295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 1

Chapter 1

Intrusion Detection Systems

Solutions in this Chapter:

- Introducing Intrusion Detection Systems
- Answering Common IDS Questions
- Fitting Snort into Your Security Architecture
- Determining Your IDS Design and Configuration
- Defining IDS Terminology

☑ Summary

☑ Solutions Fast Track

☑ Frequently Asked Questions

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 2

Introducing Intrusion Detection Systems

It's three o'clock in the morning, and Andy Attacker is hard at work. With the results from the latest round of portscans at hand, Andy targets the servers that appear vulnerable. Service by service, Andy fires off exploits, attempting to overflow buffers and overwrite pointers, aiming at taking over other peoples' servers. Some of these attempts are successful. Encouraged, Andy quickly installs rootkits on the compromised machines, opening backdoor access mechanisms, securing the machines enough to lock other attackers out, and consolidating control. Once that is accomplished, Andy begins the next round of scan-and-exploit, from the newly compromised machines.

It's three o'clock in the morning, and a shrill insistent beeping rouses Jennifer Sysadmin from her bed. Blearily, she finds her pager on the nightstand and stares at the message it displays. A customized message alerts her to a Secure Shell overflow attempt… outbound from one of her servers. She is startled into wakefulness. Throwing back the covers and grumbling about the tendency of network malefactors to attack during prime sleeping hours, she grabs her cell phone and heads purposefully for the nearest computer.

It's three o'clock in the morning, and across town, Bob Sysadmin is sleeping peacefully. No pager or cell phone disturbs his rest.

Is Bob's security that much better than Jennifer's, so that he can sleep soundly while she cusses and does damage control? Or has he also been compromised and just doesn't know it yet? With only this information, we don't know. And if he doesn't have an Intrusion Detection System (IDS), neither does Bob. IDSs are a weapon in the arsenal of system administrators, network administrators, and security professionals, allowing real-time reporting of suspicious and malicious system and network activity. While they are not perfect and will not show you every possible attack, IDSs can provide much-needed intelligence about what's really going on on your hosts and your network.

What Is an Intrusion?

To understand what "intrusion detection" does, it is first necessary to understand what an intrusion is. Webster's dictionary defines an intrusion as "the act of thrusting in, or of entering into a place or state without invitation, right, or welcome." For our purposes, an intrusion is simply unauthorized system or network activity on one of your computers or networks. This can take the form of a legitimate user of a system trying to escalate his privileges and gain greater access to

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 3

the system than he has been allowed, a remote and unauthenticated user trying to compromise a running service in order to create an account on a system, a virus running rampant through your e-mail system, or many other similar scenarios. Intrusions can come from the deliberately malicious Andy Attackers of the world, or from the terribly clueless Archibald Endusers of the world, who will click on every e-mail attachment sent to them, despite repeated admonitions not to do so. Intrusions can come from a total stranger three continents away, from a disgruntled ex-employee across town, or from your own trusted staff.

OINK!

Detecting malicious activity when it comes from your own employees or users is one of the most important purposes for IDSs in many environments. In fact, a properly implemented IDS that is watched by someone besides your system administrators may be one of the few methods that can actually catch a system administrator when she is doing something malicious. This is one of the main reasons why you should have network security personnel analyzing IDS events and system administrators managing systems.

Legal Definitions

Legally, there are not clear and universal standards for what constitutes an intrusion. There are federal laws about computer crime in many countries, such as the United States and Australia, but none in others. There are various state laws, and regional statutes in place, but not everywhere. Jurisdiction for computer crime cases can be unclear, especially when the laws of the attacker's location are vastly different from the laws in place in the compromised machine's region. To add to this confusion even if an intrusion is clearly within the legal definitions, many law enforcement agencies will not spend time working on it unless there is a clear dollar cost that is greater than some fixed amount. Some agencies use US$10,000 for their guideline, while others use US$100,000—this number varies from place to place.

Another legal concern when using IDSs is privacy. Technically, an IDS is a full content wiretap. In the United States, full content wiretaps are regulated by federal laws, including Title III of the Omnibus Crime Control and Safe Streets Act of 1968 (Title III), 18 U.S.C. §§ 2510–2522 and the Electronic

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 4

Communications Privacy Act of 1986. They are also subject to less stringent laws governing Pen Registers or Trap and Trace situations, such as the Pen Register, Trap and Trace Statue "Provider Exception," 18 U.S.C. § 2511(2)(h). These generally involve tapping the characteristics and patterns of traffic without examining the data payload. Under these laws, intercepting network data may be illegal, particularly if it is not done by the network operator in the pursuit of his normal duties or in direct support of an ongoing criminal investigation of a computer trespasser. We strongly advise that you consult your legal department about your particular jurisdiction's laws and the ramifications of deploying an IDS on your network.

Some enterprises rely on the status of their data as "protected trade secrets" under local common uniform trade secrets statutes. Such laws usually require the data to not be known to the public at large, and for some efforts to have been made to secure the data. Therefore, if you're relying on such laws to save you when your data is stolen, you may be in for a nasty shock if the court deems your security measures insufficient. However, the U.S. Economic Espionage Act of 1996 (viewable at www.cybercrime.gov/eea.html) can make such activity a federal crime.

The type and scope of the activity can affect this as well. In computer security forums, there are often arguments about whether portscanning is legal. The answer depends on your jurisdiction. In 1998, Norway ruled that portscanning was not illegal. Michigan law, however, states that unauthorized use or access of a computer is illegal unless you have reason to think the system is designed for public access. Lawyers are still arguing about whether portscanning is "unauthorized use." In some jurisdictions, login banners explicitly prohibiting access are required to prove that a given use of the system was unauthorized. Privacy expectations can play into the equation, too—if the user has an expectation that her system activity may be private, logging and prosecuting her for that activity may be difficult even if it is obviously malicious.

The best practices solution to this legal morass is usually to secure your systems as much as possible, clearly label all accessible services with login banners stating the terms of use, and know your local and federal computer crime laws, if there are any. That will help you protect your systems and identify what is considered an intrusion in your jurisdiction.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 5

Scanning vs. Compromise

When watching network activity, one of the first things that usually jumps out is scanning activity; specifically, lots of scanning activity. Whether it's scanning for particular vulnerabilities or just scanning for open ports, this type of activity is very common on the unfiltered Internet, and on many private networks. Many IDSs are configured to flag scanning activity, and it's not uncommon to see the bulk of your alerts be caused by some form of scanning. While scanning is not necessarily malicious activity in and of itself, and may have legitimate causes (a local system administrator checking his own network for vulnerabilities prior to patching, for example, or a third-party company hired to perform a security audit of your systems), very often scanning is the prelude to an attempted attack. As such, many administrators want to be alerted when they are being scanned. Tracking scanning activity can also be useful for correlation in case of later attack.

Many popular network scanning tools are free, and freely available. A quick Google search will turn up everything from the ping and File Transfer Protocol (FTP) "Grim's Ping" to the full-featured portscanner Nmap, from the commercially available SolarWinds scanner to the vulnerability scanner Nessus. Since scanning tools are so easily accessible, it's not that surprising that they are so widely used.

However, it is important to realize that scanning is not an attempted compromise in and of itself, and should not be treated with the same level of escalated response that an actual attempted attack would merit. There are people who just scan systems out of curiosity and do not intend to attack them. A fellow that we met at a security conference once confided that before he engages in online financial transactions with any business, he scans all the company's machines that he can find. That's his way of determining whether he feels he trusts their security enough to trust them with his money.

It's also important to note that scanning activity is nearly constant. On the Wild West of the modern Internet, all sorts of automated programs are scanning large ranges of addresses, all the time. Some of them might be yours. Network monitoring tools, worms and viruses, automated optimization applications, script kiddies, and more are constantly probing your machines and your network. If you don't make a deliberate effort to filter it out, seeing this traffic on the Internet is a fact of life.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 6

OINK!

While it is important to know when your network is being scanned, you don't want to make the mistake of spending your valuable time tracking down every fool who appears to be scanning your network. One of the best things you can do with information about scans is to track the source IPs that are scanning you and then use them to correlate against alerts for higher priority events or look for repeat scanners. We talk about correlation methods and data analysis in depth in Chapter 8, "Dealing with the Data."

Viruses and Worms—SQL Slammer

Now that we've discussed scanning activity, let's get into a little more detail about some of the actual attempted compromises out there. Another very common type of traffic that you'll see triggering your IDSs is automated worms. Worms and viruses are often good candidates for IDSs, because they have repeatable and consistently identifiable behavior. Even polymorphic worms and viruses that attempt many attack vectors will have some network behavior in common, some traffic pattern that can be matched and detected by your IDS. As an example, let's look at the SQL Slammer worm.

On January 25, 2003, the SQL Slammer worm was released into the wild. Also known as Sapphire, the worm exploits a weakness in the Microsoft Structured Query Language (SQL) server. It sends a 376-byte User Datagram Protocol (UDP) packet to port 1434, overflows a buffer on the SQL server, and gains SYSTEM privileges, the highest possible level of compromise on a Windows operating system. Once it has successfully compromised a host, it starts scanning other IP addresses to further spread.

OINK!

Worms that use multiple attack paths are an excellent example of the value of correlation. The individual alerts from CodeRed or Nimda are common enough, but when they are seen together (as they would be from CodeRed or Nimda), they are a very distinct fingerprint for that worm. As mentioned before, we discuss correlation more in Chapter 8.

It is also worth noting that SQL Slammer is a perfect example of a situation where an "active response" IDS would not be able to prevent

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 7

infection, but an inline IDS would. The pluses and minuses of "active response" vs. inline IDS are discussed in Chapter 12, "Active Response."

From the moment of its release, it is estimated that the worm spread worldwide in approximately 10 minutes. Massive amounts of network bandwidth were chewed up by the worm's scanning and propagation attempts. Many systems were compromised. Five of the 13 root Domain Name servers that provide name service to the Internet were knocked down by the worm. You can read the Microsoft advisory about the worm at www.microsoft.com/technet/treeview/default.asp?url=/technet/security/alerts/slammer.asp, and the Computer Emergency Response Team Coordination Center's (CERT-CC) advisory about the worm at www.cert.org/advisories/CA-2003-04.html.

OINK!

The CERT/CC is a center of Internet security expertise located at the Software Engineering Institute, a federally funded research and development center operated by Carnegie-Mellon University.

So, what's a good candidate rule for catching this with an IDS? Obviously, this is just the type of activity that you want to detect on your network. One thing common among every Slammer-infected host is the exploit payload it sends out. And indeed, that's exactly what the Snort IDS signature for the rule matches against. Here's the Snort signature that matches this activity:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1434 (msg:"MS-SQL Worm propagation
attempt"; content:"|04|"; depth:1; content:"|81 F1 03 01 04 9B 81 F1 01|";
content:"sock"; content:"send"; reference:bugtraq,5310; classtype:misc-
attack; reference:bugtraq,5311;
reference:url,vil.nai.com/vil/content/v_99992.htm; sid:2003; rev:2;)
```

We'll get into much greater detail about Snort rules and their construction in Chapter 5, "Playing by the Rules," but you can see that the alert is labeled as an attempt at worm propagation, and that it matches UDP traffic headed to our network $HOME_NET on port 1434 with a specific payload. Using this signature, we can detect and enumerate how many attack attempts we saw, and what hosts on our network they were attempting to reach. Massive automated attacks

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 8

like this one usually engender a coordinated response from the security community—IDS programmers writing new signatures, antivirus vendors writing checks and fixes, backbone providers tracking the traffic and mitigating its effect by filtering as requested and as needed. This signature can help us track infection attempts by the worm on our network, and make sure that our systems under attack remain secure. Coordinating responses between companies and defenders is one of the few ways we can keep up with the attackers. A large number of organizations are dedicated to helping responders deal with attacks and share information.

OINK!

Here are some of the many organizations chartered to help mitigate attacks:

- The Forum of Incident Response and Security Teams, also known as FIRST, is a cluster of security professionals at various organizations. Membership is restricted to eligible teams with a clear charter and organizational scope, sponsored by an existing team, and capable of conducting secure communications with PGP.
- Information Sharing and Analysis Centers, or ISACs, were chartered in the United States in 1998 under the PDD 63, Protecting America's Critical Infrastructure policy. ISACs cover areas as diverse as electricity, financial services, drinking water, and surface transportation, but the most relevant ISAC for network security is the Information Technology ISAC, online at www.it-isac.org/.
- The Distributed Intrusion Detection System Dshield correlates firewall logs and reports of network attacks worldwide. Anyone can join, or submit his or her logfiles for analysis anonymously. Membership is free.
- Many commercial offerings will outsource your network security, firewall and IDS administration, log analysis, and attack correlation for you. Some providers will correlate data between their customers to increase the likelihood of detecting loud and active attackers, others will not. Specifics of the offered services depend on the vendor.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 9

Live Attacks—Sendmail Buffer Overflow

We have seen what an IDS can do to let you know about an automated attack. However, what about attacks that are driven by a person, one single attempt at overflowing a network service rather than a virtual flood of packets? Snort can help with that, too. Let's look at an exploitable vulnerability, the Wingate POP3 buffer overflow.

The vulnerability is a remotely exploitable buffer overflow in the Wingate implementation of the POP3 daemon. After the USER command is sent, a sufficiently large amount of data following "USER" will overrun the buffer and may possibly lead to executing whatever exploit code is inserted. Normal use of the POP3 daemon would just supply a username after the USER command, and a normal username is unlikely to be very long. Now, let's look at the Snort rule that detects this attempted exploit:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 110 (msg:"POP3 USER overflow
attempt"; flow:to_server,established; content:"USER"; nocase;
isdataat:50,relative; pcre:"/^USER\s[^\n]{50,}/smi"; reference:bugtraq,789;
reference:cve,CVE-1999-0494; reference:nessus,10311; classtype:attempted-
admin; sid:1866; rev:7;)
```

This rule looks for data with the content USER followed by more than 50 bytes of data, where those 50 bytes of data after USER don't contain a newline character. This should match the pattern of data we'd see in a real attempt at overflowing this buffer, and should not match legitimate user logins.

Again, we describe Snort rules and how to configure them to alert optimally for your network in much more detail in later chapters.

How an IDS Works

Now that we have looked at some of the capabilities of an IDS as far as alerting on malicious traffic, it's time to take a closer look at what exactly IDSs can keep an eye on, what data sources they use to do this monitoring, how they separate attack traffic from normal traffic, and some possible responses to seeing malicious traffic.

What the IDS Is Watching

Let's start by looking at what your IDS is able to see. This is going to depend greatly on what type of IDS it is, and where it's placed in your network. IDSs are classified by their functionality, loosely grouped into the following three categories:

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 10

- Network-Based Intrusion Detection System (NIDS)
- Host-Based Intrusion Detection System (HIDS)
- Distributed Intrusion Detection System (DIDS)

Network IDS

The NIDS derives its name from the fact that it monitors an entire network segment, or subnet. This is done by changing the mode on the NIDS' network interface card (NIC). Normally, a NIC operates in nonpromiscuous mode, listening only for packets destined for its own media access control (MAC) address. Other packets are not forwarded up the stack for analysis; they are ignored. To monitor all traffic on the subnet, not just those packets addressed to the NIDS machine itself, the NIDS must accept all packets and forward them up the stack. This is known as promiscuous mode.

In promiscuous mode, the NIDS can eavesdrop on all communications on the network segment. However, that's not all that is necessary to ensure that your NIDS is capable of listening to all traffic on the subnet. The network device immediately upstream of your NIDS must also be configured to send all packets on the subnet to your NIDS. If that device is a hub, all packets are automatically sent since all ports on a hub receive all traffic flowing through the hub. However, if that device is a switch, you may have to put the port on the switch into a monitoring mode, turning it into a span port. After setting up your NIDS, it is advisable to run a sniffing tool on the interface, to ensure that you can see all traffic on the subnet.

The advantage of a NIDS is that it has no impact on the systems or networks it is monitoring. It doesn't add any load to the hosts, and an attacker who compromises one of the systems being watched can't touch the NIDS and may not even know it is there. One downside of the monitoring is maxing out your span ports that you are allotted on a given network, and maxing out the bandwidth on the span itself. If you have 20 100MB ports spanning to one port, you begin filling up backplane… once that 5GB or 11GB backplane is saturated, you are in a world of hurt.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 11

Tools & Traps…

Network Sniffing Tools

When setting up or debugging a NIDS, it is vital to ensure that you are seeing all the traffic for the subnet to which you are connected. Snort is capable of functioning as a fine packet sniffer. When invoked from the command line with the **–i** switch, Snort will listen on a particular interface. Make sure you see traffic from and to other machines on the network, not just the broadcast traffic and the traffic to the local machine.

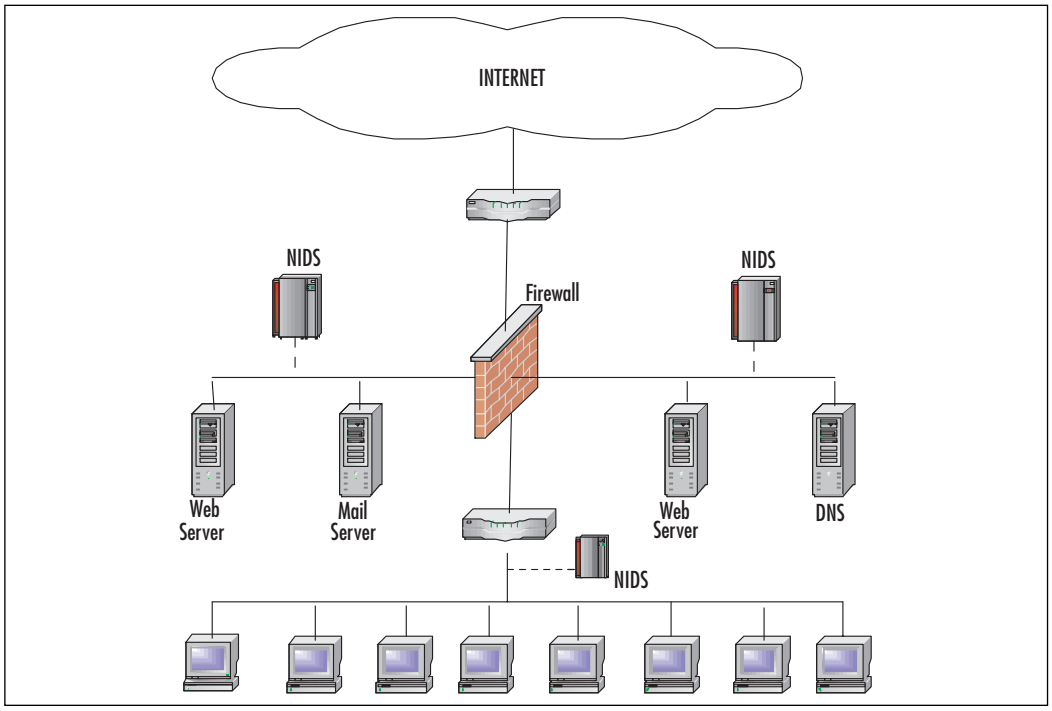
In addition to Snort, several other programs are perfectly good packet sniffers. Ethereal, available from www.ethereal.com, is a cross-platform packet sniffer. Tcpdump (www.tcpdump.com) is present on many Unix systems already, and Windump (http://windump.polito.it) serves the same function for Windows, although it usually will have to be installed on the system.

In view of emerging privacy regulations, monitoring network communications is a responsibility that must be considered carefully. Make sure that you are familiar with your local legal requirements for such activity.

In Figure 1.1, we see a network using three NIDS. The units have been placed on strategic network segments and can monitor network traffic for all devices on the segment. This configuration represents a standard perimeter security network topology where the screened subnets housing the public servers are protected by NIDSs. When a public server is compromised on a screened subnet, the server can become a launching platform for additional exploits. Careful monitoring is necessary to prevent further damage.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 12

Figure 1.1 NIDS Network



The internal host systems are protected by an additional NIDS to mitigate exposure to internal compromise. The use of multiple NIDS within a network is an example of a defense-in-depth security architecture.

OINK!

In case you missed it, let's say that again—privacy regulations can be a dangerous trap. Even if you have your users sign an Acceptable Use Policy that stipulates you have the right to watch them, there may still be situations where they can claim an assumption of privacy. Be sure to get approval from your management (if you are the one deploying the IDS), or your Human Resources department (if your company has one), or as a last resort, talk to your lawyer and make sure you aren't violating any laws. If you do this incorrectly, you may find that *you* are being prosecuted instead of the person you were trying to monitor! The PATRIOT Act, despite its many critics, does appear to grant the service provider and system administrators the ability to monitor the use of their networks and systems for the purpose of identifying misuse.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 13

Careful consideration must be paid to who sees the data, and to the process of keeping that data secure. Finally, remember that any legal advice given in this book is not offered by a lawyer—you should check it with your own before depending on it.

Host-Based IDS

Host-based IDSs, or HIDSs, differ from NIDSs in two ways. First, an installed HIDS protects only the system on which it resides, not the entire subnet, and second, the network card of a system with a HIDS installed normally operates in nonpromiscuous mode. This can be an advantage in some cases—not all NICs are capable of promiscuous mode, although most modern NICs are. In addition, promiscuous mode can be CPU intensive for a slow host machine.

Another advantage of HIDS is the ability to tailor the ruleset to be very specific to the particular host system. For example, there is no need to configure multiple rules designed to detect Network File System (NFS) exploits on a host that is not using the NFS. Being able to fine-tune your ruleset will enhance performance and decrease false positives (or true positives that you simply don't care about). The major advantage of a HIDS, however, lies in its capability to detect specific changes to the files and operating system of its host. It can monitor file sizes and checksums to ensure that crucial system files are not maliciously modified without someone noticing. It can intercept rogue system calls that may be an attempt at exploiting a local vulnerability. Moreover, it can watch traffic within a system that never crosses the network, and therefore would never be seen by the NIDS.

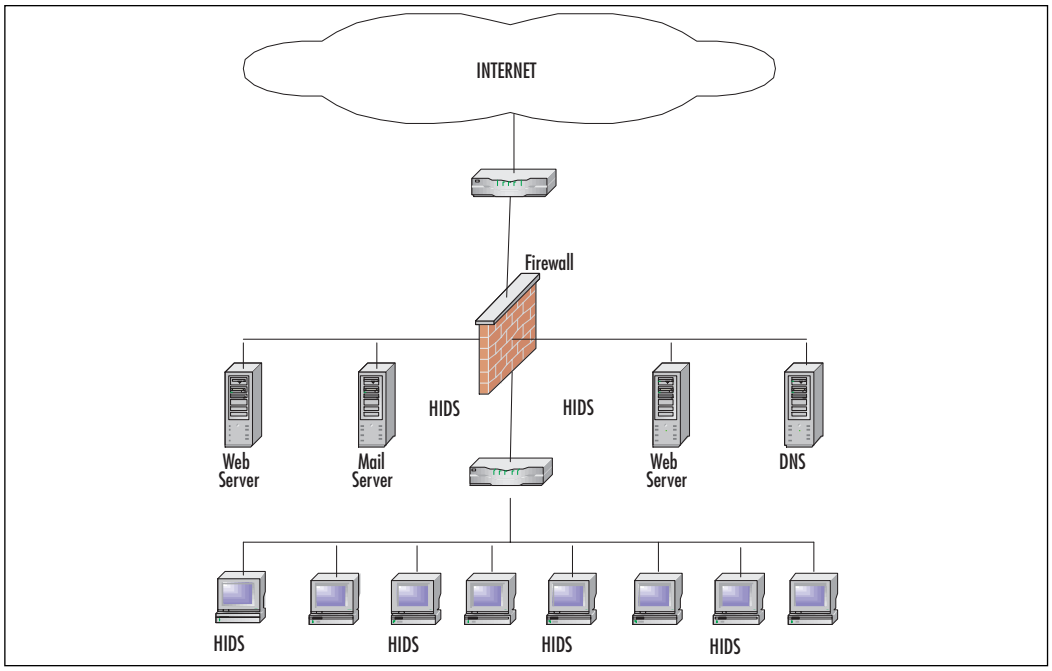
There are a few downsides to electing to use a HIDS. You'll have to choose one that is tailored to your operating system. If you have many different operating systems on your network and want to use the same vendor for all your HIDSs, you may have to do a little shopping to find the right vendor that supports all of your operating systems. A HIDS will add load to the host on which it is configured, as the HIDS process(es) will consume resources. This is usually not a problem on an individual's desktop, but can become one on a busy network server. Make sure you are familiar with the details of any HIDS that you choose and how it operates—some HIDSs will watch file accesses, usage times, process loads, and/or system calls, while others may also watch network activity from the point of view of that host. Know what features you want in your HIDS, and make sure that the HIDS you select will support those features on all the platforms you need.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 14

In addition, maintaining a large network of systems with many HIDS deployed can be very challenging. The HIDS solution alone does not always scale well, and without centralized management, you may be a very busy system administrator indeed trying to keep up with all those alerts.

Figure 1.2 depicts a network using a HIDS on specific servers and host computers. As previously mentioned, the ruleset for the HIDS on the mail server is customized to protect it from mail server exploits, while the Web server rules are tailored for Web exploits. During installation, individual host machines can be configured with a common set of rules. New rules can be loaded periodically to account for new vulnerabilities.

Figure 1.2 HIDS Network



Distributed IDS

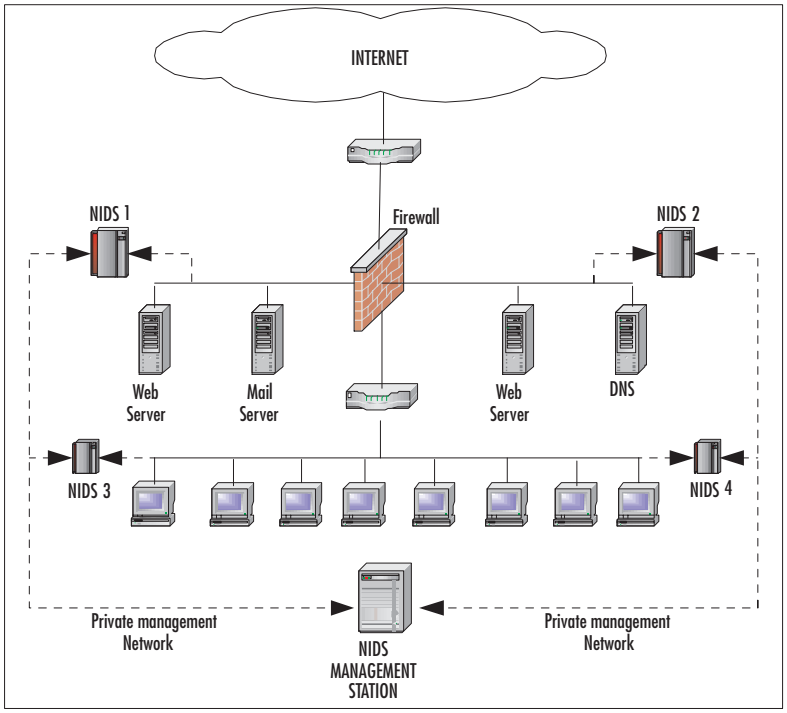
A Distributed Intrusion Detection System, or DIDS, is a combination of NIDS sensors, HIDS sensors, or both, distributed across your enterprise, and all reporting to a central correlation system. Attack logs are generated on the sensors and uploaded (either periodically or continuously) to the central server station where they can be stored in a central database. New attack signatures are created

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 15

or downloaded to the management station as they become available, and can then be downloaded to the sensors on an as-needed basis. The different kinds of sensors may or may not be managed by the same server, and the management servers are frequently separate from the servers that collect the logs. The rules for each sensor can be tailored to meet its individual needs, suiting the network or the host that each sensor monitors. Alerts can be forwarded to a messaging system located on the correlation system station and used to notify the IDS administrator.

In Figure 1.3, we see a DIDS system comprised of four sensors and a centralized management station. Sensors NIDS 1 and NIDS 2 are operating in stealth promiscuous mode and are protecting the public servers. Sensors NIDS 3 and NIDS 4 are protecting the host systems in the trusted computing base.

Figure 1.3 DIDS Network



The network transactions between sensor and manager can be on a private network, as depicted, or the network traffic can use the existing infrastructure. When using the existing network for management data, the additional security

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 16

afforded by encryption, or VPN technology, is highly recommended. Sending all the security information about your network across it in cleartext is just asking clever attackers to intercept those communications. At best, they can tell when they are triggering your IDS, and can tailor their behavior to avoid detection. At worst, they could intercept and change your alerting mechanism, hopelessly corrupting your data and any chance you might have of relying on it for analysis and/or prosecution. Another issue to keep in mind if you choose to have your DIDS communicate over your normal network is that if your company network is ever flooded or disabled by malicious traffic (as happened to many networks as a result of SQL Slammer), your IDS sensors won't be able to communicate with the correlation or management servers, which significantly reduces their usefulness.

OINK!

We'll refer to "stealth mode" for NIDS on occasion. This means that the NIDS is not visible to the network it is monitoring. This is generally done by not giving an IP address to the NIC that is being used for monitoring, and by using a device known as a "Tap" that only allows the receipt of traffic, not sending it. This method of watching network traffic is key to preventing attackers from knowing about your NIDS.

One of the main advantages of analyzing events using DIDs is to be able to observe system-wide, or even Internet-wide incidents from the 50,000-foot view. What might look like an isolated portscan to a class C subnet could look like a global worm propagating to a system like Dshield.

A friend of this book's editors, and frequent contributor to Dshield, is responsible for performing intrusion detection on two class Cs on opposite ends of a class B. He will watch a scan come through the lower class C, and return minutes later on the higher class C. DIDSs can be fairly complex to design, and require a talented hand to tune them and correlate and manage the data that is generated by all the sensors. The scope and functionality of the system varies greatly from implementation to implementation. The individual sensors can be NIDS, HIDS, or a combination thereof. The sensor can function in promiscuous mode or nonpromiscuous mode.

Now that we are familiar with how different types of IDSs can be deployed, let's look at the information they can gather.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 17

Application-Specific Information

All three types of IDSs are able to watch at least some application-specific information. This can vary from the traffic that goes to and from your Web server to the internal data structures of your custom-coded application. (Of course, for a custom application, you'd have to have custom IDS rules to match its traffic.) As application traffic goes over the wire across your network, the NIDS will be able to detect it. If it's sent in cleartext like Telnet or HyperText Transfer Protocol (HTTP) traffic, the NIDS should have no problem matching against it. For example, look at this signature, looking for access to a vulnerable PHP: Hypertext Preprocessor (PHP) application "Proxy2.de Advanced Poll 2.0.2."

Tools & Traps…

PHP and Shifting Acronyms

At its inception, PHP stood for "Personal Home Page." It was, according to the PHP history at www.cknow.com/ckinfo/acro_p/php_1.shtml, a wrapper for Web pages around Perl. Over time, as the functionality of PHP shifted into a full-blown server-side scripting language for Web servers, the acronym came to mean nothing, and then to the current recursive acronym "PHP: Hypertext Preprocessor," as described at www.php.net/manual/en/faq.general.php.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-PHP
Advanced Poll admin_tpl_misc_new.php access"; flow:to_server,established;
uricontent:"/admin_tpl_misc_new.php"; nocase; reference:bugtraq,8890;
classtype:web-application-activity; sid:2299; rev:2;)
```

Even if the traffic is sent in binary format, if there is a known payload or a consistent part of the packet (that is unique to avoid false positives) that the NIDS can match against, signature-based rule matching may be possible. Encrypted application traffic that is sent with sufficiently good cryptography, though, may be outside the scope of what most NIDSs are capable of detecting. Writing a good NIDS rule for traffic encrypted with a good random seed (for example, the same input string results in a different output every time it is encrypted) would be difficult. To learn more about rules and writing them for Snort, see Chapter 5 for an in-depth discussion.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 18

Encrypted traffic is where host-based IDSs shine. Application traffic that crosses the network in an encrypted format is usually decrypted at each endpoint. Consequently, traffic that was previously randomized gibberish becomes sensible patterns on the host, and can be matched against with signatures.

What types of things does one look for in application-specific information? Attempts to exploit input fields by entering too much data, known overflows or underflows exploiting lack of input validation, and attempted SQL injection are only a few possibilities. Of course, the signatures will vary greatly depending on the application that's being protected.

OINK!

Even though we just said that HIDSs shine when it comes to looking inside encrypted data because they are on the host that is sending or receiving the data (and as a result are more likely to see the information before it is encrypted or after it is decrypted), that isn't completely true. It is important to remember that they are only better if they are actually seeing the unencrypted data. That means that if the encryption is occurring at the application layer (for example, your Web browser or an SSH client or e-mail client is doing the encryption) and your HIDS is seeing the network traffic as it leaves or enters your system, after the encryption has taken place or before the decryption has taken place, then it doesn't matter that it is a HIDS; it still can't see through the encryption and is just as blind as any NIDS would be.

Host-Specific Information

While most HIDSs don't actually watch *everything* that happens on a host, they are capable of seeing all the behavior of a given host, from file creation and access to system calls to local network activity to the loopback interface. It is very common for HIDSs to create a database of the state of the system (file sizes, permissions, access times) when they are installed, and then monitor for deviations from that baseline. In fact, for many types of HIDSs the tuning process requires installing the HIDS software and then progressing with normal system activity to establish a baseline of what is changed when, and by whom.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 19

Subnet-Specific Information

Most networks have common patterns to their traffic flows. If you know that one machine on your network is a mail server, you will not be surprised to see Simple Mail Transfer Protocol (SMTP) traffic going to and coming from it. If you are used to seeing a network-monitoring device ping every device on your network every five minutes, that traffic is acceptable even though the same behavior from another device on your network would be worrisome. Over time, a good NIDS should be tuned to recognize the expected behavior of the subnet on which it resides, permitting traffic that is known to be expected and acceptable, and sending alerts for similar traffic from unauthorized hosts. The workstation of your authorized pen-tester may scan your network, while the workstation of your new intern may not.

OINK!

The NIDS deployment described in the previous paragraph is frequently referred to as a policy-based IDS. It is most effective in environments where you have strict control over what type of traffic is acceptable. As a result, it is very common in military deployments or for companies that exercise extremely tight control over their networks and systems. If you have a very dynamic or extremely complex environment, it may be harder to implement a strict policy-based IDS approach. We discuss this approach in more detail later in this chapter and in Chapter 12.

Another worthwhile and often overlooked component of the subnet traffic is the Layer 2 protocol mapping that can be done. Most IDSs overlook Address Resolution Protocol (ARP) traffic, used to map MAC addresses to IP addresses on the local network. It is possible for attackers to spoof traffic by changing their MAC address or forging an IP address that is not theirs and then trying to intercept the return traffic. This type of tomfoolery may be viewable at the subnet level, depending on your network topology. If your NIDS is not on the same local subnet on which the Layer 2 attacks are happening, it will not detect them correctly. When network traffic crosses a router, the MAC address changes. Since we're checking for the ports and locations of MAC addresses, we cannot afford to have them change before examination or the data becomes unreliable. Therefore, if you want to capture Layer 2 data with your NIDS, ensure that your

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 20

NIDS is on the same local subnet as all the machines you want to monitor, before any routers become involved in the data stream.

Distributed IDS

All of the information can be collected and correlated with a DIDS, but the scale is much greater. Instead of getting the local-network view of your subnet and its machines, you get a view of the activity across your entire enterprise. You can pick out data patterns that would have been baffling or inconsequential at a smaller scale, and what seems to have been an automated backup of one server turns out to be a coordinated (malicious) replication of data network-wide, when you look at the big picture. Looking at traffic from the DIDS level allows you to see large-scale data flows and overall trends more clearly. The downside is that you must have the tools to effectively comprehend the amount of data you are collecting; otherwise, the subtle attacks that you had hoped to discover will be lost in the general noise from your environment.

How the IDS Watches Your Network

Without an effective method of collecting data to analyze, there really isn't any purpose to an IDS. Luckily, there are several possible ways for your IDS to collect data to analyze. The following are the most common methods of collecting data for your IDS to analyze. Each has its own strengths and weaknesses, and all are best suited for different tasks. There are several possible sources of data for your IDS.

Packet Sniffing

Any IDS that looks at network traffic performs packet sniffing. As we mentioned, NIDSs operate by setting an interface into promiscuous mode and packet-sniffing on that interface. By doing so, they capture each packet that crosses the wire on the local subnet. They will not see packets that cross a TCP/IP stack internal to a machine, but they will potentially see everything on the local wire. However, many HIDSs that perform analysis of network traffic also use similar techniques without the use of promiscuous mode, to collect traffic specific to the host on which they reside. Packet sniffing is a classic way of doing intrusion detection, and there are equally classic techniques of IDS evasion that can be used against packet sniffing IDS; for example, fragmentation attacks, which split the attack payload among several packets. We discuss evasion techniques and provide some key references later in this chapter. We strongly encourage you to read them and then keep them in mind when listening to vendors talk about never

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 21

missing an attack. The IDS response to this was to create the capability for the IDS to reassemble packets and then match against the assembled packet. The attacker response was to change the way the packets are fragmented, causing some data to overwrite itself. Then, IDS techniques were created for that, and so on, and so on. In case you hadn't guessed, Snort uses packet sniffing.

Log Parsing

Another excellent source of security data is from system log files. Many IDS systems can pull data from the system logs and alert if they see anything anomalous. In fact, some of the original IDS implementations used log monitoring as their data collection method. Some attacks are blatant in the footprints they leave in your system logs; the Secure Shell CRC32 overflow, for example, can leave

```
sshd[3698]: fatal: Local: crc32 compensation attack: network attack
detected
```

in your logs.

OINK!

Dr. Tina Bird has done quite a bit of work in log analysis of intrusion attempts; you can read the results of her research at www.loganalysis.org.

System Call Monitoring

HIDSs are capable of setting themselves up as resident in the operating system's kernel, and watching (or in some cases intercepting) potentially malicious system calls. A system call is a request that a program makes of the operating system kernel. If the HIDS thinks that the system call might be malicious, such as requesting a change of one's user ID to that of the root user, it can create an alert or, in the case of some HIDSs such as the Linux Intrusion Detection System (LIDS), disallow the system call unless specifically overridden.

Filesystem Watching

Another very common tactic of HIDSs is to keep an eye on the sizes and attributes of crucial files in a filesystem. If your operating system kernel suddenly

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 22

changes size and none of your system administrators knows anything about it, this is probably something to check into. If you find yourself with world-writable directories or you find that your common system binaries have changed, it's possible that they have been Trojaned. Watching the filesystem like this helps alert administrators to possible malicious activity; if not before the fact, at least as soon after as possible. Tripwire is perhaps the best-known example of a tool to monitor files for changes, but there are many others that do the same thing, including the open-source tool Advanced Intrusion Detection Environment (AIDE).

How the IDS Takes the Data It Gathers and Finds Intrusion Attempts

Any IDS is going to collect a vast amount of data—networks are busy, servers are buzzing, there is data transfer constantly going on, processes constantly being run, and a general low hum of electronic noise on your network. To be effective, an IDS must have at least one (and possibly several) algorithm for determining what traffic is worth the attention of your administrators. There are several strategies, but at the most basic level there are two tactical options.

Known Good versus Known Bad

Network traffic can be identified and classified in several fashions. You can seek to have your traffic conform to a given security policy, dictated by the particular needs of your enterprise or your network. Some administrators choose to only allow traffic that they know to be good, while others choose to only block traffic that they know to be bad. Most often, policy-based approaches will center on a known-good approach. To make the best decision for your enterprise, consider what types of traffic you are likely to see, how much staffing you have to deal with the alerts, and how paranoid you want to be.

Do you want to identify the known acceptable traffic on your network, and flag on everything else, or do you want to identify the known attacks and let everything else go by without comment? That's the basic conundrum of IDS strategy; firewall administrators are no doubt familiar with the dilemma. The known-good strategy will be orders of magnitude more work, as you try to sort through all the traffic on your network, determining what is supposed to be happening and what is dodgy. You'll immediately be faced with a large amount of false positives spewed forth by a frantically busy IDS, and will have to slowly winnow them down to a manageable level as you identify the known-good traffic on your network. In addition, unless nothing ever changes on your network, you will have

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 23

to constantly tune and retune the IDS to adjust to the normal changes that happen over time in almost any environment. There are automated tools for defining "normal," where "normal" is expected to be an acceptable approximation of "good." However, such tools suffer from issues of false positives in complex or highly dynamic environments. They can also be tricked into deciding that something is "normal" if the new activity occurs in small enough amounts over a long enough period of time. (Think of the story of boiling a frog—if you drop a frog in boiling water it jumps out. If you put a frog in cool water and slowly raise the temperature, it won't notice and will simply be cooked.)

However, following a strategy of only alerting on known or suspected malicious traffic will result in much lower alert volume. In addition, because the rules can be very specific about what the definition is of something bad, when an alert does go off (assuming the rules are well written), you can be fairly confident that the "bad" activity was actually seen. This means that the person monitoring the IDS doesn't have to be as skilled (because he doesn't have to be able to troubleshoot the IDS), which can be a significant issue. However, this approach carries the strong likelihood of missing attack traffic that doesn't happen to match your rules or algorithms, and if you write more flexible rules, the number of false positives will go up. In some scenarios, such as with Archibald Enduser's home box, where Archibald doesn't know a lot about intrusion detection and doesn't have the time or inclination to learn, this may be the better solution. However, if you want to increase your likelihood of catching a given attack, and you have the resources available to monitor and maintain the IDS, you might want to consider the other approach. Your choice of strategy is a cost/benefit analysis; weigh the time and resources that you are willing to devote to IDSs with the importance of catching the maximum number of attacks.

OINK!

In reality, most well-planned IDS implementations use a combination of both approaches. Where you can tightly define allowed traffic, use a "known-good" approach. Where you have to be a little more permissive or the environment changes too frequently to define, use "known-bad." Use each where it makes sense and you'll be a much happier intrusion analyst.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 24

Technologies for Implementing Your Strategy

IDSs differentiate attack traffic from innocuous network and system activity in several ways. Some primarily use a technique called *rule-based (a.k.a signature-based) analysis,* matching a known pattern to activity seen on the system or network. We have seen examples of Snort rules already, looking for packet content on the network and matching it to a series of predefined rules. The same thing can be done when looking at entries in log files or sets of system calls. This is very similar to the way many antivirus programs use virus signatures to recognize and block infected files, programs, or active Web content from entering a computer system (and why you have to constantly update your anti-virus software). Signature detection is the most widely used approach in commercial IDS technology today, since it is easily demonstrable, effective, and very customizable with limited training or experience. As new attacks are developed and seen in the wild, new signatures can be written to match and alert against the new forms of attack.

A more complex version of rule-based analysis is protocol analysis. Instead of writing a relatively simple rule that defines something about a specific event (good or bad), protocol analysis attempts to define every possible acceptable behavior for a specific kind of activity. For example, when our computer wants to set up a TCP connection, it sends a SYN packet. The acceptable responses are either RST/ACK or SYN/ACK. Anything else would be a violation of the protocol. This approach allows a little more flexibility in defining what "bad" is. Instead of saying, "If you see a string of greater than 500 bytes, filled with a specific character, it is an attack of this type," you can say, "At this point in the connection, you should not see strings greater than 500 bytes. If you do, it is an attack. If you see more than 500 bytes at some other point in the connection, it is okay." The problem is that while protocols are tightly and clearly defined, not all vendors choose to pay attention to everything in the protocol definition. As a result, you may find that your protocol analysis-based IDS is correctly complaining about something that is not allowed in the RFC (Request For Comments—the documents used to define most Internet protocols. For a full list, see www.rfc-editor.org) but is completely normal for applications from a specific vendor. In addition, it is tremendously time consuming and complex to write a good protocol model, and to implement it in an efficient enough fashion that it can be used to watch high-speed networks. This takes years of experience. This means that most vendors tend to be *very* unwilling to share their protocol models openly, even with customers. Consequently, troubleshooting false positives for protocol analysis IDS, or getting a false positive fixed can be a long process while you wait for your vendor. Another approach is called *anomaly*

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 25

detection. It uses learned or predefined concepts about "normal" and "abnormal" system activity (called *heuristics*) to distinguish anomalies from normal system behavior and to monitor, report on, or block anomalies as they occur. Some anomaly detection IDSs come with predefined standards for what normal network traffic should look like, and others watch the traffic on your network (or activities on your systems) and use a learning algorithm to develop a baseline profile from that. These profiles are baselines of normal activity and can be constructed using statistical sampling, a rule-based approach, or neural networks, to name just a few of the methods.

Literally hundreds of vendors offer various forms of commercial IDS implementations. Because of the simplicity of implementation, the majority of implementations are primarily signature based, with fewer protocol analysis solutions and only limited anomaly-based detection capabilities present in certain specific products or solutions.

OINK!

While most effective IDS deployments combine network- and host-based IDS implementations, very few vendors have been able to successfully offer both kinds of IDSs or IDSs that combine multiple technological approaches The products end up doing everything in a barely acceptable fashion but nothing tremendously well. This may actually be changing due to the large number of acquisitions that we've seen in the IDS space in recent years. The vendors who are left may actually have the resources to dedicate to each separate area of focus, or they may just manage to do a miserable job in all the areas—which is what we've seen so often after acquisitions in the past.

What the IDS Does When It Finds an Attack Attempt

Most modern IDSs include some limited automatic response capabilities, but these usually concentrate on automated traffic filtering, blocking, or disconnects as a last resort. Although some systems claim to be able to launch counterstrikes against attacks, best practices indicate that automated identification and back-trace facilities are the most useful aspects (and the ones least likely to get you sued) that such facilities provide and are therefore those most likely to be used. There are different and highly configurable approaches to what the IDS actually

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 26

does when it detects an intrusion attempt. Although Chapter 12 will get into this in more detail, it is worth discussing briefly the merits of active IDS response (sometimes mistakenly known as IPS, or Intrusion Prevention Systems) versus the more traditional passive detection and alerting.

Passive Response

Traditionally, IDSs will watch the activity, and can be configured to log to a file and/or send alerts to the administrator(s). These alerts can take many forms—Simple Network Management Protocol (SNMP) traps, outgoing e-mails, pages or text messages to the system administrator, even automated phone calls. Most administrators configure the IDS to alert them in various ways depending on the severity of the perceived attack and the frequency of its occurrence. You don't want to be paged 10 times an hour for something that seems dire at first but turns out to be a false positive every time. However, you do want to be notified for an alert indicating a serious compromise, especially if it doesn't false-positive very often.

Traditional IDSs stop there. They are usually set up with a management interface entirely separate from their listening tap on the network, so that they don't betray their presence on the tap by sending alerts all the time. Very often, the listening tap doesn't even have an IP address, and is a stealth interface configured not to respond to any traffic.

Active Response

IDSs with Active response capabilities and IPSs (the two are different, see Chapter 12 for an explanation of why) emulate all the behavior of traditional passive IDSs as far as detection goes. However, when they see an attempted attack, they can be configured to take proactive measures against it rather than just alerting the administrator and waiting for him to take action. They can be placed inline and drop traffic they see as malicious, they can spoof Transmission Control Protocol (TCP) resets to either the source or destination systems (or both) to abruptly terminate a TCP session that they see attack traffic coming through, or they can send Internet Control Message Protocol (ICMP) Unreachable messages to the source system in an effort to convince it that the target system is unreachable; some reconfigure firewalls or routers between the targets and the attackers to block the traffic. Some systems will do nameserver lookups or traceroutes on the attacking system in an attempt to gather informa-

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 27

tion about it. Some will even portscan the attacking system back, and give you a report of its likely operating system and possible vulnerabilities.

The appeal of active response is that you don't have to have a system administrator watching the wire in real time. The peril is that the consequences of a misconfiguration become much graver. We have set up brand new IDSs with prevention capabilities before, only to watch them listen to the network traffic, decide that our DNS server was portscanning the network, and block all access to it. Without name service, many network applications come to a screeching halt. IPSs should be checked for whitelisting capabilities beforehand in order to avoid just such scenarios. It would also be advisable to check the legalities in your jurisdiction if you're planning to have your system automatically trace or scan "attacking" systems.

Inline IDS

Another common configuration debate is whether your IDS should sit on a tap on your switched network, or sit inline between you and the Internet. There are advantages and disadvantages to both configurations. If you intend to have your IDS act as an IPS, setting it inline might be something you would strongly want to consider. Prevention is far more effective when the IDS is capable of simply dropping traffic that it has determined should not be allowed through. When your IDS is not inline, you can send ICMP unreachables or TCP Resets to both source and destination, but you have to hope that the devices themselves behave properly. You're not controlling the network segment between them, so there is only so much you can do. With an inline IDS, far more control is in your hands. Chapter 12 discusses this issue in greater detail.

There are two prime worries with this type of configuration—false positives have even more disastrous consequences than with your average IPS, and performance can be a significant concern. Since all of your network traffic is going through this one box, a single point of failure is often worrisome from a redundancy and performance point of view.

Answering Common IDS Questions

Let's look at some of the major questions that people often have when considering an IDS for their network. It's important to understand the function of an IDS within your overall security design, the differences between an IDS and your other security devices, and what an IDS can and cannot do for you in terms of enhancing the security of your network.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 28

Why Are Intrusion Detection Systems Important?

IDSs provide an integral audit component of a robust security design and policy. They let you know when you're being scanned and when you're being attacked. They provide more information than you could get just by checking your server and firewall logs. You can see the attacks that fail and the attacks that succeed, and get real-time notification of attempted attacks. You can watch your own network traffic and become aware of misconfigurations as well as malicious attacks earlier than you may have noticed without an IDS. They are not the be-all, end-all solution to every security woe, but they are a valuable tool in the hands of a skilled security administrator.

Why Doesn't My Firewall Serve as an IDS?

While some integrated appliances out there claim to be both a firewall and an IDS, and we are probably going to see more of those in the future, a firewall's function is to filter packets, not to alert on potentially malicious traffic. Firewalls are primarily designed to deny or allow traffic to access the network, not to alert administrators of malevolent activity. Many firewalls are only network-level packet filters, allowing or denying traffic based purely on the source and destination IP address and port. This doesn't begin to touch the complexity of the traffic analysis that an IDS handles. We discuss this in depth in Chapter 12, but the simple analogy is that you don't trust the locks on your doors to also act as cameras, so why should your locks on your network (the firewalls) be expected to be cameras (the IDS)?

Why Are Attackers Interested in Me?

Put simply, because you're there. While attackers certainly do look for high-value targets (targets that have something they specifically want), any system connected to the Internet these days is a potential target. While many attackers will go for juicy-looking targets and other low-hanging fruit, not being the most tempting target out there doesn't mean you are safe. You don't want to be just a little bit more secure than the next guy… in today's digital environment, you want to be actually safe. Many managers make the mistake of thinking that the attacker wants the company's data. In most cases, the attacker wants to steal bandwidth, not secrets.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 29

Automated Scanning/ Attacking Doesn't Care Who You Are

Many attackers scan (or even attack without scanning) entire class B subnets at a time. For those of you who don't do exponential math in your head, that's 65,536 machines at a time. Many script kiddies aren't looking for any particular machine; they just want as many compromised "zombie" machines as possible. Therefore, they will launch their automated scans, and attempt to exploit all machines that they see as vulnerable, regardless of who they are. You@example.com is treated just the same as you@whitehouse.gov or you@google.com. And that's more consideration than you'll get from many of the automated worms and viruses, which will happily scan random subnets and all the machines on them without any cognizance whatsoever of what machines are on those networks and whether they should be doing that after all.

So why do these attackers want so many random machines that may or may not be valuable to them? They want something you have, whether that's bandwidth, clock cycles of your CPU, or data.

Desirable Resources Make You a Target

The more you have, the more others will want it. If even Archibald Enduser is a target, larger machines and corporate networks are that much more so. But what are these miscreants hoping to do with your computer?

Bandwidth

Well-connected computers are valued in the underground for several purposes. One of the most popular is to launch distributed denial-of-service attacks (DDoS), using your bandwidth to send attack traffic to people whom they don't like. Of course, this will make your legitimate use of your computer and its network a lot slower, but they don't really care about that. Bandwidth can also be used for for-profit spambots, hijacking your computer to churn out ads for generic Viagra and plastic surgery, or for hosting high-volume warez servers of pirated software, movies, porn, and music.

Disk Space

Disk space is usually a concern for attackers planning on setting up warez servers to share out pirated software, movies, porn, and music. The more disk space you have, the more attractive your server will be to use for such purposes.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 30

Valuable Information

If your machine has any type of sensitive information on it, it is possible that the attackers are after that. Whether it's a targeted attack to attempt to steal your secret corporate plans to build Isengard 2.0, or some attacker who got lucky, corporate espionage or information selling is not unfeasible. Look at the scandal involving partisan information theft in the U.S. Congress in 2004, for just one example.

OINK!

Because there is a profit to be made from stealing information, these attackers are frequently the best funded and most highly skilled of the threats you or any company you work for are likely to face. Case in point: Six months prior to Slammer, there was another worm that exploited a weakness in Microsoft's SQL server. The worm, known as SQL Snake, took advantage of the fact that many SQL server installations had a default SA (admin) password that was blank. The person who released the worm is said to have stolen hundreds of databases, and was offering them for sale.

Political or Emotional Motivations

Some attackers are motivated by political gain, or some sort of a feeling of revenge upon someone they don't like. The DDoS attacks generated by the MyDoom worm variants in early 2004 are an example of this, targeting sco.com and Microsoft.com, and reportedly passing over domains like google.com and Berkeley.edu. Internet Relay Chat (IRC) servers are well known in the security community for drawing fire—when Internet flamewars break out, DDoS attacks are often the result. There's a well-known ongoing series of cyber hostilities between Indian and Pakistani hackers, for example, with viruses flying back and forth and defacements proclaiming political causes and the superiority or inferiority of one nationality over the other. Since the September 11 terrorist attacks on the United States, there have been reported acts of technical jihad, with American hackers attacking sites they perceive as affiliated with al Qaeda, and vice versa.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 31

Where Does an IDS Fit with the Rest of My Security Plan?

Alongside a good security policy, incident response plan, firewall architecture, virus checkers, and all the other features of a modern security plan for enterprise networks, an IDS can play a vital role in securing your enterprise. Your IDS can be an early warning of network trouble, often picking up malicious activity before any of your other layers of defense. Your IDS can provide necessary logs and proof of activity, should you ever need to go to court regarding a network intrusion. Your IDS can alert your system administrators and security staff to problems in time for them to take effective action, and it can be a useful tool in enforcing enterprise IT policy and flagging violations. Last but certainly not least, it can provide a warning that your other security measures may have failed in time to fix them. Many companies and organizations put a NIDS sensor on each side of their firewall and then tune the sensor on the protected side to send high-priority alerts if any traffic is seen that should not have gotten through the firewall.

Where Should I Be Looking for Intrusions?

A good security policy addresses multiple layers of security, protecting your enterprise assets in many ways. This philosophy is called "defense in depth," and is central to mounting an effective defense against the multiple threats facing a modern enterprise. If attackers can't get past your firewall, they may call the help desk and try to bluff them into giving away account credentials. If they can't get in to your headquarters by walking on in, they may send your vice president an e-mail with a backdoor disguised as a holiday card. The creative ways in which attackers can approach your network are limited only by their imaginations. Unfortunately, this means that the most correct answer to this question is, "you should be looking everywhere." However, when talking strictly about IDS placement, you should be watching every point where your network connects to another network (Internet connections, DMZs, modem banks, VPN gateways, and so forth), and any server that is important enough that you would be upset if it were compromised. If you would like to know more about some of the alternate ways that attackers use to get into companies, Kevin Mitnick's recent book *The Art of Deception* describes some of the various nontraditional ways that security can be subverted.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 32

Operating System Security—Backdoors and Trojans

This is the classic sort of thing that most people think of when they consider network security—Trojans, backdoors, compromises of individual boxes through weaknesses of software or configurations. In addition to good system administration practices like keeping up to date on your patching and turning off services that you don't need by default, you should consider a regular scan or vulnerability assessment of your own network. This will help you detect unknown listening services or unapproved configurations. You should have standard, documented, hardened configuration templates so that when a new machine is attached to your network, it's not going to be the gateway through which a thousand preventable compromises pour. IDSs can help greatly in watching for this type of traffic.

OINK!

There has been an interesting development from a couple of vendors (well… two so far) who are now offering software that supposedly can identify vulnerabilities on systems just by passively watching their network traffic. If it works, this would allow you to have your IDS sensor actually perform some amount of vulnerability monitoring and analysis. One of the biggest complaints many companies have with vulnerability scanning is the risk of having it crash a server or the added load on the network. This approach has the advantage of not ever touching the servers and not adding any load to the network at all. At publication time, the two vendors we know of who offer this are Tenable Security and Sourcefire.

Physical Security

Good security practices look at more than just your network connectivity. Physical attacks and approaches are alive and well. Can someone walk in to your enterprise, pick up a laptop with valuable data on it, and stroll out the door undetected? Don't laugh! This happens more often than you might imagine. It happened recently to an airline; two men dressed as technicians went in to an office and walked out with two of the company's mainframe computers. We can only speculate as to what they wanted or have done with the information they

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 33

got, since they haven't been caught. It is highly doubtful that they were just doing it for the thrill. If so, you need to give some thought to your physical security model as well as your network security. Are your servers located in a separate space with some type of access control for your staff? Any network security consultant will tell you that physical access to a device is extremely dangerous. In most cases, all you have to do is reboot the machine and set the BIOS to boot from a CD-ROM. There are security toolkits small enough to fit on a credit card-sized CD-ROM that contain all the forensics tools you'd need to discover almost any type of information about the servers' hard drives and data, and plenty that will change things at will. These toolkits are operating-system agnostic; a bootable Linux CD can reset your Administrator password for a Windows machine, for example. Even more dangerous, bootable USB drives are becoming common now, which counters the remove-all-disk-and-CDROM-drives defense.

Tools & Traps…

Bootable CD Toolkits

- **FIRE** A portable CD-ROM based Linux distribution with 196 security and forensics tools at the time of writing (version 0.4). FIRE is designed to provide an environment to do vulnerability assessment, data forensics, virus scanning, and incident response from a bootable CD-ROM. Tremendously useful to the security administrator, FIRE is also extremely useful to people of variable morality in physical vulnerability assessment scenarios. Anything you can do with this tool, an attacker can also do. Available online at http://fire.dmzs.com/.
- **Knoppix** A full-featured Linux environment including graphical user interface (GUI), OpenOffice, the Gimp, Abiword, and Mozilla. Less obviously useful to the attacker or the security administrator than FIRE, but offers the capability to look at office documents on the local machine right there from your own operating system, edit, and leave without having had to log in or access the system through legitimate means. Available online at www.knoppix.net.

Continued

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 34

- **Linux-BBC** Well known in the Linux community, the Linux Bootable Business Card (BBC) is a Linux distribution on CD-ROM cut to the form factor of a mini-business card. Small enough to slip into anyone's wallet unnoticed, the Linux-BBC supports large IDE disks, BitTorrent, and The Coroner's Toolkit, a software forensics package. Available online at www.lnx-bbc.org.
- **Offline NT Password & Registry Editor, Bootdisk/CD** Need to change the Administrator password (or any other password) on a Windows system? Don't have a login currently? Go to http://home.eunet.no/~pnordah/ntpasswd/bootdisk.html and download this toolkit. In less than 10 minutes, you can change the password, boot back to Windows, and log in with your new password.

Keeping your servers away from miscreants and attackers isn't the limit of physical security, though. Guarding against someone running off with a laptop containing sensitive data, ensuring that if someone sets fire to your main data center that you have an offsite backup of all your important information, and training your staff to be aware of social engineering attempts and what to do in case of an attempted security breach are all important facets of physical security.

Application Security and Data Integrity

Are you sure that your data has not been tampered with? How do you know that the source code in your central CVS repository is the same as the source code that was there last night? How can you prove that the figures in your banking database are true and accurate rather than jimmied? Provable authentication of the integrity of your data is crucial to the modern enterprise, and there are highly motivated attackers out there just waiting to get their hands on your resources. From the attempted backdooring of the Linux source code tree in November 2003 to the wireless hack of an Israeli post office's network, leading to the alleged theft of 80,000 credit card numbers, we can see that attackers have every reason to want to take advantage of vulnerable applications. If you don't have some way of verifying that your data is unmodified or that your transactions are secure, you will be in very bad shape indeed in the event of a successful intrusion, or even a potentially successful one. Saying "I don't know" when asked about data integrity is rarely good enough for the customers.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 35

Correlation of All These Sources

Although Chapter 10, "Optimizing Snort," addresses this issue in depth, it is worth mentioning that correlating your security information from multiple sources is much more likely to help you reconstruct what happened when analyzing intrusion attempts. Data from your firewalls and routers can back up the alerts seen by your IDS. Overlapping sources can cover for each other in case of the failure of one system, and when you can correlate alerts from multiple sources, you can have a much higher confidence that you aren't dealing with a false positive. Logs of keycard swipes can help you determine who (or at the least, whose access credentials) was in a given area at the time in question, network access credentials can help you determine who logged in, and security cameras can help you verify whether the person at the keyboard was the person whose password you have on file.

What Will an IDS Do for Me?

An IDS can be a valuable addition to your security toolkit. It can give you unprecedented insight into what's really going on in your network, and alert you to new trouble or attacks before you otherwise would have seen them. It can help you monitor and enforce your company's security policies, gain deeper insight into trends in your system and network usage, and plan better for future budgeting and purchases through seeing where your blind spots and problems are. It can notify your administrators of a likely system compromise, or even of a failed attempt. And it never gets tired, never needs a coffee break, and doesn't demand a raise every time you yell at it.

Continuously Watch Packets on Your Network and Understand Them

We have yet to meet the system administrator or security engineer who is capable of this for more than five minutes, and that's on a slow network connection and generally reading hex, not binary. An IDS is perfectly capable of tirelessly matching packet after packet to its known signatures, and comparing their payloads without needing to translate into a human-readable form. Its algorithms are normally at least several orders of magnitude faster than a human attempting to perform the same job, and generally less prone to mistakes.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 36

Read Hundreds of Megs of Logs Daily and Look for Specific Issues

An IDS can significantly speed up the amount of log files that you can parse on a daily basis. When you are responsible for the security of a large environment, the volume of log files that you'll find yourself accumulating is truly astounding (think terabytes for a large group of systems and an active high-speed network). Going over them all by hand becomes increasingly impossible the bigger your network grows. A log-parsing IDS provides a sane and sensible way to look for particular issues and signatures in your log files, giving you a better idea of what's going on with all your various devices.

Create Tremendous Amounts of Data No Matter How Well You Tune It

Even the most precisely tuned IDS is going to have voluminous output. Although it seems almost a contradiction to say so, anomalous network and system events are happening all the time. Users are becoming root. Commands are being sent over Web interfaces. Administrator passwords are being changed, packets with bad combinations of TCP flags are being sent, applications are abusing protocols in ways that only the most twisted and tortured of minds could come up with, and automated worms and viruses continue in their blind quest for self-propagation. Each of these events can trigger an IDS alert. And when you have a few thousand of them a day, well, managing your alerts becomes a major challenge.

Very often, IDS administrators are faced with the daily prospect of having to sort through a few thousand (or a few hundred thousand) alerts, many of which are known issues, but not tuned out because someone eventually intends to get around to correcting them. Some are just difficult to tune out by their very nature—many operating systems and applications send packets that just should not be! However, you can't spend your time tuning out every individual system on the Internet that might be running one of those operating systems, and you don't want to junk the signature entirely for fear of missing the actual stealthy portscans that might be network reconnaissance. When you decide to set up an IDS, be prepared for some situations akin to this to occur. No matter how well you tune, you will get data—and lots of it. Some of it will be false positives. Writing good rules and correlating your data can decrease the false positives and even the number of true positives that need to be looked at individually, but you still end up with lots of data.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 37

Create So Much Data that If You Don't Tune It, You Might as Well Not Have It

One of our special frustrations as security geeks is encountering situations where a company has invested a fortune in the latest cutting-edge IDSs, sparing no expense, and then has hired one person with no security background whatsoever to monitor and administer them all. The poor administrator has no idea how to tune an IDS, and still less idea of how to deal with the barrage of alerts she's being hammered with. The pointy-haired boss's inevitable conclusion to this scenario is that all IDSs are worthless. After all, they paid for the best, didn't they?

Tuning the false positives out of your IDS is crucial. Having knowledgeable administrators involved in the design and placement of the sensors and then in the tuning of the ruleset is essential. If you don't know your network well enough to winnow out the known issues and the definite false positives, you'll be awash in a sea of portscans and informational alerts, with no easy way of wading through all that data to find the relatively few blatant attacks and/or subtle system compromises. Every IDS out of the box will generate massive amounts of false positives, and an unknowledgeable security geek might as well not have one.

Find Subtle Trends in Large Amounts of Data that Might Not Otherwise Be Noticed

One of the benefits of having such a massive base of data is the ability to look at trends in the alerts or packet flows. Are you getting more scans for an unusual port today than you were yesterday? Has it been steadily on the rise recently? Perhaps a new tool or exploit out there targets that port. Have you been seeing more failed logins to various servers on your network? Perhaps someone is walking around and trying to guess passwords. The ability to see the big picture in the reams of data may be enhanced by an IDS, particularly an IDS with correlation capabilities.

Supplement Your Other Protection Mechanisms

An IDS can act as confirmation or backup for your other network security systems. This goes back to the principle of defense in depth. If you are seeing exploit traffic aimed at your Web proxy and you're not sure if your proxy sanitizes the traffic before passing it on to your end user, check your IDS. See if it's alerting on the traffic both before and after the proxy. If you know that someone with Administrator access used Remote Desktop to connect to the Exchange

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 38

server right before it broke yesterday, check your IDS logs to see if you have a record of who accessed that server, from where, and (if you have both HIDSs and NIDSs) what sort of traffic he sent. The absence of an IDS alert should not be used as proof positive that everything is okay. As we said earlier, IDSs will not catch every attack. Even if they have a signature for it, a sufficiently high volume of traffic will cause the IDS to drop packets. However, the presence of an alert can be used as a backup and support to other network security systems and logs.

Act as a Force Multiplier Competent System/Network Administrator

Using an IDS, good security geeks will be able to go through far more logs and far more network data than they could without one. While an IDS will not replace additional skilled help, it can make each competent geek more effective than he would have been without the additional tools. When investigating an intrusion attempt, it is greatly helpful to be able to say, "What other alerts did this source IP or user generate? What other alerts were associated with this destination IP?" Being able to quickly put your fingers on other relevant data can help administrators understand the kind and scope of their issue far more quickly than if they had to do all the log parsing and searching by hand.

OINK!

What Is a Force Multiplier? A force multiplier is something that increases the amount of result you get back for the force exerted. Look at any book on mechanical engineering (*The Way Things Work* is a good one) for examples.

Let You Know When It Looks Like You Are Under Attack

With the myriad alerting capabilities of most IDSs out there, there are a plethora of ways to notify your on-call or on-duty system administrators when it appears that an attack is ongoing. This time saved can be an invaluable asset to an incident response team. It can make the difference between pulling one compromised system off the network before it has a chance to branch out and launch

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 39

attacks at others, or dealing with a massive enterprise-wide security breach that will take endless hours of labor to address.

What Won't an IDS Do for Me?

An IDS is not the be-all and end-all solution to all your security woes. It will not replace your system administrator, make that guy on IRC who doesn't like you go away, or answer that e-mail that you've been avoiding. It will not secure the physical perimeter of your site, magically detect every possible malicious bit flipped on your network, or tell you when one of your employees is thinking about selling you out to the competition. To get the most out of an IDS, it is important to understand its capabilities and limitations, and to design your security policy accordingly.

Replace the Need for Someone Who Is Knowledgeable about Security

Even the best IDS is only as good as its programming. It will do what you tell it to do faithfully, it will alert as you tell it to alert and, if an IPS, will respond as you tell it to respond. However, it can't tell you what to do in a new and unprecedented situation. It can't write its own signatures for new attacks, and it can't deal with an intelligent, flexible, adaptive attacker who takes an approach outside of its specifications. It cannot determine what your security policy should be. It cannot make informed recommendations for your network based on the latest industry developments. In short, it cannot replace a skilled security geek.

Catch Every Attack that Occurs

New attacks are being developed all the time. Even as we write this, even as you read this, attackers are out there trying to figure out new ways to break into systems. Sometimes these are new ways to exploit old vulnerabilities, but other times they are totally new approaches. Your IDS is not configured to handle all possible attacks, simply because some of them haven't been invented yet. You can only protect against the type of attacks of which you are aware. And even some of the attacks that are known are not guarded against by all IDSs. Your IDS will help you see the attacks and potential attacks that are out there, but it won't catch everything.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 40

Damage & Defense…

fragroute and the Newsham/Ptacek Paper

In 1998, Tim Newsham and Tom Ptacek wrote a paper entitled, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," describing ways to evade detection by most of the IDSs then available (www.insecure.org/stf/secnet_ids/secnet_ids.pdf). The techniques in question included testing the timeouts on IDSs, checking reassembly of fragmented packets (overwriting the same data with different content), simulating delays and packet loss in network programs, and randomization of IP parameters to evade operating system fingerprinting. Although this made many people in the intrusion detection community sit up and pay attention, it was nothing compared to the stir when Dug Song released first fragrouter and later fragroute, tools that implemented most of these attacks (www.monkey.org/~dugsong/fragroute/). The theory was now reality. Many of these attacks are addressed and now detected by Snort since Snort version 1.9, but there are still many IDSs that may miss them, and some of the attacks are simply hard to address from a network perspective. One approach currently getting a lot of attention is target-based IDSs, which combine a knowledge of your network, operating systems, and configuration with live detection of attacks. The aim of target-based IDSs is to present the administrator with alerts with a tighter focus, drastically cutting the number of false positives and centering analysis on the most likely real alerts. You can read more about target-based IDSs in *Information Security Magazine* at http://infosecuritymag.techtarget.com/ss/0,295796,sid6_iss306_art540,00.html—target-based IDS reviews were featured in their cover story in January 2004.

Prevent Attacks from Occurring

No IDS out there is going to magically make attackers stop attacking you. Your defenses may prevent these attacks from succeeding, but the attackers will keep trying to break down your digital walls. No matter how good your IDS is, it will not change human nature or the inclination of malicious attackers to try to own your network.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 41

When you are choosing and installing a NIDS, it is instructive to consider what you will not see as well as what you will see. If traffic is encrypted, you will still be able to see the IP headers and transport layer protocol headers, but you will not be able to decode the contents of the packet without breaking that encryption. You can watch how much traffic is sent, and from whom to whom, and how often, but you won't be able to see what they're saying. Depending on the type of NIDS you have deployed, this may or may not put a cramp in your style. Signature-based IDSs that depend on traffic being sent in cleartext may not alert if the traffic is encrypted. Protocol analysis may still work for encrypted traffic, but may break if the traffic is sent on an unexpected port. Traffic pattern analysis is likely to be your best bet when dealing with encryption.

OINK!

It should be obvious that your NIDS won't be able to see inside your network traffic if it is encrypted (unless you use special tools and change how you do encryption). What might not be quite as obvious is that even most HIDSs that look at network traffic (a.k.a Network Node IDS or NNIDS) won't be able to see inside encrypted traffic either. The reason for this is simply that almost all HIDSs watch network traffic as it is coming in to or going out of your system, somewhere around Layer 2 on the network stack (just before the traffic goes to the hardware from the OS). Currently, the majority of encryption is being done at the application layer (Layer 7) by applications such as your Web browser or SSH. This means that the traffic is still encrypted when the IDS sees it entering or leaving the system. This is unfortunately something that most vendors forget to mention when talking about the benefits of their products.

This is becoming more and more of a problem, as more and more environments begin using encryption in more and more of their network communications. Fortunately, IDS vendors are aware of this and are working on solutions. We hope they'll be good ones.

Prevent Attacks from Succeeding Automatically (in Most Cases)

With the exception of some IPSs, in most cases, by the time the IDS has seen the attack attempt cross the wire, it has either succeeded or it has not. In the case

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 42

of an e-mail with a viral payload, for example, it's possible that the IPS would trigger on the subject line and have time to send a reset-kill and end the mail transfer before the entire message, complete with virus, could be delivered. However, in many other cases, attack and success of the exploit follow hard on each other's heels, and there just simply isn't enough time for the IDS or IPS to jump in there between the last no-operation command and the execution of the shell code.

Replace Your Other Protection Mechanisms

While there are many all-in-one security products out there, don't be fooled into thinking that any one security product can do the job of a different type of security product. Just because you have an IDS doesn't mean that you can junk your firewall. The presence of a VPN does not mean that you don't need to patch your systems, either. The process of securing your network is aided by redundancy and layers of reinforced security. An IDS will not by itself be the only security device you'll ever need or want.

What Else Can Be Done with Intrusion Detection?

These are only some of the possible uses for an IDS. Many HIDSs allow you to audit and monitor use of shared resources. They provide enhanced capabilities of determining who is using shared network resources, provide benchmarking and resource utilization statistics for monitoring server functions, and can match subject lines or content of e-mail to be able to alert on and/or get rid of mails with known malware content. The possibilities are endless, and as flexible as your ruleset and IDS implementation.

Fitting Snort into Your Security Architecture

Since you're holding this book, we assume that you have or are interested in having Snort in your network. Snort is a very flexible network IDS, offering a multitude of rules already authored as well as the ability to write your own. There are several mailing lists where people trade new Snort rules that they've written in response to the latest attacks, and offer commentary on the rules and the new incidents they see on their networks. Snort is very full-featured, with

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 43

many preprocessors to parse different types of data, a bevy of keywords to allow matching of the content, port, protocol, and more, portscan detection, buffer length detection, and many other features—and since it's open source, you can add any functionality you like. There are also many other add-ons to support logging alerts in database formats, management and automated downloads of new rules, distribution of rules to sensors without clobbering the local rulesets, a Web interface for Snort sensor management, and others. Although all these features are explored at much greater length in later chapters, let's take a quick tour of Snort's usefulness in an enterprise network.

Viruses, Worms, and Snort

Within days if not hours of the release of a new worm, Snort signatures are being written for it. Those signatures are often incorporated into the main Snort ruleset, so that all Snort users can benefit from them. Signatures for SQL Slammer were out on the NANOG mailing list within hours of the initial detection of the worm (www.merit.edu/mail.archives/nanog/2003-01/msg00775.html). Signatures for the MyDoom.A worm were out within a day of the initial detects by antivirus labs. This type of quick responsiveness allows Snort users to update their rulesets when a new attack comes out, and begin detection and remediation of their vulnerabilities sooner. In fact, if you use some of the add-ons that are available for Snort, you can actually detect signs of worm propagation before signatures are available.

Known Exploit Tools and Snort

Snort has many signatures that are tailored to let you know when a known exploit tool is being used against your network. Some of these tools are marked by their self-advertising in the packet payloads, like the SolarWinds ICMP and SNMP scanner. Here's the Snort signature (www.snort.org/snort-db/sid.html?sid=1918):

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SolarWinds IP scan
attempt"; content:"SolarWinds.Net"; itype:8; icode:0; classtype:network-
scan; sid:1918; rev:3;)
```

Note the "SolarWinds.Net" content in the ICMP echo packet. In this case, that's the fingerprint of the tool. However, not all known exploit tools are quite so self-advertising. Consider this signature, for a Trin00 attacker client attempting to connect to the Trin00 master server on the default port with the default password:

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 44


```
alert tcp $EXTERNAL_NET any -> $HOME_NET 27665 (msg:"DDOS Trin00 Attacker
to Master default startup password"; flow:established,to_server;
content:"betaalmostdone"; reference:arachnids,197; classtype:attempted-dos;
sid:233; rev:3;)
```

Although many of the Snort signatures are written as generically as possible to allow you to see the attack no matter which tool was used to generate it, the rule authors won't hesitate to write a rule for a particular tool as well if one should flag itself in a clear fashion.

Writing Your Own Signatures with Snort

It should now be obvious that one of the greatest strengths of Snort is the ability to write customized rules for your network and the traffic you see. The syntax is precise and flexible, allowing you to match all sorts of different network traffic. Chapter 5 in this book covers writing rules for Snort, and additional information can be found online at www.snort.org/.

Using an IDS to Monitor Your Company Policy

A common use of customized Snort rules is to monitor traffic that, while not actively malicious, is restricted or frowned upon by company policy. Some enterprises write rules to alert them when their users access a Web page with content matching particular keywords, or a site with unauthorized software, or other policy violations. Snort actually comes with a set of rules for traffic that is likely to be pornography. You can even write your own Snort rules to match any type of network traffic, letting you know when someone has shut down the mail server and started up the Quake server.

Analyzing Your IDS Design and Investment

Once you have decided which type(s) of IDSs you want to deploy and where you'd like to place them in your network, it's time to give some thoughtful consideration to how you might improve your design. Are you likely to be inundated with false alerts, or miss alerts you would like to see? Could a real attack slip by in the midst of a storm of false positives?

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 45

False Positives versus False Negatives

When trying to establish an IDS policy, one expects to be inundated with false positives; at least until some IDS tuning has been done to get them down to a manageable roar. More concerning, however, is the possibility of false negatives, those attacks that the IDS misses. It is all too easy to be lulled into a false sense of security—seeing many alerts every day often gives us the impression that since we're seeing so many potential attacks, surely we must be seeing them all. However, skilled attackers can scan and code their exploits specifically to be stealthy and not detected. There are a variety of techniques available for doing this, which we will discuss.

Fooling an IDS

The Ptacek & Newsham paper previously mentioned discusses many individual techniques for fooling a NIDS, but in general, there are two main approaches. One approach is to give it so much data that it chokes on it, either missing packets or drowning the administrator in so many alerts that she never sees the real attack. The other general approach is to frame your attack in such a way that it won't match the signatures or algorithms that the IDS is using to pull out the attacks from the network background noise. The former technique is what the tools Stick and Snot depend on, as well as Nmap's decoy scan. The latter technique is what the stealth Nmap scans and tools like Dug Song's fragrouter or Rain Forest Puppy's Whiskeruse.

IDS Evasion Techniques

First, let's look at the noisy way. Stick and Snot (see the sidebar) are tools designed to generate as many alerts as possible on your IDS. They do this by generating alerts from a ruleset that is likely similar to the ruleset your IDS is using to match traffic. Some miscreants hope to slip in some attack traffic while you're distracted by all the false positives, or while your IDS is dropping packets. Others just like the idea of killing your IDS.

If the attacker used Stick or Snot to cover his tracks and then launched a TCP attack, this could be easily compensated for by only having Snort alert on established TCP sessions. However, this would be an ideal time for the attacker to launch a UDP-based attack—Remote Procedure Call (RPC), DNS, something like that.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 46

For maximum stealth, the attacker could even spoof the source; that doesn't matter in connectionless UDP. There is some likelihood that the attack packets would get dropped if the network links were too oversaturated with the Stick/Snot output, but it is likely that the actual attack packets would not be picked up by the IDS, either because it's only listening to established TCP sessions and our attack is UDP or ICMP, or because the IDS is still listening to all connections but is mobbed with false positives.

Notes from the Underground…

Stick, Snot, and Snort

Stick, Snot, and Snort are tools billed as "IDS Killers," designed to overload your IDS to the point it becomes unusable.

- **Stick** (www.eurocompton.net/stick/projects8.html) is a C program based on an old version of the Snort ruleset, designed to spew out so many alert-triggering packets per second that it would force IDSs to come to a grinding halt. It was very effective for its time, but Snort now has measures in place to adjust to and compensate for this style of attack.
- **Snot** is another similar tool (www.stolenshoes.net/sniph/index.html) that takes a Snort ruleset as argument and generates a series of packets that will trigger that ruleset. Cross-platform and flexible, Snot allows script kiddies all over the world to annoy to their IDS administrators.

If your Snort installation is being harried by these tools or similar ones, you can limit your Snort alerts to noticing established TCP sessions only with the *snort –z est* arguments. For this to work, however, the stream4 preprocessor must be configured. Also keep in mind that this will limit you from seeing all other nonstateful TCP alerts, so you will be missing UDP, ICMP, and ARP-based alerts. However, your IDS will still be up and running. We go into depth on configuring snort in Chapter 3, "Installing Snort."

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 47

Nmap offers a noisy scan that generates a whole bunch of fake packets as alternate "sources," using the –D "decoy" option. To the target, it looks like they are being scanned by all the decoy machines at once, and your real scan is masked among the fake ones.

Now, the quiet way. These are the attackers you really need to worry about. We have already described fragroute and Dug Song's evasive techniques as laid out in the original Newsham-Ptacek paper, but Nmap also offers options for stealth. There is the idle scan, the FTP bounce attack, timing-based attacks like a very slow scan stretched out over days, fragmentation and reassembly based attacks, TCP flag combination attacks, and even an idle scan off an unwitting zombie host. To read details about the packet construction behind all these attacks, refer to the Nmap man page at www.insecure.org/nmap/data/nmap_manpage.html.

Return on Investment—Is It Worth It?

At the end of the day, the deciding factor for many businesses is what the expected return on investment is. Is there truly going to be enough enhancement to your network security that it's worth installing, configuring, and maintaining an IDS? Security is often referred to as an economic sinkhole for businesses; they spend money on it, but if all goes well, they rarely see returns. Instead, the returns are in costs saved rather than in products made. Because of this, many CEOs are reluctant to spend the money necessary for expensive systems or solutions, more so if they've already spent money on an IDS and have seen few positive results from it but many false positives.

If you are considering adding an IDS to your network, consider it as a business case. How much money does your company lose if there is an intrusion? What are the odds of that intrusion happening? How much will it cost to install and maintain an IDS? How much will the IDS offset or mitigate the risks of that intrusion? How will an IDS affect your organization legally? Earlier in the chapter, we discussed the possible implications of wiretap and privacy laws on a company's use of an IDS. However, an IDS can also assist in compliance with corporate accounting laws such as the Sarbanes-Oxley requirements, and in establishing an audit trail in the event of a compromise. Sections 302 and 304 of the Sarbanes-Oxley requirements place the responsibility on a corporation to establish internal controls within their network. An IDS can be a demonstrable part of these controls. When combined with a third-party penetration test of your network security, this can go a long way toward validating your own data

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 48

with an external audit, complete with trail. Some locations now require companies to notify customers when their data has been compromised; the State of California is one such place. Having an IDS can allow you to detect compromise attempts more reliably. Being able to go to your CEO with strong numbers, legal backing, and business precedent will be far more impressive than "uh, I guess we need one of those, everyone else seems to have one."

Defining IDS Terminology

Being able to understand the differences between different types of IDSs and their features is crucial when trying to design a security architecture. Let's look at some of the most common terminology in the IDS field, and make sure we understand all the options available.

Intrusion Prevention Systems (HIPS and NIPS)

An IDS that not only detects possible attack, but also responds to prevent the attack from being successful. This response can be anything from creating firewall rules to black-hole the attacker, to killing the offending process (when dealing with a Host IPS), to dropping the offending traffic (when dealing with a Network IPS).

Gateway IDS

An IDS that sits at the bottleneck between your network and the Internet (or whatever peering upstream you may be connected to). Also known as an inline IDS, all traffic must pass through this gateway to leave your local network. This may also function as an IPS if it includes the capability to make decisions about whether traffic should be allowed.

Network Node IDS

The method of intrusion detection where one establishes a baseline of "normal" network traffic, and then looks for deviations from that norm and flags them as possible attack traffic.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 49

Protocol Analysis

The method of intrusion detection where one looks at the flow of data within the specifications of each protocol, looking for anomalies and possible malicious traffic based on the expected protocol behavior.

Target-Based IDS

A new flavor of IDSs specifically aimed at what is actually on the network. They are designed to have fewer false positives and only alert on attacks that are relevant to your network and the specific services running on your network.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 50

Summary

IDSs can serve many purposes in a defense-in-depth architecture. In addition to identifying attacks and suspicious activity, you can use IDS data to identify security vulnerabilities and weaknesses.

IDSs can audit and enforce security policy. For example, if your security policy prohibits the use of file-sharing applications such as Kazaa, Gnutella, or messaging services such as Internet Relay Chat (IRC) or Instant Messenger, you could configure your IDS to detect and report this breach of policy.

IDSs are an invaluable source of evidence. Logs from an IDS can become an important part of computer forensics and incident-handling efforts. Detection systems are used to detect insider attacks by monitoring traffic from Trojans or malicious code and can be used as incident management tools to track an attack.

Correlation of data, whether from a HIDS or NIDS or DIDS, is probably the best way to approach intrusion detection data. While an IDS can be a valuable contributor to a security architecture, it is by no means enough in and of itself to protect a network.

A NIDS can be used to record and correlate malicious network activities. The NIDS is stealthy and can be implemented to passively monitor or to react to an intrusion. The HIDS plays a vital role in a defense-in-depth posture; it represents the last bastion of hope in an attack. If the attacker has bypassed all of the perimeter defenses, the HIDS might be the only thing preventing total compromise. The HIDS resides on the host machine and is responsible for packet inspection to and from that host only. It can monitor encrypted traffic at the host level, and is useful for correlating attacks that are detected by different network sensors. Used in this manner it can determine whether the attack was successful. The logs from a HIDS can be a vital resource in reconstructing an attack or determining the severity of an incident.

Solutions Fast Track

Introducing Intrusion Detection Systems

- ☑ An intrusion is an unauthorized access, use, or attack on your network or computers.
- ☑ IDSs work by watching network and system activity, and comparing that to known signatures or against algorithms to separate legitimate activity from suspicious activity.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 51

- ☑ IDSs can then log the attack and respond in a number of ways. The most common response is to alert the system administrators through SNMP traps, text messages, phone calls, or pages.

Answering Common IDS Questions

- ☑ Attackers are interested in everyone connected to the Internet these days; it's not necessarily personal.
- ☑ An IDS can alert you to network traffic and system activity of which you may not have been aware. It can increase the effectiveness of a good system administrator, and provide him with additional data.
- ☑ An IDS will not replace your existing security staff, or make people stop attacking you.

Fitting Snort into Your Security Policy

- ☑ Snort is a network IDS with sophisticated pattern-matching capabilities that are used to uniquely describe attack traffic.
- ☑ Snort signatures for the latest viruses, worms, and other new vulnerabilities are usually written and released within hours or days of the new attacks' debut.
- ☑ You can write your own Snort signatures to match company policy violation, new or unique traffic, or anything else.

Analyzing IDS Design and Architecture

- ☑ IDSs can be configured to just detect and alert, or to respond as well.
- ☑ Possible responses include dropping the traffic, spoofing ICMP or TCP Reset packets, or identifying and tracing back toward the attack source.
- ☑ IDSs are not perfect or foolproof—they can be tricked or eluded. They are valuable contributors to a security policy, but not enough all by themselves to enforce it.

295_Snort_2e_01.qxd 5/4/04 4:50 PM Page 52

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: Why doesn't my firewall serve as an IDS?

A: Firewalls are designed primarily to pass, drop, or reject traffic, not to alert on suspicious traffic. IDSs are designed to let you know when suspicious activity is occurring. The two functions are different and conflict in key issues. We discuss this further in Chapter 12.

Q: Can IDSs gather data from anywhere besides sniffing on a network?

A: Yes, some IDSs can also gather data from log parsing, watching system calls, or monitoring a filesystem.

Q: What can an IDS do for me that my system administrator can't?

A: Parse a few hundred million packets or log entries (or more) a day in binary. Most administrators get tired after a while.

Q: What can my system administrator do for me that my IDS can't?

A: Bring creative thinking and an understanding of the significance of this network activity to the analysis.

Q: Will I have to spend time tuning my IDS?

A: Yes. If you don't want to be drowning in false positives, it really is best to tune your IDS to fit its environment.

Q: Does physical security still matter if I have the best network security in the world?

A: Absolutely. If we can walk in to your office and walk out with your server, you've still been rooted.

Q: Why should I bother writing my own signatures, when Snort has so many already?

A: You certainly don't have to, but you might want to add functionality that's not present in the extant ruleset, like rules tailored to your enterprise policy or to detect attacks targeting specific proprietary applications.