Editors: Michael Howard, michael.howard@microsoft.com
James A. Whittaker, jw@se.fit.edu

Network Security Basics

Writing a basic article on network security is something like writing a brief introduction to flying a commercial airliner. Much must be omitted, and an optimistic goal is to enable the reader to appreciate the skills required.

Gerald A. Marin
Florida Institute of Technology

The first question to address is what we mean by "network security." Several possible fields of endeavor come to mind within this broad topic, and each is worthy of a lengthy article. To begin, virtually all the security policy issues raised in Matt Bishop's book, *Computer Security Art and Science*,[1] apply to network as well as general computer security considerations. In fact, viewed from this perspective, network security is a subset of computer security.

The art and science of cryptography and its role in providing confidentiality, integrity, and authentication represents another distinct focus even though it's an integral feature of network security policy. Readers looking for a good introduction (and more) to this area should consider *Practical Cryptography* by Niels Ferguson and Bruce Schneier.[2]

The topic also includes design and configuration issues for both network-perimeter and computer system security. References in this area include Stephen Northcutt and colleagues' *Inside Network Perimeter Security*,[3] the classic *Firewalls and Network Security*[4] by Steven Bellovin and William Cheswick, and too many specific system configuration texts to list. These are merely starting points for the interested novice.

The practical networking aspects of security include computer intrusion detection, traffic analysis, and network monitoring. This article focuses on these aspects because they principally entail a networking perspective.

Network traffic

To analyze network traffic, we need a basic understanding of its composition. In this regard, networking people often speak of *flows* and *formats*. Flow is a laconic reference to networking protocols and the messages that travel back and forth between their endpoints. Format refers to the structure of the cells, frames, packets, datagrams, and segments (the awkward generic term is *protocol data units*) that comprise the flow.

The vast majority of network traffic today uses the Internet Protocol (IP) as its network-layer protocol.[5] IP addresses represent sources and destinations, and IP routers work together to forward traffic between them. Link-layer protocols such as Ethernet (IEEE 802.3), token ring, frame relay, and asynchronous transfer mode (ATM) forward IP packets, called datagrams, across many types of links.

Networks can be attacked at multiple layers; here, I focus on the network layer and the layer above it (the transport layer). The Internet network layer is "unreliable," meaning it doesn't guarantee end-to-end data delivery. To get reliable end-to-end service, a user invokes the Transport Control Protocol (TCP).

Figure 1 shows the format for an IP datagram; Figure 2 shows the format for a TCP segment, which is the protocol data unit associated with the TCP protocol. These formats are essential for understanding network traffic composition and something of the methods that can be used to corrupt them.

TCP/IP traffic accounts for much of the traffic on the Internet (although TCP isn't typically used for voice or video traffic). Figure 3 illustrates how a tool such as Ethereal (www.ethereal.com) can help capture and analyze traffic.

We now have a fairly representative picture of the traffic flowing across the Internet. It consists of IP datagrams (which can be carried inside link-layer frames, for example) carrying higher-layer information, often including TCP segments.

Those with malicious intent could misuse any of the fields shown in Figures 1 and 2. The attackers would know the protocol's intent and the rules to use to interpret the associated formats and flows. They can create a networking attack by changing values in any of the fields—any ensuing problems constitute attacks on the network. *Spoofing*, or changing the source address, lets an attacker disguise malicious traffic's origin.

Network intrusions

Typical network traffic consists of millions of packets per second being exchanged among hosts on a

LAN and between hosts on the LAN and other hosts on the Internet that can be reached via routers. Network intrusions consist of packets that are introduced specifically to cause problems for any of the following reasons:

- to consume resources uselessly,
- to interfere with any system resource's intended function, or
- to gain system knowledge that can be exploited in later attacks.

The simplest example of a network intrusion is probably the *land* attack. Some early IP implementations failed to take into account that a datagram might be generated with identical source and destination IP addresses. Some older operating systems (and perhaps unpatched ones) simply crashed if they received such datagrams.

Somewhat more complicated is the *smurf* attack in which an attacker spoofs the source address and sets it equal to the targeted machine's address. The attacker then broadcasts an echo request to perhaps hundreds of machines on distant networks—a capability provided by the Internet Control Message Protocol (ICMP). Each distant machine responds to the received echo request with an echo response message to the targeted IP address, thus overwhelming the targeted machine's resources.

The *teardrop* attack is somewhat more sophisticated in its use of the header fields shown in Figure 1. IP version 4 (IPv4) can break large datagrams into sequences of smaller IP datagrams through a process referred to as *fragmentation*. It uses certain bit flags and the `fragment offset` field to ensure that the fragments can be reassembled at the destination (see Figure 1). In a teardrop attack, an attacker sends fragments that are purposely made to overlap so that they don't fit together properly at the destination. Again, older (or unpatched) operating systems could have severe problems with such fragments.

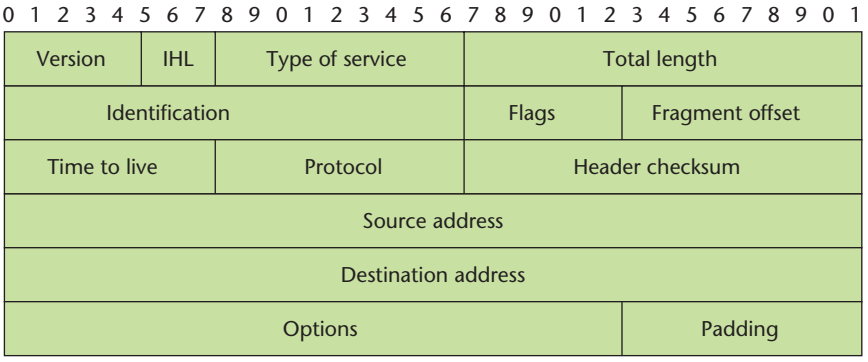

Figure 1. Internet datagram header format. As defined in RFC 791, Internet datagrams running under version 4 of the Internet Protocol (IPv4) carry most of today's Internet traffic, although a newer version has been defined as IPv6. (The numbers across the top indicate bit positions.)

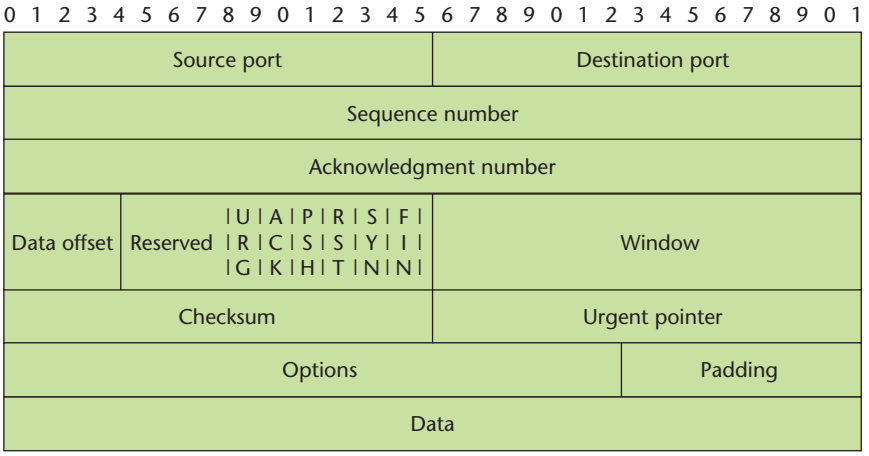

Figure 2. Transport Control Protocol header format. As defined in RFC 793, TCP provides a reliable end-to-end transport service across the unreliable Internet.

DDoS attacks

In February 2000, hackers attacked several high-profile Web sites, including Amazon.com, Buy.com, CNN Interactive, and eBay, by sending large numbers of bogus packets with the intent of slowing or interrupting offered services.[6] Many articles have since examined these attacks and potential defenses, and several Web sites offer overviews, case histories, suggested defenses, and other resources. In spite of all the work done in this area, the threat of DoS attacks remains, as high-profile attacks described periodically in the networking trade press will attest.

Typically, a hacker launches a distributed denial-of-service (DDoS) attack by issuing commands to "attack zombie" computer programs that have penetrated unsuspecting users' machines via the Internet—perhaps propagated by viruses or worms, for example. Once present, the zombies allow hackers to leverage user machines as part of an attack against a given target. Note that the

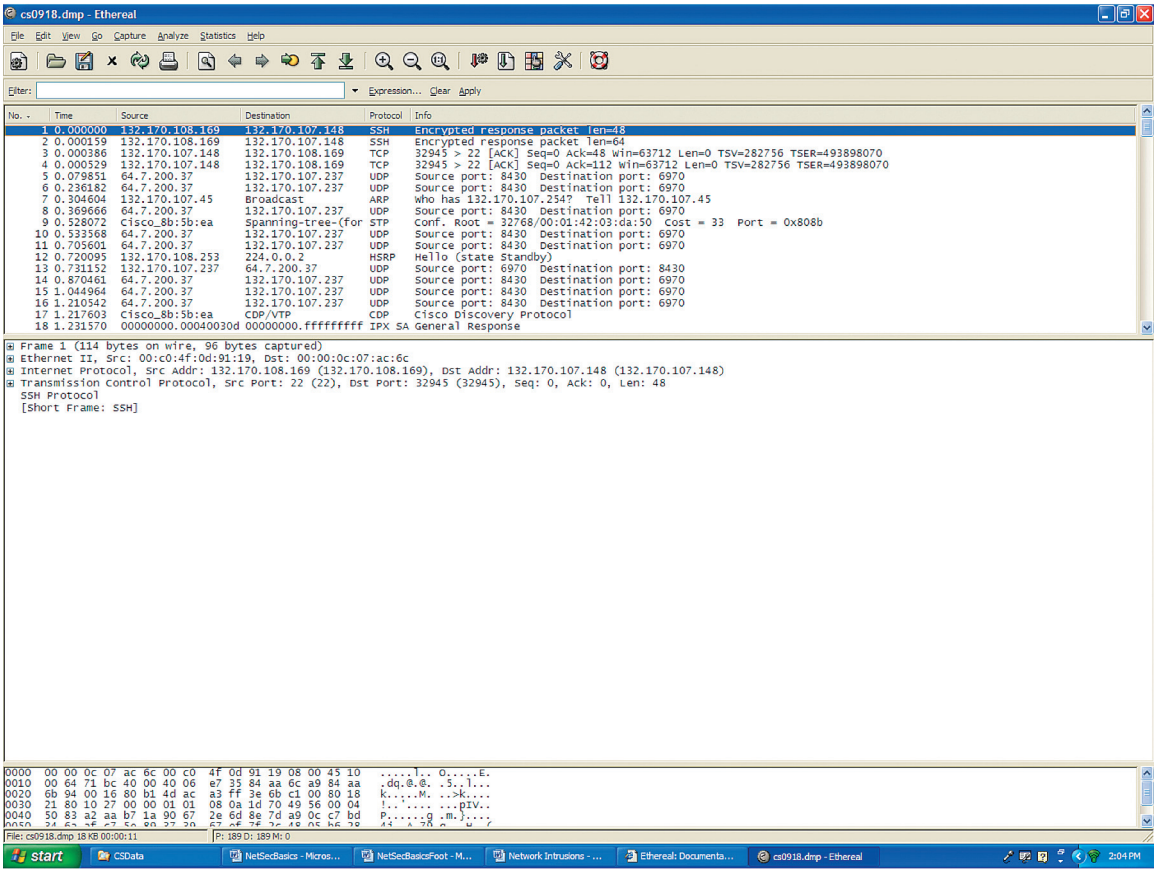

Figure 3. Example traffic-analysis output. This screenshot from the Ethereal tool shows a list of 18 packets. The middle section describes the highlighted packet; the third section displays the packet in hex format. Ethereal is open-source software released under the GNU General Public License.

generated traffic might seem to be normal Web browser requests and other innocent-looking traffic that, in fact, differs from valid traffic principally in its intent. This makes identifying such attacks extremely difficult. For particularly interesting reading, Steve Gibson provides a case history of one of the early DDoS attacks.[7]

Intrusion detection systems

No single technique is likely to detect all possible types of network intrusions—especially because new intrusion types are still waiting to be exploited. Reviewing the attacks described here, it's clear that land attacks can be discovered by looking for arriving packets in which the source and destination IP addresses are identical. Smurf attacks can't be detected on the basis of content from single packets; only the arrival of an unusually large number of ICMP echo requests and responses would signal such an attack's presence. We could respond by killing all echo requests at a gateway router, but doing so would interfere with other network functions that might be vital to the organization being protected. We might discover the teardrop attack by looking for illegal fragmentation in arriving packet trains, but the router (or firewall) would have to maintain a significant amount of state information.

Intrusion detection systems (IDSs) use particular collections of analytical techniques to detect attacks, identify their sources, alert network administrators, and possibly mitigate an attack's effects. An IDS uses one or both of the following techniques to detect intrusions:

- Signature detection—the IDS scans packets or audit logs to look for specific signatures (sequences of commands or events) that were previously determined to indicate a given attack's presence.
- Anomaly detection—the IDS uses its knowledge of behavior patterns that might indicate malicious activity and analyzes past activities to determine whether observed behaviors are normal.

It's fairly easy to understand how signature detection can help find

identifying characteristics in previously observed attacks. This is far from simple to accomplish, however, because attackers can change some identifier (a port number, a particular sequence number, a particular protocol indicator) that alters the signature without affecting the attack's fundamental nature. Moreover, someone constructing an alert based on signature detection must be mindful that normal traffic could have the same characteristics. A useful signature must reflect a reliable attack identifier that doesn't generate many alerts on nonmalicious traffic. With the huge number of packets arriving at most modern subnets, even a miniscule error rate could generate tens of thousands of false alarms within a few minutes.

Several commercial and a few public IDSs are available. The trade press frequently evaluates them, but research journals generally do not. Early IDSs largely used signature detection. Generally speaking, they detected all the attacks captured in their signature databases, but they suffered from unacceptably high false-alarm rates.[8] More innovative approaches have appeared recently, including behavior-based modeling.[9]

To clarify how traffic or behavioral anomalies can be used to identify attack traffic for attacks that haven't been seen before, consider the following example. IP addresses generally suffice to enable a datagram to reach its intended destination machine, but many processes typically run at once on any given machine. TCP/IP uses *port numbers* to distinguish among them. A security analyst might be able to analyze daily or hourly patterns in the use of source addresses, destination addresses, and both source and destination port numbers to determine when a pattern change suggests possible malicious activity. (We must be careful to observe that "different" doesn't always imply "evil.") In Figure 4, for example, we see port activity displayed from data produced at the Lincoln Labs at the Massachusetts Institute of Technology (MIT) for a particular subnet over a 10-hour period.[10,11] This often-used data set includes data with and without attacks present, which are difficult to obtain on "live" networks. (Data are from Monday of week five in the Lincoln Lab data.) Figure 5 shows the result when we remove all the port activity found during a similar 10-hour period from an attack-free data set: three areas clearly represent unusual (or anomalous) port activity. Further investigation reveals that these are, indeed, attacks—in this case, inserted by MIT researchers.

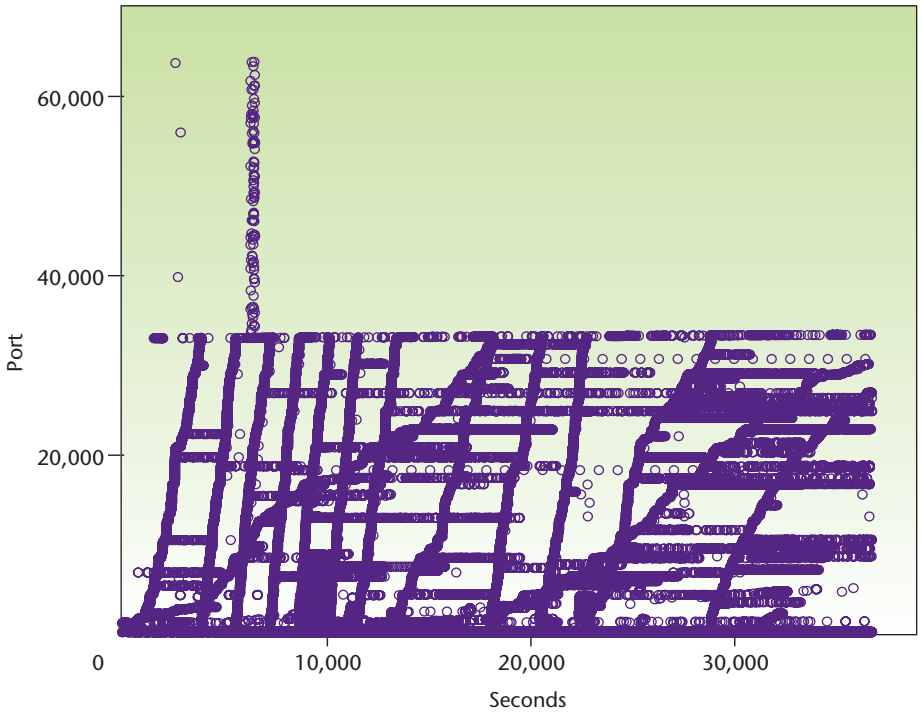

Figure 4. Port usage at MIT's Lincoln Lab. This data set illustrates patterns in the use of source and destination ports over a 10-hour period. Dots indicate the use of a port at a particular moment in time.

Researchers have applied many other techniques to detecting traffic anomalies including data mining, statistical analysis, artificial intelligence, neural networks, Markov modeling, sensor correlation, and analysis of management information data. It's safe to say that the ultimate solution remains to be found.

Although intrusion detection is a good place to start "basic training," we should note that network security people are probably more concerned about worms, viruses, and spam; they worry at least as much about active methods to combat these pests as they do about IDSs. Network worms seek to exploit software weaknesses on servers that must keep particular ports open to provide service. If a worm succeeds in penetrating the network perimeter security, it can introduce Trojan code that changes the target machine in ways that users won't detect. At present, therefore, detecting the presence of malicious traffic from outside the network probably doesn't worry network administrators as much as the likelihood that Trojans and spyware might already

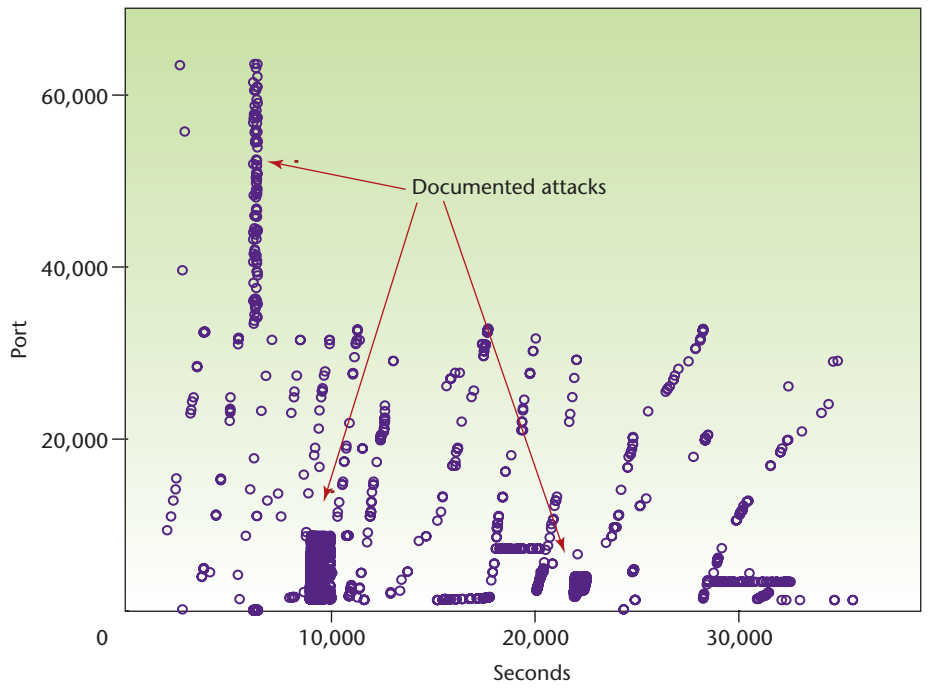

Figure 5. Anomalous port activity on the Lincoln Lab machines. Subtracting all `(time,port)` pairs that were active during the base comparison period in Figure 4 shows three areas that represent unusual port activity, which could be attacks.

reside in internal machines that access sensitive data.

Techniques for detecting malicious code bring us back to general computer security issues and methods. Analysis of network activity associated with problems such as worm infections could complement other system security work in determining which machines are infected. Based on both traffic analysis and system behavioral analysis, for example, sufficiently suspicious machines might be isolated from their peers via (perhaps new) security protocols until administrators took steps to secure them. Whether such isolation can be accomplished before a critical subset of the Internet becomes infected is one concern of current and future research. There are others, and they also depend, to some extent, on the basics covered in this article. ▢

References

1. M. Bishop, *Computer Security Art and Science*, Pearson Education, 2003.
2. N. Ferguson and B. Schneier, *Practical Cryptography*, John Wiley & Sons, 2003.
3. S. Northcutt et al., *Inside Network Perimeter Security*, New Riders Publishing, 2003.
4. S. Bellovin and R.W. Cheswick, *Firewalls and Internet Security: Repelling the Wily Hacker*, Pearson Education, 1994.
5. *Internet Protocol*, RFC 791, Sept. 1981; www.ietf.org/rfc/rfc791.txt.
6. S. Bonisteel, "Yahoo DoS Attack Was Sophisticated," Computer User.com, 4 April 2003; www.computeruser.com/news/00/02/14/news1.html.
7. S. Gibson, "The Strange Tale of the Denial of Service Attacks Against grc.com," Gibson Research, 2002; http://grc.com/dos/grcdos.htm.
8. D. Newman, J. Snyder, and R. Thayer, "Crying Wolf: False Alarms Hide Attacks," *Network World*, 24 June 2002; www.networkworld.com/techinsider/2002/0624security1.html.
9. R. Thayer, "Intrusion Detection Systems," *Network World*, 31 Jan. 2005; www.networkworld.com/reviews/2005/013105rev.html.
10. J. Haines et al., *1999 DARPA Intrusion Detection Evaluation: Design and Procedures*, Lincoln Lab tech. report 1062, Massachusetts Inst. Technology, 2001.
11. J. Haines, L. Rossey, and R. Lippman, "Extending the DARPA Off-Line Intrusion Detection Evaluations," *Proc. IEEE/DARPA Information Survivability Conf. and Exposition* (DISCEXII), vol. I, vol. 1, IEEE CS Press, 2001, p. 0035.

Gerald A. Marin *is a professor at the Florida Institute of Technology. His research interests include computer communication networks, system and network performance, system and network security, and simulation modeling. Marin has a PhD in mathematics from North Carolina State University. He has several years of industry experience, both with IBM and the Center for Naval Analyses. Contact him at gmarin@fit.edu.*