access control) address. (We discuss an unusual use Microsoft makes of this address in Chapter 8.) This address enables all of the devices on a LAN segment—those devices that can see each other's network traffic—to refer to each other. At the network layer, devices recognize traffic intended for themselves on the basis of the MAC addresses incorporated within the chunks of data on the network, which are called *packets*. It is entirely impractical for every device on the Internet to refer to devices outside of their LAN segment by this hardware address, so when a computer joins the Internet, it has a numeric IP address assigned to it. An IP address is usually written as a series of four numbers in the range 0–255, separated by dots, such as 192.168.0.55.

Certain IP addresses, or ranges of addresses, are reserved for special purposes. For example, IP addresses that end with 0 denote a network address, such as 192.168.0.0. An IP address that ends with 255 denotes a broadcast address, such as 192.168.0.255. "Private addresses" in the 192.168.0.0 to 192.168.255.255 range may be used on internal networks. These addresses "are intended for intra-enterprise communications, without any intention to ever directly connect to other enterprises or the Internet itself." [1] When tracking offenders, if you locate an address within this range, don't pack your bags for California (the location of the Internet

Assigned Numbers Authority[2]); you have to determine the suspects' external IP address to locate them.

An Internet address actually contains two parts. The network portion is unique among all the networks interconnected to the LAN segment (which often means the entire Internet), and the host section is unique among all the devices using the same network portion. The effect is that all IP addresses on the Internet are both unique and identifiable as being within a specific network. Private networks use addressing that is unique within their networks, but any two private networks can use the same "address space" as long as they are not interconnected to each other.

The uniqueness of addresses and the distinction between network and host portions of the address make it practical for routers to know where to route to. Entire books have been written about routing. For our simplified purposes, *routers* are devices that automatically forward your data packets to another network when the destination is not your network. Routers base their decision on where to forward your packet on current conditions and their programmed instructions—routers do whatever is most expedient, which means that the route between any two points can change. This is completely different from the Public Switched Telephone Network (PSTN). When you make a telephone call, the switches within the PSTN sequentially establish a circuit from end to end, and it is maintained throughout the duration of the call. On the Internet, it may often seem as if you are using a circuit, but the actual path taken by each individual packet is dependent upon the whims of the intermediate routers.

The network part of an Internet address is assigned by the Internet Assigned Numbers Authority (IANA) to each network owner, and the host part is assigned to individual hosts and devices by the network owner. The network may be run by an organization (business or government agency), or it may be run by an Internet service provider (ISP) to provide Internet access to its customers. In the latter case, the IP addresses may be used by individuals or multiple organizations. Because IP addresses are used for routing, when a device is moved to a new network, it often requires a new address.

IP address can be statically or dynamically assigned. Computers that are assigned a static IP address always use the same IP address until it is manually changed to a new address, which is becoming increasingly less convenient in a time of constant reorganizations and mobile computers. Dynamic addresses are automatically assigned to a computer when it registers itself on a network using a protocol called Dynamic Host Configuration Protocol (DHCP) or Windows Internet Naming Service (WINS), a Microsoft protocol that is rapidly becoming obsolete. For network administrators, DHCP neatly solves the tedium and confusion of manually assigning constantly moving Internet devices. Virtually all ISPs use DHCP to assign addresses to their dial-up customers, and many permanently connected home users have dynamically assigned addresses that can change whenever their cable modems are powered off and on. Use of DHCP is definitely on the increase, but unfortunately, DHCP makes detective work a little more difficult.

Reading Obfuscated IP Addresses

Those who send spam, unsolicited commercial junk mail, usually try to keep their true identities secret—otherwise, they would be overwhelmed by disgruntled Internet citizens who wish to retaliate. In addition to using a bogus return email address, they often include obfuscated URLs. Instead of having a human-readable name, or the dotted-decimal format such as `135.17.243.191`, a URL may appear in 10 Digit Integer Format (base 256), so it appears like this: `http://2280853951`.

It's fairly easy to convert a number in this format back into the normal quad format so that you can research the ownership of a Web site.

1. Open Windows Calculator in scientific mode (in the Calculator window, choose View | Scientific).

2. Convert 2280853951 to hexadecimal format = 87F311BF.

3. Now convert each pair to decimal notation and add the dots:

```
87 = 135
F3 = 243
11 = 17
BF = 191
135.17.243.191

Dotted Quad to 10 Digit Decimal (base 256):
Dotted Quad format: A.B.C.D = 10 Digit Decimal #
A(256^3) + B(256^2) + C(256^1) + D =

Example:
185.127.185.152 =
185(256^3) + 127(256^2) + 185(256^1) + 152 =
3103784960 + 8323072 + 47360 + 152 = 3112155544
```

Or if you hate math like we do, the easiest way to convert is to let ping or traceroute do it for you. Running ping or traceroute on the 10 Digit Decimal Number will resolve its Dotted Decimal notation, showing you the dotted quad format equivalent. For example:

```
C:\>ping 2280853951
Pinging 135.17.243.191 with 32 bytes of data:
```

In case you were worried that we hadn't figured out what to do with media address control (MAC) addresses, don't worry, we still need them. Remember that devices on the same LAN segment are somewhat on a first-name basis. They don't refer to each other by the formal IP addresses used on the Internet. However, the MAC address is used only at the hardware layer, so when a process or application "up the stack" specifies another device on a network segment by IP address, it has to be translated into a MAC address. This is done by looking it up in the ARP table, which is automatically created by the Address Resolution Protocol. ARP is just one of a number of network services that run in the background, invisible to most users but essential to the operation of a network. Networked computers can be quite chatty, constantly comparing notes on routing tables, network conditions, and each other's presence.

There is a common belief that because MAC addresses are burned into the network interface card (NIC), they never can be changed. The MAC address can be changed by using the ifconfig command in Unix. Given that MAC addresses are sometimes used to identify the source of hostile activity, it should also come as no surprise that programs are available that can randomly change a

MAC address.[3] Don't automatically assume that a piece of equipment is useless as evidence because its MAC is different than you expected. The MAC you are seeking may have been changed through software, or the NIC may have been changed.

You probably already have used the most common tool for network debugging, ping. (By the way, the name is not an acronym; it is a reference to the underwater echolocation system called SONAR.) ping is a simple, yet greatly valuable program, that uses Internet Control Message Protocol's ECHO_REQUEST datagram. This datagram sends a request to the target machine and listens for an ICMP response. You can use ping to determine when a machine is alive and sometimes the DNS name of the machine. If you want to continuously keep checking for a "live" machine, you can use a program like What's Up Gold. With this program and others, you can input the IP address, and at preset intervals, it automatically checks to ensure that a specific service on a specific host is still reachable. Be aware that ping is a relatively noisy process—it is easily detected by the remote system. Assume that a moderately savvy Internet criminal may be monitoring all forms of connection to his or her host, so don't ping someone when you don't want that person to know about it.

Domain Name Service (DNS)

We expect computers to refer to each other by numbers—not by name. Unfortunately, most humans don't do as well with numbers and prefer to use names that can be easily remembered, spoken, and typed, such as http://www.cia.gov or http://www.amazon.com. To accommodate this difference between humans and machine, the Domain Name Service (DNS) was developed. Internet DNS is effectively a huge global database, usable from any point in the Internet and capable of mapping human-readable names such as http://www.lucent.com to a corresponding numeric IP address. This process is called *domain name resolution*. Use of domain names and associated IP address ranges is controlled by the Internet Corporation for Assigned Names and Numbers (ICANN) through accredited registrars.[4] The owner of each domain is responsible for placing all host names and corresponding IP addresses on a name server so that outsiders can resolve their names. Most name servers also support *reverse lookups,* which is the process of providing the human-readable domain name that corresponds to a specific numeric IP address. Many Internet applications perform reverse lookups as a simple security measure, checking to ensure that the IP address associated with an incoming connection attempt is associated with a registered domain name—a weak but useful test.

The domain name server responsible for a particular domain may resolve any query with any IP address. The IP address may not be one within an IP address range assigned to that organization, and that doesn't matter. The owners of a particular domain, such as bubbabbq.com, may choose to host their Web site at someone else's facility. In this case, the specific machine, http://www.billybob.com, won't have an IP address contiguous with the rest of bubbabbq.com. This provides a great deal of flex-ibility, allowing organizations to move their machines from network to network, changing Web host service providers or ISPs without having to change their human-readable domain names.

Another type of network tool that will be useful to you in tracking an offender is one that can be manually used to resolve a domain name. The classic tool for this purpose, nslookup, is available on Unix, Windows NT, and Windows 2000. You can use nslookup to perform both forward and reverse lookups, resolving the IP address associated with a specific host name or obtaining the name associated with a numeric address.

In order to use a domain name, the owner must register it with the appropriate authority—a task that is usually facilitated by an ISP or one of several online services. At the time of registration—and ideally whenever it is changed—the owner of the domain is required to include name and contact information for a domain administrator. This person is expected to respond to email messages or telephone calls regarding activities associated with his or her domain. It should come as no surprise that these people are frequently not easy to contact. The whois utility can be used to obtain contact information on a specific domain from a server maintained by the appropriate Internet naming authority. Remember that whois information is furnished by the person who provided the registration information. It isn't really verified for accuracy; either through deliberate deception or an innocent mistake, it is possible to register an address and include inaccurate or totally false contact names, addresses, and phone numbers.

After pinging a system that we're researching (from a computer that is not going to resolve to our company in case the suspect is watching his or her network), we like to perform a whois just to see what comes up, keeping in mind that the information can be bogus. You don't have to have a whois utility on your workstation because several sites enable you to perform a whois over the Web. One of the most popular is the Sam Spade Web site.[5]Another popular and reliable Web-based whois service is provided by Network Solutions.[6]After we perform a whois, we like to follow up with an inverse name server lookup to see what it provides and compare the results to the whois output. The inverse lookup can be accomplished on a Unix or Linux machine (or with software such as NetScanTools Pro for Windows) with either the nslookup or the dig –x command, and Sam Spade provides reverse lookup services also. You can use dig on an IP address like this:

```
dig –x @123.456.789.000
```

or

```
dig –x %domainname.com
```

dig is an alternative to nslookup, but we usually run nslookup again just to compare the results to all the previous queries.

After we have obtained contact information using the tools previously described, we usually run traceroute (or tracert) to see what route the packets are taking to get to their destination. Like ping, this handy utility sends your packets to the computer you are examining, so don't use it if you don't want to tip off the suspect that you are watching. We use the results from traceroute to help confirm or question the results of whois (see the example in Figure 2-2). For example, if the site is registered to the Netherlands but traceroute takes a few hops and stops at an ISP in Philadelphia, we might suspect that something is amiss. Be aware that many corporations have their Web sites hosted by an ISP, and not necessarily an ISP in their home town—or even their home country.

Tracing route to awl.com [204.179.152.52]over a maximum of 30 hops:

Figure 2-2 Example traceroute output

Application Addresses

We're almost done with our discussion of Internet addresses—we just have one more layer to discuss: application addresses. Email, Web browsing, ICQ, and Inter-net Relay Chat (IRC) are just a few of the services that have their own application-specific addressing. When you send email, you know to use a two-part address that includes both a mailbox and a domain, such as wkruse@computer-forensic.com. You can't send email to wkruse, nor can you just send it to computer-forensic.com—you need to specify both in order for a message to reach a destination.

Another ubiquitous form of Internet addressing that includes both domain- and application-specific information is the Universal Resource Locators (URLs) used with Web browsers. For instance, the URL http://www.lucent.com/services provides three types of information. The letters "`http://`" indicate the application protocol, which in this case is Hypertext Transfer Protocol (HTTP). "`http://www.lucent.com`," of course, represents a specific numeric IP address, and "services" points to a specific page.

Stalking the Stalker

When using programs such as ping, and some of the scanning tools, keep in mind that their use may easily tip off even a novice computer crook that he or she is under investigation. In the online world of spy versus spy, you often use the same tools to track an intruder that an intruder uses. Clever attackers are like careful scouts. Those who suspect that they are being tracked will try to cover their tracks (see Chapter 10), and they also sometimes will backtrack to see if someone is following them. Skillful hackers carefully monitor systems they've compromised for signs of unwelcome attention. If your goal is to track and catch your prey, you don't want to be blasting away at the suspect's box—especially from machines that easily resolve back to your company's name. To use the SONAR analogy, there are both passive and active surveillance techniques. ping and traceroute are active and cannot be hidden from the object of the scan. While pings cannot be hidden from the object of the ping, it is easy enough to perform the scans from a system that isn't obviously associated with your organization. You can maintain a few accounts with a dial-up ISP for just this purpose. When it's time to start investigating, unplug the workstation, set it for DHCP, dial your ISP, and you're off the company network. The intruder who has been breaking into CorpNet does not suddenly see CorpNet probing back.

A Dial-Up Session

Now that you have an understanding of some Internetworking basics, let's take a look at how a typical Internet dial-up session works (see Figure 2-3). When you dial to an ISP with a modem, you might use a layer 3 protocol called Point to Point Protocol (PPP). Referring back to Figure 2-1, layer 3 is the network layer, and in the case of a dial-up connection, PPP replaces IP. Connectivity is not automatic, though. A dial-up session must first be authenticated, and then an IP address is assigned. The modem at the ISP's Point of Presence (POP) is directly connected to—or even a component within—a router that is designed to accommodate PPP connections. When a connection attempt occurs, the dial-up router first prompts the user for a login name and password. A single ISP may have hundreds of POPs spread over an entire continent— it is certainly not practical for each dial-up router to maintain a list of all users and their encrypted passwords. A centralized directory contains this list, and the RADIUS protocol is used to support the authentication request between the dial-up routers and the centralized user directory.

Figure 2-3 Connecting to the Internet through an ISP's dial-up service

After a user is authenticated to the ISP, an IP address is dynamically assigned to that user with DHCP. Although it is possible for individual subscribers to have their own permanently assigned IP addresses, such an inefficient use of valuable IP address space is virtually unheard of. The IP address is almost always associated with a DNS name, allowing reverse lookups. The name will be something generic, such as ppp589.city.isp.com.

RADIUS is used not just for authentication; it is also used for accounting. The RADIUS server is normally the only ISP device that maintains records that can be used to track an offender, so it is very important to your investigation. The server normally maintains records of every login attempt, both successful and unsuccessful, and also every logoff or session end. This information is necessary so the ISP can keep track of subscriber connection time. The information associated with a RADIUS session also includes the IP address assigned to a specific login during a session, and ISPs often use caller ID to keep track of the telephone number used to originate the session. This allows the ISP to determine which login name was using a specific IP address at a specific time, but the association of this login with a specific individual is only as good as the authentication mechanism. Most dial-up accounts authenticate with reusable passwords, and it is common for cyber criminals to guess or otherwise steal passwords (most subscribers have no way of knowing that their accounts are sometimes being abused by someone else). America Online (AOL) users have been especially prone to ID theft, and AOL is just one of many ISPs that provide free trial accounts that are frequently associated with phony names.

Because the RADIUS logs are used for accounting purposes, an ISP has to maintain them for at least a one-month billing cycle. In practice, ISPs keep them for periods of up to a year in order to respond to customer complaints about billing mistakes. Even relatively small ISPs are used to responding to court orders that require providing the Internet equivalent of a trap and trace record. According to Lucent consultant Aaron Higbee, who has worked with the abuse departments of several large Internet service providers:

> ISPs do not like abusers because their mischief affects the bottom line and gives the ISP a black eye within the Internet community. If you want to identify an abuser, these are the necessary steps:
>
> 1. Document the abuse with dates, time, time zone, and logs.
> 2. Send the logs as a complaint to abuse@isp.com.
> 3. Follow up your email with a phone call. (Do not call a tech support or customer service line.) Ask for the legal department's fax number or ask to speak directly with the abuse/security department.
> 4. Fax the same logs to the legal staff and let them know that you will follow up your complaint with a court-ordered subpoena for any and all subscriber information including all captured caller IDs.
>
> You must assume the subscriber information is fraudulent unless the account has a bill payment history and the session in question can be pinpointed as originating in the same calling area as the rest of the subscriber's usage history. If you are lucky, the caller ID will be captured for the session you are interested in. You then subpoena the local phone company for

subscriber information for the phone number that was captured in the caller ID. Sometimes reverse telephone lookup sites like http://www.anywho.com/rl.html can give you clues as to who you are tracking, but the definitive answer will come from the subpoenaed subscriber information.

You might think the biggest problem with obtaining information from ISPs would be the result of the terms of service and confidentiality agreements that most service providers have with their customers. But to the contrary, most service providers are willing to assist you because they do not want anyone misusing their system. In a prominent privacy case several years ago, AOL was sued by a subscriber who accused the company of illegally providing sensitive personal information to a law enforcement agency, so ISPs are now very sensitive to the correct legal procedures.

When you obtain the information from the service provider, keep in mind that the subscriber information can be completely bogus. There is little to no authentication for any of the information associated with the subscriber. The value of the information is in determining the telephone number that was used to connect to the ISP. If you can obtain the phone number and the date and time that a session was set up, you are yet another step closer to finding your suspect. You can then start the subpoena process again and try to find other connections originating from that same phone number. This still might not lead directly to your suspect, but you're getting closer and closer to a suspect who thought he or she was well hidden by the free service.

Tracing Email and News Postings

Before heading down the messaging path and looking for tracks in the sand, let's quickly discuss how these messaging services operate. News groups and email are cousins. Descending from original siblings on pre-Internet Unix systems, they have continued to evolve in parallel, with much sharing of genetic material. Both services have the following attributes:

- Simple Internet application protocols that use text commands
- Store-and-forward architecture allowing messages to be shuttled through a series of intermediate systems
- Message body composed entirely of printable characters (7-bit, not 8-bit)
- Human-readable message headers indicating the path between sender and receiver

You'll need the assistance of systems administrators, perhaps on every system the message transited, and they won't be able to help you unless they have logging information on their messaging hosts. If the originator wants to cover his or her tracks, determining the real sender of either bogus news postings or suspicious email can be challenging. News is probably a bit easier, but email is more common today, so let's start with it.

Tracking Email

An email program such as Outlook, Notes, or Eudora is considered a *client* application, which means that it is network-enabled software that is intended to interact with a *server*. In the case of email, it is normal to interact with two different servers: one for outgoing and one for incoming mail. When you want to read email, your client connects to a mail server using one of three different protocols:

- Post Office Protocol (POP, not to be confused with Point of Presence)
- Internet Mail Access Protocol (IMAP)
- Microsoft's Mail API (MAPI)

For the purposes of investigation, the protocol used to gather incoming email from a server is of minimal interest. The most important thing to understand about these different protocols is that their use affects where mail messages are stored (as depicted in Table 2-1). All incoming mail is initially stored on a mail server, sorted by that mail server into individual *mailboxes* for access by the addressee. POP users have the choice of either downloading a copy of their mail from their server, or downloading it and subsequently allowing it to be automatically deleted. Email that has been read or stored for future use is stored on the computer that is running the email client. IMAP and MAPI users have the option of leaving all their mail on their mail server.

There are two major advantages to leaving email stored on the server. First, all of the stored email for an entire organization can be easily backed up from a central location. Second, it provides users the flexibility of accessing their mailboxes from multiple client machines: office, home, through the Web, and so forth. The implications of this to the investigator is that POP mail users always use their local machine for their email archives: copies of outgoing mail, mail stored in folders for future reference, deleted mail that hasn't been purged, all are stored on the individual's workstation. Organizations that provide IMAP or MAPI service, or a proprietary service like Lotus Notes, probably store email on the server, although individual users may or may not have the option of storing their email locally.

Table 2-1 Internet Email Protocols

Post Office Service	Protocol	Relevance to Investigation
Incoming message	POP	Must access workstation in order

Post Office Service	Protocol	Relevance to Investigation
store only		to trace mail.
Storage of all messages (optional)	Open: MAPI Proprietary: Microsoft MAPI Lotus Notes	Copies of both incoming and outgoing messages may be stored on server or workstation (and server/workstation backup tapes).
Web-based: send and receive	http	Incoming and outgoing messages are stored on server, possibly with optional manual download to workstation. Facilitates identity spoofing.

Outgoing email uses a completely different protocol called Simple Mail Transfer Protocol (SMTP). Unlike the protocols used to retrieve mail from a post office, SMTP doesn't require any authentication—it is much like tossing a message into a mail slot at the post office. Servers that accept mail and relay it to other mail servers are sometimes called mail transfer agents (MTAs), and they also use SMTP. Your ISP will give you the name of the mail server that you should use for outgoing mail, often something along the lines of smtp.bobsisp.com. The SMTP server that the ISP uses relays messages to their destinations. Either the destination server recognizes a message as being addressed to one of its local users, and places it into the appropriate mailbox for that user, or based on a set of rules, it relays the message on further.

SMTP is a very simple protocol. Like many Internet protocols, such as HTTP, it consists of a few simple text-based commands or keywords. One of the first tricks an Internet hacker learns is how to manually send an email message by telneting to port 25, the SMTP port. Not only is it a fun trick to become a human email forwarder, but it also enables you to put any information you want into the headers of the email message you are sending—including fake origination and return addresses. Actually, you needn't do this manually if you want to fake email. When you configure your personal email client, you tell it what return address to put on your outgoing mail. You can always change that configuration, but if you want to send only a single message coming from Pres@whitehouse.gov, it is much easier to use one of several GUI-based hacker tools that enable you to quickly send a message with your choice of return addresses.

SMTP mail has no strong authentication and without using PGP or S/MIME (Secure Multipurpose Internet Mail Extensions) to add a digital signature, the level of trust associated with a mail message is pretty low. The following steps (our input is in boldface) show how incredibly easy it is to fake the return address in an Inter-net mail message:

```
[root@njektd /root]# telnet localhost 25
Trying 127.0.0.1...
Connected to njektd.com.
Escape character is '^]'.
220 njektd.com ESMTP Sendmail 8.9.3/8.9.3; Tue, 5 Dec 2000 17:37:02 –
0500
helo
250 OK
mail from: teswt@test.com
250 teswt@test.com Sender ok
rcpt to: you@domain.com
250 you@domain.com Recipient ok
data
354
Haha-this is a spoofed mail message!
.
250 RAA25927 Message accepted for delivery
quit
221 njektd.com closing connection
Connection closed by foreign host.
```

The results of this spoofed mail message are shown in Figure 2-4. The test.com domain is just one we made up for demonstration purposes, but the email client reports whatever information it was provided.

Figure 2-4 Reading the spoofed mail message

As we'll discuss later in this chapter, some identification information is associated with the mail header that is a bit harder to spoof. As recorded in the following mail header, we were indeed logged in as 208.164.95.173:

```
Received: from dhcp-95–173.ins.com ([208.164.95.173]) by
dtctxexchims01.ins.com with SMTP (Microsoft Exchange Internet Mail
Service Version 5.5.2653.13)
id YM4CM2VP; Sun, 10 Dec 2000 08:46:30 -0600
From: teswt@test.com
Date: Sun, 10 Dec 2000 09:46:46 -0500 (EST)
Message-Id: 200012101446.JAA06830@horh1.emsr.lucent.com
```

When relaying a message from another relay host, current versions of SMTP also keep track of the IP address of the system connecting to them, and they add that IP address to the header of the message. If you want to show that a mail message originated from a specific computer, the best way to do so is to investigate the entire path that appears in the extended header information.

Although SMTP servers won't perform any authentication when receiving mail from the client, most Internet mail servers are configured to accept mail for relay only when that mail originates from a specific range of IP addresses. If an ISP does not place any limits on which systems can connect to the ISP's mail server, allowing it to be used as a free relay station, it won't take spammers long to find it. To reduce the amount of spam mail that originates with their mail servers, most ISPs allow relay connections only from IP addresses within the range that they assign to their own subscribers. Authentication based just on IP address is very weak, but for the purposes of preventing unauthorized use of SMTP servers, it is adequate.

It should come as no surprise that Web-based email is not only available, but is becoming increasingly popular. The Internet browser is rapidly becoming the universal front end. Web-based email enables users to access all of their email—both incoming and saved messages—through a Web browser. Not only does this free the user from installing and configuring an email client on his or her workstation, but it also means that the user can easily access email from any workstation virtually anywhere in the world. Undoubtedly, many people are accessing free Web-based email from work for their personal use.

It also shouldn't come as a surprise that free email services are being used by some people to hide their identities. For the novice computer criminal, these services appear to be an easy way to hide their identity, and by adding at least one more server and involving another service provider, it certainly does complicate the association of a mail account with a specific person. The only way to find out who the ISP thinks is using a specific email address is to obtain a subpoena for the account information. If you are working with law enforcement agencies, they can obtain a subpoena to facilitate their investigation, or you can obtain a subpoena from a lawsuit (for more information, see Chapter 12). Fortunately, some providers of free email service are including the originator's IP address in the header information. Previously, you would have to subpoena the email provider and then the originating ISP to determine the originator. We recommend issuing a subpoena for the email logs from the email provider, but at the same time, you can also subpoena the originating ISP.

Reading the Mail Trail

When you are investigating a case involving email, you have to decipher email headers. If you have never viewed a header before, it might first appear to be gibberish, but once you get the hang of it and understand how SMTP works, it makes sense. The first annoyance you encounter is that most client email programs hide the header information from the user. Depending on the mail client you're using, you may have to do a little bit of digging to get to the header. In most programs, if you click File|Properties, an option to view the header is displayed. If your particular program provides a different way to access header information, consult the Help menu and documentation or try the company's Web site for instructions.

Most users don't want to be bothered with deciphering email headers, which encourages the email software vendors to make access to it as obscure as possible. Let's look at Microsoft Outlook Express. It is a popular email program for many reasons, including the fact that it comes free with Internet Explorer and recent versions of Windows.

As shown in Figure 2-5, the header information is available in Outlook Express by clicking on File and then Properties, which displays the dialog box that looks like that shown in Figure 2-6.

Figure 2-5 Outlook Express File menu

Figure 2-6 Outlook Express Message Properties window

Figure 2-7 Viewing the message source in Outlook Express

The General Tab for the properties in Outlook Express displays some basics about the message such as the subject of the message, the apparent sender, and the date and time sent and received. Click on the Details tab to display the information like that shown in Figure 2-6. By examining the headers of this message, it is clear that both the From address (test@testing.org) and the Reply-To address are fake addresses (another_test@test. org). This is a real message that we sent from the Internet, but before sending the message, we first changed the From address to "HTCIA NE Chapter." The From address is completely arbitrary—it contains whatever the sender configures into their email program.

The most important tracks are found at the top of the message. In this case, the first line shows the computer that the message was originally sent from. While the name of the PC, "mypc," can easily be spoofed, the IP address that mypc was assigned when we logged on to the ISP is much more difficult to spoof. While it is not impossible to spoof an IP address, we are not aware of a case in which one has been spoofed to counterfeit email. The practical details involved in spoofing an IP address make it virtually impossible in an email transaction, which involves several round trips between the SMTP server and the connecting system. (Do be aware, though, that the actual sender of the message could have cracked the system from which it was sent, and logged on as somebody else.) In this case, the email was sent from a computer in the same domain, monmouth.com, as the SMTP server that relayed the mail, shell.monmouth.com. Do a whois on the IP address and see if you get something that matches the purported domains of both the originating client and the relay server. Then follow up using the Dig w/AXFR advanced query, as shown in Figure 2-8, using NetscanTools.

Figure 2-8 Using NetScanTools to investigate an IP address

In contrast to Outlook Express, Microsoft Outlook (included with Microsoft Office) places the full email header information in an obscure position. As shown in Figure 2-9, to view the header information, you click on View and then Options. Clicking on Message Header seems to be a more obvious way to access header information—a mistake that we make all the time—but all that does is hide the To, From, and Subject lines from the message itself. It does not show you the detailed header information that you need to track an intruder. By clicking on Options, you access the Message Options window shown in Figure 2-10.

Figure 2-9 Outlook View menu

Figure 2-10 Viewing a message header in Microsoft Outlook

You've probably already noticed "Joe Anonymous" in the Have replies sent to field. We faked this deliberately to illustrate how you cannot believe everything you read. The only way to extract this information from this window is to select it all (hint: try Control-A), copy it, and then paste it into a text document, which we've done in the following:

```
Received: from hoemlsrv.firewall.lucent.com ([192.11.226.161]) by
nj7460exch002h.wins.lucent.com with SMTP (Microsoft Exchange
Internet Mail Service Version 5.5.2448.0) id W4VCF23A; Sat, 20
Nov 1999 21:19:10 –0500
Received: from hoemlsrv.firewall.lucent.com (localhost [127.0.0.1]) by
hoemlsrv.firewall.lucent.com (Pro-8.9.3/8.9.3) with ESMTP id
VAA06660 for <wgkruse@holmdel.exchange.lucent.com>; Sat, 20 Nov 1999
21:19:10 –0500 (EST)
Received: from shell.?nmouth.com (shell.?onmouth.com [205.231.236.9])
by hoemlsrv.firewall.lucent.com (Pro-8.9.3/8.9.3) with ESMTP id
VAA06652 for <wgkruse@lucent.com>; Sat, 20 Nov 1999 21:19:09
–0500 (EST)
Received: from mypc (bg-tc-ppp961.?onmouth.com [209.191.51.149]) by
shell.?onmouth.com (8.9.3/8.9.3) with SMTP id VAA01448 for
<wgkruse@lucent.com>; Sat, 20 Nov 1999 21:17:06 –0500 (EST)
Message-ID: <001401bf33c6$b7f214e0$9533bfd1@mypc>
Reply-To: "Joe Anonymous" <another_test@test.org>
From: "Joe Anonymous" <test@testing.org>
To: <wgkruse@lucent.com>
Subject: test from outlook express
Date: Sat, 20 Nov 1999 21:18:35 –0500
MIME-Version: 1.0 Content-Type: text/plain;
charset="iso-8859–1"
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 4.72.3155.0
X-MimeOLE: Produced By Microsoft MimeOLE V4.72.3155.0
```

This header is longer than it was in our first example. This time, the message was relayed through four different servers. Each time an SMTP server accepts a message, it places a new Received header at the top of message before forwarding it on to another server or a user mailbox. In the first message, we intentionally sent the message *from* our ISP account *to* our ISP account. The second message originated externally and then was relayed through the Lucent firewall and to the mail server used to host our mailbox. But even with the extra headers, it is still apparent that the original message was received from: "mypc," and our address at the time was: ?onmouth.com [209.191.51.149]. The lines in the header tell a story about where the email message has been, when it was there, and when it was delivered to its destination. It may be a contrived story, but it is still a story. Virtually any of the headers with the exception of the topmost one could be bogus—it is up to you to verify each one of them and determine the actual history of the message.

The last example that we will look at is from Eudora, another popular email client.[7] Eudora hides the header information by default, like the rest of the client programs, but as you can see from the Eudora Lite example in Figure 2-11, the full header is only a mouse-click away. A helpful piece of information is X-Sender, which shows the mail server and the account the message was sent from. One of the quirky characteristics of Eudora is that the icon is labeled "Blah, Blah, Blah." Strange label, but it provides the information we need. When you click on the Blah button, your email message changes from that shown in Figure 2-11 to something that looks like that shown in Figure 2-12.

Figure 2-11 Viewing the X-Sender header on Eudora Lite

Figure 2-12 Viewing mail headers with Eudora

When you are conducting an investigation involving email, even if the computer's name is bogus, you have the IP address that the user was assigned. If you trust the ISP, or the company that the address resolves to, you can ask for their assistance in identifying your suspect. They probably won't disclose the information to you immediately, but by noting that IP address, along with the exact date and time that the message was sent, you can ask the ISP to save the logs until you have a chance to get a court order. As long as the logs are still available, the ISP or other organization should be able to identify the user that was assigned that IP address at the time the message was sent.

Look at the two headers the arrows are pointing to in Figure 2-13. Compare the domain name in the Received header, "monmouth.com," to the domain in the From header, "test.org." Because they do not match, we can assume that the user configured his or her email incorrectly or that the user is trying to hide his or her identity. A message can be injected anywhere in the chain of Received headers—you can be sure that only the topmost one is accurate. Do an nslookup against each domain—especially the purportedly original domain—and see if they exist. Do a whois against each of those domains to find out who the administrator is and contact that person. Keep in mind that if the administrator is the originator of a phony or illegal message, he or she probably won't be inclined to cooperate.

Figure 2-13 Extended email header with discrepancies in originator fields

When you are investigating a case on behalf of a victim, but you can't visit the victim or otherwise obtain the original message on your own, it is possible for the victim to email you a copy of it. You must give the victim very specific instructions on the appropriate way to send the mail to you—especially if the victim usually deletes messages right after receiving them. Ask the victim to send the message as an *attachment* and not to forward the message. Forwarding replaces the suspect's information with your victim's information. You might want to ask your victim not to delete the original message until he or she hears from you.

SMTP Server Logs

All email servers have the ability to maintain logging information. If available, these logs are usually a more reliable source of information than the mail headers. Because of the high volume of email traffic, the ISP may not store records for very long. SMTP server logs are important tools, used by both ISPs and user organizations to troubleshoot email problems and track down the unauthorized use of mail. The only way to completely verify the path of the mail is for the administrator of each host to check the logs to find that they both sent the message to the recipient as indicated and that their host received it from a specific IP address. If you are lucky, you can follow the chain all the way back to an original IP address. Examination of the original system, which might need to be cross-referenced against RADIUS logs and phone records, is the only way to prove that a specific message existed at a spe-cific place. Accessing this workstation usually requires a court order, but once a perpetrator is confronted with having the incriminating evidence on his or her personal computer, a confession is often not far behind.

Usenet

Usenet is a vast, distributed bulletin board, consisting of thousands of hierarchically arranged topics contributed to and read by millions of people worldwide. It was originally developed during a time in which Unix computers used modems for external connectivity. Email and even individual files could be sent between noncontiguous UNIX computers, as long as you knew the intervening series of connecting systems and chained their names within the destination mail address (such as ober!doebling!seismo!krampus!kirk). Usenet still takes advantage of this same point-to-point model, although now it uses the Internet as a transport backbone. As shown in Figure 2-14, when a post is made, it is transmitted through a chain of news servers and then it is diffused back out through all of the servers in the world that subscribe to the same newsgroup.

Figure 2-14 Example news server hierarchy

Usenet has adapted itself well to changes in Internet technology, resulting in a nearly seamless integration of traditional news readers with email and Web interfaces. Today, Usenet users have their choice of news clients, email or Web interfaces, both for posting and for reading news messages (see Figure 2-15). Some Usenet mail gateways are also configured as email distribution listservers, enabling Internet citizens to have all the postings for a certain newsgroup delivered directly to their mailboxes, either a message at a time or collected periodically into *digests*. The implications to you as the tracker is that your attempt to research the origin of a news message may involve multiple types of servers.

Figure 2-15 Those reading and posting news have a choice of using a traditional news reader, a web-based service, or an email gateway.

If you are not familiar with Usenet, and have not configured a client to read postings, the best place to start is the original Internet archive for Usenet, the DejaNews archives provided by Google.[8] An example search is shown in Figure 2-16.

Figure 2-16 Example Google page

Inappropriate material is probably the most significant use of news that you are likely to encounter in an investigation. Unfortunately, huge amounts of pornographic pictures are disseminated through news, and it is common for some corporate employees to download this material, exposing their employers to the risk of sexual harassment lawsuits. Pictures, and other binaries, will be found in the uuen-code format (see Chapter 4).

Usenet can provide a forum where you can publicly ask technical questions. You may also want to search archived news postings to see if you can find anything related to one of your suspects or an organization involved in your investigation. Some suspects may be active news posters. Searching for their names on a server such as Google may give you valuable clues to their activities and interests. At times, news postings are directly relevant to an investigation. It is not unheard of for disgruntled employees to try to damage the reputation of their company by making news posts containing harmful information. In some cases, it is possible to trace these inappropriate postings to IP addresses actually within the corporation.

Tracking Usenet Posts

Just like when you are tracking email, you are dependent not only upon the cooperation of all the news server administrators relevant to the message that you are tracking, but you are also dependent upon their having adequate logs. News volume tends to be very high, so the information in the logs is extremely volatile. It is not unusual for a busy Usenet site to turn over their logs every day, so if you want to track a posting, your chances of success are slim if you can't do it within 24 hours. The process you follow will be very similar to tracking a mail message, and some ISP abuse department staffers feel that news is easier to trace because, after some practice, the bogus header information is relatively easy to discern. Just like when tracking email, you work your way back from the recipient and verify each of the machine names in the path. Either you'll find a bogus connection point, which is probably where the message was inserted, or you'll actually verify that the message apparently did transit all of the hosts shown. If the origination host has a record of the poster, you can either subpoena that organization for more information or notify their abuse department.

Let's take a look at the header of a news posting. The following one is an example of a pornographic post that has spent quite a lot of time zipping around the Internet:

```
From: "YourMate" binkie1@xxx.com
Newsgroups: alt.binaries.dominion.erotica,
alt.binaries.dominion.erotica.female, alt.binaries.erotica,
alt.binaries.erotica.amateur.female, alt.binaries.erotica.blondes,
alt.binaries.erotica.centerfolds
Subject: - FREE CD !!!
Date: Sun, 5 Dec 1999 10:59:38 –0500
Lines: 2484
X-Newsreader: Microsoft Outlook Express 4.72.3110.1
X-MimeOLE: Produced By Microsoft MimeOLE V4.72.3110.3
NNTP-Posting-Host: dialup599.xxx.com Message-ID: 384a8a4b@news.xxx.com
X-Trace: 5 Dec 1999 10:52:43 -0500, dialup599.nni.com
Path:news.rdc1.nj.home.com!newshub2.home.com!newshub1.home.com!news.ho
 me.com!feeder.via.net!news.idt.net!netnews.com!newspeer1.nac.net!news.
newyork.net!ffx.uu.net!uunet!ams.uu.net!tank.news.pipex.net!pipex!news -
lond.gip.net!news.gsl.net!gip.net!nntp.news.xara.net!
xara.net!gxn.net!news.good.net!news.xxx.com!phila-dialup599.xxx.com
Xref: newshub2.home.com alt.binaries.dominion.erotica:30040448
 alt.binaries.dominion.erotica.female:30252819
alt.binaries.erotica:31064499
alt.binaries.erotica.amateur.female:30199540
alt.binaries.erotica.blondes:30083027
alt.binaries.erotica.centerfolds:30058010
```

With the understanding that virtually everything on this header can be faked, each header purportedly contains the following:

- **From:** This is the name and email address of the original poster.
- **Newsgroups:** In this case, the message was cross-posted to six different news-groups.
- **NNTP-Posting-Host:** This is the machine from which the posting was sent. It is usually the same as the lower rightmost entry in the Path header.
- **Message-ID:** This is a unique serial number assigned to every post on NNTP servers. The logfile on news.xxx.com should have an entry that associates this serial number with this specific message and a specific account. This ID should be unique throughout the worldwide Usenet, and it can be used to cancel messages across all (well, most) news servers worldwide.
- **Path:** This is the meat of your investigation. In reverse chronological order, it shows every host that the message has transited. The first host, news.rdc1. nj.home.com, is the host on which this message was received, and it is the only information in this entire header that you can trust. The succeeding hostnames are the hosts that purportedly relayed this message. The next-to-last host, news.xxx.com, is apparently the news server at an ISP, while the final hostname looks like a dial-up account at that same ISP.

Note that on the preceding header, the domain name in the From field is consistent with the domain name in the NNTP-Posting-Host and Path fields. If it were not, you would know immediately that someone was attempting to cover his or her tracks. If they do match, you probably still need to verify the intermediate hosts, but you can start with the origination point. If they don't match, you need to figure out which parts of the path are real and which are bogus. It often helps to obtain other copies of the same message from other servers (either Google or servers belonging to friends of yours) and compare the paths. Whatever part of the paths is consistent among the different news hosts probably contains the actual host at which the message was inserted.

Like mail hosts, Network News Transfer Protocol (NNTP) hosts may or may not accept posts from people outside of their organization. You can test this by telneting to port 119 on that host. If you cannot connect to that host or you receive a message that posts are not accepted, odds are lower that a bogus post was made using that host as the news server. However, if you receive a message that indicates you can connect, it means that other people can connect to that host also. The chances are reduced that the administrator will really know who is using the NNTP server. Such a session may look like this:

```
C:> telnet ferkel.piglet.com 119
200 NNTP Service Microsoft® Internet Services 5.00.7515
Version: 5.0.7515 Posting Allowed
```

or this:

```
$ telnet news.isp.com 119
200 mercury2.isp.com Netscape-Collabra/3.52 17222 NNRP ready (posting
ok).
```

But if the host does not accept posts, it may look something like this:

```
$ telnet nntp.mindspring.com 119
502 You are not in my access file. Goodbye.
```

NetBios

For historical reasons, Windows computers often use a protocol called NetBIOS. Although originally used only within LANs, NetBIOS has been extended so that it can run over TCP/IP, allowing organizations to provide Windows file- and print-sharing services across a WAN. A helpful command to identify a user over a network using NetBIOS is nbtstat. nbtstat is a standard component on all current Windows platforms, and a Linux version is also available.[8]

From your remote computer you can run this command against either the suspect's IP address:

```
nbtstat –a 123.456.789.000
or against a specific machine name:
nbtstat –A suspect.computer.com
```

nbtstat displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP). If the remote computer is reachable over the network, you can receive the following information:

NBTSTAT [-a RemoteName] [-A IP address] [-c] [-n] [-r] [-R] [-s] [S] [interval]		
-a	(adapter status)	Lists the remote machine's name table given its name
-A	(Adapter status)	Lists the remote machine's name table given its IP address.

`-c`	`(cache)`	Lists the remote name cache including the IP addresses
`-n`	`(names)`	Lists local NetBIOS names.
`-r`	`(resolved)`	Lists names resolved by broadcast and via WINS
`-R`	`(Reload)`	Purges and reloads the remote cache name table
`-S`	`(Sessions)`	Lists sessions table with the destination IP addresses
`-s`	`(sessions)`	Lists sessions table converting destination IP addresses to host names via the hosts file.
`RemoteName`		Remote host machine name.
`IP address:`		Dotted decimal representation of the IP address.
`Interval:`		Redisplays selected statistics, pausing interval seconds between each display. Press Ctrl+C to stop redisplaying statistics.

If a user is logged into the computer, you receive output similar to that shown in Figure 2-17. As you can see, it provides the machine name, the Windows NT domain the computer is registered in (in this case, a domain named "security"), and the MAC address. Since the MAC address is unique, it is a positive method of identifying a computer after it has been seized. Unless the NIC is swapped out, you have a promising lead that this is the computer you're looking for. nbtstat is a handy command because it enables you to associate a user with an IP address and then copy and paste that information into a document that you can print.

Figure 2-17 nbtstat output

We mostly use the nbtstat command from within our network since nbtstat issues a User Datagram Protocol (UPD) request and is blocked by default on many firewalls. Don't be surprised if you can ping the system, but an nbtstat returns "host not found" on a computer you know to be a Windows platform.

Third-Party Programs

If you have the budget, you can purchase additional programs that will do any of the previously discussed commands for you, usually with a convenient GUI. Two of the tools we use most often are Neotrace and Netscan Pro (the latter was shown previously in Figure 2-8). Both can do a traceroute for you, but Neotrace overlays a map and attempts to geographically plot the traceroute. This is a sexy feature, but its utility is questionable. It appears to map based on the zip code that the domain names are registered to. It would be more useful if the map could accurately show where the intermediate routers are located, but it should be apparent to you by now that this information is not available. Both programs are chock-full of capabilities. The NetScanTools Pro version obviously has a few more features, but for tracking purposes either will do nicely.

NetBIOS Tool

Essential NetTools (shown in Figure 2-18) is a set of network tools that are especially useful when investigating Microsoft networks. It includes

Figure 2-18 Essential NetTools

- **NBScan:** A fast multithreaded NetBIOS scanner for locating computers that are sharing resources on the network. Instead of having to manually check each system on the network individually (nbtstat –a 135.17.243.1, nbtstat –a 135.17.243.2, and so forth), simply enter a beginning address and an ending address and let NBScan run nbtstat for you.

- **NATShell:** A user-friendly interface for the popular NetBIOS Auditing Tool (NAT) network-auditing utilities.
- **NetStat:** Displays all of a computer's network connections and monitors external connections to a computer's shared resources.

You will find this handy set of utilities useful when you examine networks using Windows file-sharing or Samba on Unix.[9]

Whois Help

SmartWhois[10] automates the process of querying multiple whois databases and can retrieve information from more than 20 servers all over the world (see Figures 2-19 and 2-20).

Figure 2-19 Default and user-defined whois databases

Figure 2-20 Typical SmartWhois output

SmartWhois is designed to help inexperienced researchers find whatever information is registered on the Internet for specific IP addresses, so it comes with the most popular whois servers configured by default. Experienced users can define their own choices, making this a useful tool even for those who are accustomed to running whois from the command line.

Confirming Success

After you physically locate what you believe to be your suspect's computer, verify that its IP and machine name match that of your suspect. If the suspect computer is a Windows 9X computer, use winipcfg from either the Run box (accessible by choosing Start | Run) or from a command prompt. The initial output is shown in Figure 2-21. A drop-down window at the top of the IP configuration enables you to choose the network adapters that you want to display configuration for. In Figure 2-21, we are looking at the PPP (Point to Point Protocol) dial-up adapter, which is the default whenever PPP is configured. To access any of the other adapters, such as the network card, just drop down the list by clicking on the down arrow.

Figure 2-21 Initial winipcfg window

Whatever you do, *do not click on the Release All button!* If the computer is using DHCP, clicking on Release All wipes out the IP address, and depending on the network's configuration, you most likely will not get the same address back if you click on Renew All. Click the More Info button to access the information shown in Figure 2-22.

Figure 2-22 Extended winipcfg information

In the window shown in Figure 2-22, we not only see the IP address and the MAC address, but also the computer's name (in the Host Name field), and whether or not the computer is using DCHP (if static, meaning that the IP address was manually configured instead of being automatically obtained from the DHCP server, the Lease Obtained and Lease Expires fields will be blank). If there are IP addresses next to the Primary or the Secondary WINS Server fields, you can probably check them to see if the IP address was logged at the time of the incident.

Intrusion Detection Systems (IDS)

Intrusion detection systems, usually abbreviated IDS, are automated mechanisms that are intended to monitor specific subsystems, providing an alarm when a suspected unauthorized event is detected. Although IDS has many practical problems, the programs are increasingly popular, and in 2000, at least one Internet intruder was captured and brought to justice with the assistance of an IDS. At the time of this writing, IDS isn't necessarily a technology that every forensic technician needs to be familiar with, but during the next few years, it not only could become a standard tool in the detection of Internet intrusions, but could be routinely used to gather evidence used for successful prosecutions.

IDS is applied in one of two places: It is either network based or host based. Network-based systems are a specialized form of network sniffer. A network IDS sits on a network segment, viewing all traffic sent to every host on the segment, looking for evidence of unauthorized activity. It is common practice to have a single centralized intrusion detection engine that provides logging, and alarm functions based on the data provided by multiple remote sensors, each located on a different LAN segment. Even on switched networks, at least one port on the switch can usually be con-figured to provide all of the data being sent to each of the individual ports, which is where the sensor may be placed.

Host-based IDS places the detection capability on a single host, although the trend in host-based IDS is also to centralize the logging and alerting functions. Most business units resist having some other business unit looking over their shoulder and prefer not to have another organization place security software on their computers. Host-based IDS also tends to be more expensive because more devices have to be monitored. Although they are not as convenient to install and operate as network-based IDS, the host-based systems are generally more accurate, having fewer false positives and catching a higher percentage of actual misuses.

Intrusion detection systems are usually categorized as detecting either specific events or changes in patterns. Both types have their advantages and disadvantages. Event detection systems monitor for specific sequences of events, or sequences, that are characteristic of attempts to gain unauthorized access to a system. A simple example is a system that alerts when a specific number of failed login attempts have occurred. Commercial network-based IDS might alert when it detects a sequence of characters used to perform a buffer overflow attack against a Linux lpd daemon. These systems are dependent upon an up-to-date database of attack patterns. Although they may be able to log any arbitrary event type, they cannot alarm on events that are not in their database. Most commercial network IDS products at the time of this writing are of this type, and like the users of anti-virus software, the users of these products regularly download updated attack fingerprint databases from the publisher. Host-based IDS, such as the Tripwire product discussed in Chapter 11, work by regularly checking the consistency of system files, alerting whenever a security-relevant file has been changed. It is usually not practical to perform such a check on every host within an organization, but on those hosts that do have their files checked for consistency, intrusions are virtually always detected.

The other IDS model, the one that detects changes in patterns, is sort of an arti-ficial intelligence thing. The theory is that instead of limiting a detection engine just to the population of known attack types, you create a system that is sufficiently sophisticated to recognize anomalous behavior and alert whenever something happens that is outside of normal parameters. For instance, say a specific person normally accessed his or her account between the hours of 9 a.m. and 7 p.m. An IDS tracks and learns this normal behavior, and if the user were to access the account in the middle of the night, the IDS would notify the security administrator that an unusual event had occurred. Obviously, such a system is more prone to false alerts than one that is based on hostile event fingerprints, but it has the significant advantage of being able to detect brand-new attack forms. Research on such systems has been ongoing for at least ten years, and most commercial products rely on finger-print databases.

While the idea of capturing cyber criminals in the act should be an appealing one to most computer forensic types, widespread use of such products won't have a huge effect on what investigators do. The biggest advantage to the investigator is that IDS systems provide new and convenient forms of event logging. Once an attack or illegal activity is suspected, the logging or recording function on an IDS can be used to monitor and record the suspect's behavior.

Information Sources on IDS

If you are interested in doing some further research in IDS, several excellent books are available. Stephen Northcutt's, *Network Intrusion Detection* is the best hands-on guide for an analyst. In fact, it's a helpful book for a number of network security issues, and you should probably read it if you want to learn more about network protocol attacks and their analysis. Rebecca Bace's *Intrusion Detection* (Indianapolis: *Macmillan Technology Series,* 2000) is more theoretical, like a college textbook, making it a nice contrast to Northcutt's book. She describes the philosophy and architecture of IDS more comprehensively and provides a complete overview of the last ten years of relevant IDS research.

If you would like to use a network Intrusion Detection System but can't afford one of the commercial applications, you should take a look at Snort.[11]

Web Resources for Researching Internet Inhabitants

International Registries

Three international organizations are responsible for the administration of IP addresses within their region, so they should be considered definitive sources. Each of these organizations has a Web site that provides a whois interface, in addition to other information helpful in locating the owner of a specific IP address:

- **American Registry for Internet Numbers (ARIN):** Western Hemisphere, http://www.arin.net/whois/arinwhois.html
- **Asia Pacific Network Information Centre (APNIC):** Asia-Pacific, http://www.apnic.net
- **Reseaux IP Europeens (RIPE)**: Europe, http://www.ripe.net

Network Diagnostic and Research Sites

- **Adhoc IP Tools:** This site is a veritable Swiss Army knife of Internet tools, providing front ends to a wide variety of research services (whois, nslookup, ping, DNS dig, and others), all accessed from a single page: http://home.ag.org/iptools.htm.
- **Sam Spade:** Also provides a wide variety of research tools, http://www.samspade.org.
- **Internet Service Provider lookup:** Enables you to search for ISPs by name, providing a summary of their business characteristics, http://www.webisplist.com.
- **Dragon Star:** Provides an index, relating IP network numbers to network names and identities. It also includes a handy explanation of the IP address numbering scheme and describes the difference between Class A, B, and C networks, http://ipindex.dragonstar.net/index.html.

News and Email Abuse Information

- **The spamfaq or "Figuring out fake Email & Posts":** This site, maintained by Gandalf@digital.net*,* is the most comprehensive source we're aware of. It has detailed instructions on how to track both email and news, how to read the message headers in a dozen different mail clients, and how to reach the appropriate abuse contact. It also has a huge number of additional links. It isn't edited well, but it is worth your time if you really need to understand message headers, http://ddi.digital.net/~gandalf/spamfaq.html.
- **Fighting Email Spammers:** A site maintained by Todd Burgess, it is an excellent source of information on tracking email, http://eddie.cis.uoguelph.ca/tburgess/local/spam.html.
- **Fight Spam on the Internet!:** Another site with a number of links on the subject of unsolicited email, http://spam.abuse.net/.
- **Reading EMail Headers:** A detailed explanation of the function of dozens of different email headers, http://www.stopspam.org/email/headers/headers.html.

800 East 96th Street Indianapolis, Indiana 46240