CHAPTER 17

The Future of Intrusion Detection and Prevention

As mentioned in previous chapters, many changes are in store for intrusion detection and intrusion prevention. Some of these changes could actually be negative—at least from the perspective of intrusion detection. For example, the Gartner Group, a technology research and consulting organization, asserts that IDSs will soon be relics of the past. Gartner says that IDSs have not established themselves in the IT marketplace, that they produce too low a return on investment (ROI) for all the resources expended, and that excessive false alarms and misses have greatly impaired their usefulness. Gartner predicts that intrusion prevention technology will prevail in the belief that shutting off intrusions altogether is better than allowing intrusions to occur and just monitoring them. Accordingly, Gartner recommends that IT organizations turn to firewalls, not IDSs. Many IT security experts denounced Gartner's prediction, though, saying that Gartner does not really understand how intrusion detection fits in with a layered defense approach (of which many believe that intrusion detection is a critical part) and that intrusion detection technology is still growing and improving.

Regardless of whatever sliver of truth there may or may not be in Gartner's prediction, two things are certain—intrusion detection is still a long way from being mature, and intrusion prevention technology is in its infancy. Massive changes are in store for both areas. This chapter focuses on some of the areas within intrusion detection and intrusion prevention in which substantial and beneficial progress is likely to occur. These areas include the following:

- The continued reduction in reliance on signatures in intrusion detection
- The growth of intrusion prevention
- Advances in data correlation and alert correlation methods
- Advances in source determination
- Inclusion of integrated forensics functionality in IDSs and IPSs
- Greater use of honeypots

We'll begin by considering why signature-based intrusion detection will become less mainstream in the future.

LOWER RELIANCE ON SIGNATURE-BASED INTRUSION DETECTION

The signature approach to intrusion detection, which traces back to the early 1990s, represents a major advance over the previous statistical-based approaches of the 1980s. Signatures are not only a relatively straightforward and intuitive approach to intrusion detection, but they are also efficient—often a set of only a few hundred signatures can result in reasonably high detection rates (albeit often at the cost of false alarm rates, as discussed earlier). Signature-based IDSs have proven popular and useful, so much so that you can count of some of these tools being available for a long time.

Signature-based intrusion detection is beset with numerous limitations, however, including the following:

- Because attacks have to occur before their signatures can be identified, signatures cannot be used in discovering new attacks. The “white hat” community is thus always one step behind the “black hat” community when it comes to new attack signatures.
- Many signatures in IDSs are badly outdated. One commercial IDS for many years contained the signature for a Unix exploit in which an attacker could enter the `rlogin` command with the `–froot` switch to obtain a root shell on a victim system. This exploit was for early versions of the AIX operating system, versions almost never used anymore. You can always “weed out” obsolete signatures, but doing so requires a reasonable amount of unnecessary effort; good IDS vendors do not include such signatures in their products’ signature sets in the first place.
- Some attacks do not have single distinguishing signatures, but rather a wide range of possible variations. Each variation could conceivably be incorporated into a signature set, but doing so inflates the number of signatures, potentially hurting IDS performance. Additionally, keeping up with each possible variation is for all practical purposes an impossible task.
- Signatures are almost useless in network-based IDSs when network traffic is encrypted.
- The black hat community is becoming increasingly better in evading signature-based IDSs, as discussed in the sidebar “IDS Evasion Tools.”

IDS Evasion Tools

The number of methods for evading signature-based IDSs has been increasing dramatically over the last few years. For example, powerful tools that can defeat signature-based IDSs are available. One class of tools (such as Fragroute) launches "insertion attacks,” in which malicious commands sent to a server are disguised by inserting extra, bogus data. When the IDS processes the traffic in which these commands are embedded, the IDS does not recognize anything as an attack signature, but when the destination server processes the input it receives, it discards the extra data, allowing the commands to execute.

For example, an attacker can send the following command to a web server:

```
GET //cgi-bin//some.cgi.
```

The web server cannot recognize `some.cgi`, so it may discard this part of the input, connecting the attacker to `cgi-bin` instead. Unfortunately, `cgi-bin is` the

IDS Evasion Tools *(continued)*

directory for common gateway interface scripts, one of the places just about every web hacker wants to be.

Another trick is to insert a premature null character:

```
GET%00 /cgi-bin/some.cgi HTTP/1.0
```

Alternatively, an attacker can send ASCII-coded input that invokes a malicious command to an interpreter:

```
perpetrator@host$ perl -e
'$bad=pack("C11",47,101,116,99,47,112,97,115,115,119,100);
@hack='/bin/cat/ $bad'; print"@hack\n";’
```

Another evasion method is rexmit inconsistency. An attacker sends a TCP stream in which some of the data within the stream is garbled. The receiving host (intended victim) sends a message to the sending host (the attacking host) asking it to retransmit. The sending host then sends malicious commands. The IDS analyzes the first stream and determines that it does not match any attack signatures. However, for efficiency's sake, the IDS may not analyze the second stream (which in theory should be identical to the original one), resulting in a missed attack. Many other evasion techniques work against signature-based IDSs, too. IDS evasion techniques do not exclusively target signature-based IDSs, however. There are evasion techniques for rule-based IDSs, too, for example.

If reliance on signatures in intrusion detection will dwindle in the future, what intrusion detection methods are likely to become increasing important? Several alternatives discussed in the next section appear probable.

Protocol Analysis

Protocol analysis means analyzing the behavior of protocols to determine whether one host is communicating normally with another. For example, the TCP handshake (discussed in Chapter 2) is initiated by sending a TCP SYN packet to another host. The other host responds with a SYN ACK packet, to which the originating host responds with an ACK packet. Suppose that a host sends nothing but SYN packets to another host—an indication of a "SYN flood" attack designed to deplete memory and other resources in the receiving host. In another kind of protocol attack, a host might send malformed IP packets, perhaps IP packets in which one or more values in the IP header is out of range. In still another, a malicious code may send malformed "chunks," parcels in which data are transferred from a browser to a web server to provide an orderly way for the web server to encode the input.

Although these are simple examples, protocol analysis is by no means any kind of "lightweight" way of performing intrusion detection. A wide range of attacks (particularly DoS attacks) can be detected in terms of anomalous protocol behavior. Identifiable signatures may exist for many of the same attacks, but identifying these attacks at a lower level of networking (such as the network or transport layer by looking at the behavior of protocols such as IP, TCP, UDP, and ICMP) is more efficient than having to go to a higher layer. The rules of normal protocol behavior are well defined in RFCs (see www.ietf.com/rfc.html), so deviation is usually (but by no means always, given that a certain percentage of network traffic does not behave in accordance with any RFC) rather straightforward to determine. Additionally, many attacks that would require literally scores of signatures to detect can often be identified in terms of only a very few protocol behavior irregularities. Many of today's IDSs perform protocol analysis; IDSs of the future are likely to do more and also do it better.

Target Detection

We're also likely to see more widespread use of target detection in the future. As mentioned previously, target detection has proven to be one of the most robust and reliable methods of intrusion detection. Attackers almost invariably make changes in systems, often to create back doors, but sometimes (especially in the case of novice attackers) changes occur simply by accident. Attackers may be able to evade signature-based IDSs, and they may also be able to delete system logs to hide evidence of their activity, but they are less likely to escape the notice of a target detection tool that uses a variety of strong cryptographic algorithms and requires strong authentication for access to the target detection functions.

Although commercial target detection tools such as Tripwire (http://www.tripwiresecurity.com/) and Intruder Alert (http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=171) are widely used within Fortune 500 companies, the price of deploying these tools on many systems often serves as a deterrent to their use in smaller organizations. Freeware versions of Tripwire (http://ftp.cerias.purdue.edu/pub/tools/unix/ids/ tripwire/) and Windows-based integrity checking tools such as ForixNT (http://www.incident-response.org/forix-nt.htm) are available, but hurdles such as worries over software support have at least to some degree inhibited their widespread use.

Commercial target detection tools have established themselves in the marketplace; they will not disappear any time in the near future. What is likely to happen, however, is that operating system and application vendors will build powerful integrity checking capabilities into their products. To at least some degree, vendors already do this. Unix vendors, for example, have for a long time included the `sum` command for computing simple cryptochecksums and the `diff` command for detecting changes in file contents. Microsoft also includes the System File Checker and Windows File Protection in many of its operating system products. But these capabilities are relatively crude compared to the capabilities of many commercial tools.

Vendors are likely in time to expand the scope of integrity checking programs to include more than simply device driver or system file checking capabilities and also to provide real-time alerting capabilities. It would not be surprising, for example, to learn sometime in the near future that one or more vendors had incorporated the commercial Tripwire tool into an operating system. Doing this would make target detection easier to manage and possibly also more secure, given that operating system defenses could also be used to protect target detection executables and data files.

NOTE In the past generation (and in all likelihood, the future) of kernel-based Unix exploits, the existence of changes to the file system has almost always been well hidden from system administrators and users. An example is the sk Linux rootkit, which initially was found in systems only because of numerous implementation bugs. Ultimately, kernel-based exploits are the greatest threat to target-based IDSs, in which detection depends on subtle changes in the behavior of the system.

Rule-Based Intrusion Detection

Rule-based intrusion detection is more of an eclectic approach than the other alternatives to signature-based intrusion detection covered in this section. In this approach, logic conditions based on possible incident-related observations are defined. Observations could be signatures, irregularities in protocol behavior, unusual system events, changes in files and/or directories, and so on. Rule-based intrusion detection analyzes elements derived from these observations and then uses logic to identify attacks. For example, suppose that element A is defined as a probe from a certain IP address, that B is defined as attempted access via anonymous FTP, and that C is defined as an attempt to obtain the password file. If A or the combination of B and C occur, this could be defined as an attack pattern.

Rule-based intrusion detection is used in a number of IDSs today, especially prototype systems used in connection with intrusion-detection research. The rule-based approach is potentially more powerful than signature-based intrusion detection because it relies on multiple variables/indicators—events based on signatures, protocol analysis, target detection indicators, and so on. Because this approach seldom equates a single event with a rule, it is likely to produce a higher hit and lower false alarm rate than signature-based intrusion detection. The higher hit rate is particularly significant. An attack may manifest itself in multiple but nondeterministic ways; rule-based intrusion detection can define long strings of "or" rules, one of which might apply to a particular set of observations, enabling it to detect attacks that simple signature-based IDSs might very well miss.

The main limitation of rule-based intrusion detection is the potential complexity associated with all the rules that are normally created. Only those with advanced technical skills and knowledge are likely to be able to understand the rules in the first place. It generally is difficult to create rules (which can often involve many steps of logic) and also to maintain rules (for example, weeding out obsolete rules). Processing the rules themselves can also cause massive CPU and memory utilization in the host that houses a rule-based intrusion detection system. Still, rule-based intrusion detection represents a significant advance over simple signature-based intrusion detection; it is likely to be used increasingly over time.

Rule-Based Intrusion Detection

Rule-based intrusion detection can involve long sets of complex rules, something that may make this approach seem nebulous and impractical. Rules can, however, be as simple as needed. The tcpdump tool discussed in Chapter 5 provides an almost ideal example how rules can be created at the protocol level. Consider, for example, the following tcpdump expression:

```
(tcp src port 27374 ) and ( tcp[2] > 3 )) or ((tcp dst port 27374)
and (tcp[0] > 3))
```

This expression represents a rather simple rule—the source port must be TCP 27374 and the 2nd byte of the TCP header (destination port) must be at least 4×256 or 1024 OR the destination port must be TCP 27374 and the 0th byte of the TCP header (source port) must be at least 4×256 or 1024. A rule-based intrusion detection system could include and use this type of rule to detect the presence of the deadly SubSeven Trojan horse remote control program in Windows systems. If there is a connection from port 27374, the most used port in connection with this particular Trojan, to an ephemeral port or a connection to port 27374 from an ephemeral port, the IDS would in this example report an attack.

Neural Networks

Neural networks are systems that perform pattern recognition on inputs they receive based on models of how neurons in mammals process information. Neurons are nerve cells; they are densely interconnected and interface with each other at *synapses*, small gaps between individual neurons. They also work in parallel to other neurons at any given level of brain structure. Neural networks are sets of mathematical models that imitate how neurons learn, assigning different weights to connections between elements within the neural network similarly to how electrical potentials for neurons are built up at synaptic junctions based on their frequency of firing. The more frequently a neighboring neuron fires, the more electrical potential there is at the synapses of the neurons that react to this pattern. In neural networks, elements that receive inputs from neighboring elements receive higher weights.

Although complicated and still somewhat mysterious, the neural networks approach can be applied to a wide range of pattern recognition problems, intrusion detection included. The beauty of neural networks in intrusion detection is that no signatures or even rules are needed. You simply start feeding input—data concerning network- or host-based events—to a neural network, and it does the rest. Neural networks are, therefore, well suited to picking up new patterns of attacks readily, although some learning time is required. The neural networks approach has been around for a long time, and if anything it is likely to become more widely used and relied on in intrusion detection in the future as reliance on signatures diminishes.

INTRUSION PREVENTION

Intrusion prevention is another area that will grow dramatically in the future. Intrusion prevention is in its infancy. Anyone who thinks that IPSs and IDSs are diametrically opposed or that IPSs will eventually supplant IDSs is badly mistaken, however. An IDS is like a burglar alarm, something that provides information about past and ongoing activity that facilitates risk and threat assessment as well as investigations of suspicious and possibly wrongful activity. IPSs are designed to be defensive measures that stop or at least limit the negative consequences of attacks on systems and networks, not to yield the wealth of information that IDSs typically deliver.

The number of potential, useful variations in "intrusion prevention" is mind-boggling. Consider, for example, the first type of intrusion prevention used in connection with intrusion detection—*shunning*. Shunning is a mainstay feature in today's IDSs, yet shunning is, all things considered, a rather crude way of performing intrusion prevention. In all likelihood, malicious packets will already have arrived at the intended victim host by the time any firewall or router ACLs are changed to block future packets from the apparent attacking host—not exactly the desired results if shutting off the attack in the first place is the goal.

IPSs such as Cisco's Okena StormSystem product (see https://www.okena.com/pdf/stormwatch_datasheet.pdf) represent another extreme in the intrusion prevention continuum in that a number of hosts can be spared from having to suffer the malicious consequences of an attack because they have received a policy change based on detected malicious activity on the network. Others view intrusion prevention in terms of a set of interrelated (and very possibly cooperating) devices and capabilities that work together to diagnose system and network events and shut off incidents at any point where they can be shut off.

One of the major, new offshoots of the last permutation of intrusion prevention discussed here is called "active defense" (as opposed to "passive defense," such as passively monitoring systems and networks and deploying static access control lists [ACLs] in routers and firewalls). Active defense means analyzing the condition of systems and networks and doing what is appropriate to deal with whatever is wrong. According to Dave Dittrich of the University of Washington, there are four levels of active defense:

- Local data collection, analysis, and blocking
- Remote collection of external data
- Remote collection of internal data
- Remote data alteration, attack suppression, and "interdiction"

Figure 17-1 portrays one possible active defense architecture. Numerous hosts within a hypothetical network collect intrusion detection data and send them to a central analyzer that, whenever appropriate, sends policy changes to individual hosts to keep them from executing certain instructions in memory, changing the content of certain files, and so forth. The external firewall, the outermost layer in the active defense infrastructure, detects relatively straightforward attacks such as SYN flooding attacks and shuns offending IP addresses immediately. A network-based IDS within this network gathers information from sensors at the external gateway and at entrances to several subsets; it sends ACLs changes to the firewall and data to the central analyzer on the basis of attack patterns that it deciphers.

One of the most important (and controversial) facets of the active defense approach to intrusion prevention is determining the appropriate response. The notion of appropriate response includes a consideration called “ proportionality of response,” which ensures that the response is proportional to the threat. In the case of a host that is flooding a network with fragmented packets, blocking traffic sent from that host is almost certainly the

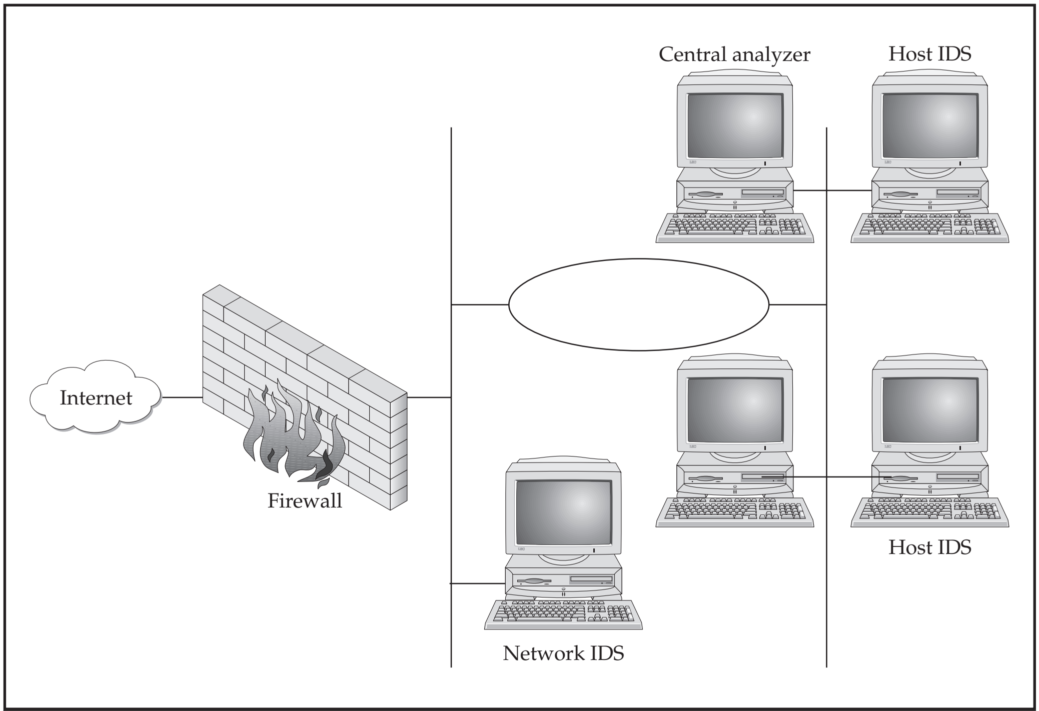

Figure 17-1. A simple active defense architecture

most appropriate response. If several dozen hosts known to be operated by an ISP repeatedly attack an organization's network, blocking all the traffic from the range of IP addresses owned by that ISP might be the most appropriate response. Some advocates of the active defense approach even believe that if a remote host is repeatedly attacking an organization's network, counterattacking that host, perhaps by flooding it with fragmented packets, thereby causing it to crash, is the appropriate course of action. Many, the authors of this book included, strongly disagree with the notion of counterattacking, however, as discussed in the "Striking Back" sidebar.

Although intrusion prevention appears promising, (as mentioned) it is very much in its infancy. Attack stave-off rates for intrusion prevention systems are nowhere as high as they need to pose a major deterrent to attacks. Additionally, false alarms can easily cause what effectively amounts to DoS within individual systems. Intrusion prevention systems of the future are likely to be able to prevent a wider range of attacks, not only at the level of the individual host, but also within organizations' networks and possibly even within the Internet itself. The last possibility is particularly intriguing. Perhaps some organization such as the U.S. government's federal incident response team, FedCIRT, will continuously monitor all traffic bound for U.S. government sites and stop selectively malicious packets long before they reach the gateways of the government sites for which they are destined.

Striking Back

"Striking back" has recently been a hot topic within information security and other discussion groups. Some advocate doing whatever is necessary to prevent attacks from a known source of trouble. In some cases this would mean causing DoS in the offending host; in other cases, it might mean destroying the offending host altogether by breaking into the host and then erasing critical system files. The lack of sufficiently strong cybercrime legislation throughout the world makes this approach more attractive to its advocates; some individuals even think that it is important to teach attackers a lesson by letting them "have a taste of their own medicine."

Striking back has raised a plethora of ethical controversies, so many that at this point in time, the information security community as a whole is firmly opposed to it. Striking back is also illegal in numerous countries around the world. But in the military arena, striking back could well be the appropriate course of action—an important countermeasure in information warfare. If, for example, an opposing military force were causing DoS in systems used for military intelligence analysis, shutting out the source of the attacks would make perfect sense. The striking back issue is likely to grow in connection with the active defense approach to intrusion prevention over time.

Striking Back *(continued)*

But striking back is dangerous, especially from a legal aspect. Anyone who considers deploying strike-back methods should at a minimum obtain management's approval and also (if management approves striking back) consult one's legal department before proceeding. Remember at the same time that you may end up getting sued by the individual or organization whose host you attacked, even if you have management approval.

DATA AND ALERT CORRELATION

As mentioned in Chapter 12, data correlation is becoming increasingly important. IDSs, IPSs, firewalls, personal firewalls, and TCP wrappers are each capable of generating large amounts of data; collectively, they are capable of overwhelming intrusion detection analysts with data. Data aggregation helps ensure that data are available in a single location; data correlation enables analysts to recognize patterns in these data. Although current data correlation methods are for the most part not very sophisticated, future data correlation is likely to become much better. How will data correlation algorithms need to change? Waltz and Llinas (in *Multisensor Data Fusion*, Boston: Artech House, 1990) have developed criteria for systems designed to fuse data must be able to, saying that these systems must be able to do the following:

- Distinguish parameters of interest (hit rate, range of events detected, and so on) from noise
- Distinguish among different objects in space and time
- Adequately track and capture each desired type of event and data
- Sample the data and events of interest with sufficient frequency
- Provide accurate measurements
- Ensure that each variable that is measured adequately represents the desired types of categories
- Provide access to both raw and correlated data
- Preserve the temporal characteristics of data and events

It is unlikely that all systems designed to fuse data will meet every one of these requirements. The more of these requirements that a system meets, however, the more useful in data fusion/correlation it is likely to be. Currently, one of the greatest barriers to automated data fusion has been the lack of a common format for data from intrusion detection systems. Although common formats have been proposed, little agreement has resulted. Agreement upon a single data format would thus constitute a giant step forward.

Additionally, user interfaces of applications that perform data correlation are likely to improve dramatically in the future. Deficits in the usability of software used in data correlation are by no means unique; usability problems plague the information security arena as a whole. Human-computer interaction methods for controlling what is displayed tend to be nonintuitive and excessively complex. Data displays are often cluttered to the point of being overwhelming to view; few options for data reduction or displaying a patterns and profiles typically are available. Fortunately, vendors are addressing these problems in their products and are already incorporating substantially improved user interfaces into these products.

The user interface of ArcSight, a current commercial product used for correlating intrusion detection data, allows flexibility in what is displayed, color codes different conditions appropriately, and offers a variety of pattern displays such as pie charts.

Alert fusion (also covered in Chapter 12) is a closely related area that is likely to become increasingly important over time. Improved algorithms for alert fusion are likely to be developed; as they are, alerting will become more efficient in that fewer alerts will be necessary to warn analysts and operators about a series of events, some of which may be related, others of which may not.

SOURCE DETERMINATION

Source determination means determining the origin of network traffic. Given how easy it is to spoof IP addresses, any source IP address in conventional IP packets must be viewed with suspicion. Tools that fabricate packets, inserting any desired IP address into the IP headers, are freely available on the Internet. Many countermeasures, most notably strong authentication methods (such as the use of Smart Cards) and digital signatures, can remove doubt concerning the identity of individuals who initiate transactions, but they are not designed to identify the source IP addresses from which transactions originate. IPsec, the secure IP protocol, effectively removes any doubt concerning the validity of IP source addresses, but IPsec has, unfortunately, not grown in popularity in proportion to its many merits.

New source determination methods that are potentially valuable in intrusion detection and intrusion prevention efforts are emerging. Some are rather crude (but nevertheless potentially useful), such as measuring the latency between the time a packet arrives at a network node and the time it moves on to the next destination. The longer the latency, the further away the origin of the connection is (at least in theory—in reality, latency can fluctuate based on traffic conditions or routing table changes.) Another intriguing source determination method measures the difference in the packet sequence numbers of packets within TCP streams over time. Packets with little deviation in the packet sequence number over time can be assumed to be part of the same stream. In still another method, packets are marked at each hop over which they travel. Each intermediate switching point on the network adds its signature or identifier to each packet, enabling investigators to determine exactly which route a packet took as it traveled from the source to the destination host.

Source determination is, unfortunately, yet another area in its relative infancy. Advances in this area are particularly important. Knowing exactly where an attack has originated is critical in determining the appropriate course of action. As someone who is involved with intrusion detection virtually every day, I often wonder how many IP addresses that are shunned are addresses of hosts that have actually launched attacks and how many are completely innocent of any wrongdoing, unrelated to anything that an intrusion detection system has reported. Readily determining the source of attacks is also extremely important for the sake of law enforcement; it can eliminate much of the work that is normally involved in investigating exactly where each attack has originated.

INTEGRATED FORENSICS CAPABILITIES

Forensics means using special procedures that preserve evidence for legal purposes. When people think of forensics, they normally envision investigators archiving the contents of hard drives to a machine that runs forensics software, making hard copies of audit logs, and labeling and bagging peripherals such as keyboards and mice. Many people fail to realize that IDSs are potentially one of the best sources of forensics data, especially if the IDSs capture and store keystrokes. A few IDS vendors are starting to build forensics capabilities into their products, capabilities that enable those who use the systems to make copies of IDS output, create a hash value of the output (to ensure its integrity), search it for special keywords or graphic content, and so on.

Having an integrated forensics capability is potentially advantageous in that everything that is needed for forensics purposes resides on a single machine (or possibly in some circumstances, a few machines) with a single user interface, greatly simplifying the process of obtaining and preserving evidence. Sophisticated forensics capabilities are likely to routinely be built into IDSs (or at least high-end IDSs) and possibly also IPSs in the future.

USE OF HONEYPOTS IN INTRUSION DETECTION AND PREVENTION

A *honeypot* is a decoy server that looks and acts like a normal server, but that does not run or support normal server functions. The main purpose of deploying honeypots is to observe the behavior of attackers in a safe environment, one in which there is (at least in theory) no threat to normal, operational systems. Having proven especially useful as a reconnaissance tool that yields information concerning what kinds of attacks are occurring and how often, honeypots have gained a great deal of acceptance within the information security arena.

Honeypots are not, however, usually used in connection with intrusion detection and prevention efforts. This is unfortunate, because honeypots often glean information that conventional IDSs and IPSs miss. A honeypot server can, for example, be given an especially interesting name such as "peoplesoft6.abc.corp," attracting connection attempts that conventional servers and workstations would not normally receive. Provided a honeypot is set up and deployed properly, there is also little risk to the rest of the network, something that is not necessarily true for a host that runs intrusion detection software.

More effective honeypots create "virtual environments" that appear to be real but that in fact do not support a normal interactive environment. The result is that attackers cannot gain control over the honeypot and then use it to launch attacks against other machines within the network in additional to external hosts. Honeypots will almost certainly be used increasingly as a source of information for analyzers and ultimately as a basis for intrusion prevention policies that are sent to hosts.

FINAL CAVEAT

This chapter has presented a glimpse of how intrusion detection and intrusion prevention are likely to change in the future. These glimpses are, of course, nothing more than predictions, some of which will come true and others of which will not. Although certain predictions have been covered this chapter, other potential predictions have for brevity's sake been omitted. For example, it is reasonable to expect that IDSs of the future are likely to improve in their ability to deal with encrypted network traffic. Although it does not provide a general encryption solution, Sandstorm's NetIntercept IDS will (if each client has a backdoor SSH client and if NetIntercept is provided the correct encryption keys) allow replay of encrypted network sessions (see http://www.sandstorm.net/products/netintercept). This represents an advance for network IDSs that is likely to be followed by similar but improved functionality (such as ability to crack SSH encryption in clients not under the control of an organization that deploys such a tool) in future IDSs.

A considerable amount of research on data visualization methods for intrusion detection data is also currently being conducted. At some point, the major breakthroughs from this research will be incorporated into IDSs of the future, resulting in output that will be much more useful in terms of identifying threat magnitudes, patterns of elements within incidents, and so forth.

The intrusion detection and intrusion prevention arenas are extremely dynamic, with new findings, functions, and models being created all the time. At the same time, it is important to be wary of the claims of some vendors who add a few "bells and whistles" to their IDS or IPS products and then claim that their competitors' products are obsolete. Watch for the many changes that are currently occurring or are about to occur with great anticipation, but be sure to carefully evaluate each in terms of its genuine value to your organization's business and operational goals.

SUMMARY

This chapter has considered the future of intrusion detection and intrusion prevention. Great change is in store for both of these areas. First, given the significant pitfalls in the signature-based approach, there will continue to be less reliance on signatures in intrusion detection and intrusion prevention. Protocol analysis, target detection (using the output of cryptographic algorithms to detect unauthorized changes in files and directories), rule-based intrusion detection (using logic based on observations combinations of elements), and neural networks (systems that process inputs to recognize patterns based on models of how nerve cells process information) are viable alternatives to signature-based intrusion detection that are likely to grow in importance.

Intrusion prevention will continue to grow rapidly because of its capability to shut off attacks, potentially preventing damage and disruption altogether. The active defense approach, evaluating the condition of systems and networks and responding appropriately to remedy whatever is wrong, is new but already gaining rapidly in popularity. Advances in data correlation and alert fusion methods are also likely to occur. Correlation and fusion methods will meet a larger number of requirements and user interfaces for access to correlated data and are likely to improve substantially. Advances in the determination of the origin of network connections are also extremely probable. Finally, it is reasonable to expect that improved forensics functionality will be built into IDSs and IPSs in the future and that honeypots will be used much more in connection with intrusion detection and intrusion prevention.