put into place to ensure that one entity cannot carry out a critical task alone.
3. C. This is just one of several examples of separation of duties. A system must
be set up for proper code maintenance to take place when necessary, instead of allowing a programmer to make changes arbitrarily. These types of changes should go through a change control process and should have more entities involved than just one programmer.
4. C. Users should be able to access only the resources they need to fulfill the duties
of their positions. They also should only have the level of permissions and rights
for those resources that are required to carry out the exact operations they need
for their jobs, and no more. This second concept is more granular than the first,
but they have a symbiotic relationship.
5. A. Documentation is a very important part of the change control process. If things
are not properly documented, employees will forget what actually took place with each device. If the environment needs to be rebuilt, for example, it may be done incorrectly if the procedure was poorly or improperly documented. When new changes need to be implemented, the current infrastructure may not be totally understood. Continually documenting when virus signatures are updated would be overkill. The other answers contain events that certainly require documentation.

7. A. Manual iris lenses have a ring around the CCTV lens that can be manually turned and controlled. A lens that has a manual iris would be used in an area that
has fixed lighting, since the iris cannot self-adjust to changes of light. An auto iris

PART VII

6. A. The depth of field refers to the portion of the environment that is in focus
when shown on the monitor. The depth of field varies depending upon the size of the lens opening, the distance of the object being focused on, and the focal length
of the lens. The depth of field increases as the size of the lens opening decreases,
the subject distance increases, or the focal length of the lens decreases. So if you
want to cover a large area and not focus on specific items, it is best to use a wideangle lens and a small lens opening.

lens should be used in environments where the light changes, such as an outdoor setting. As the environment brightens, this is sensed by the iris, which automatically
adjusts itself. Security personnel will configure the CCTV to have a specific fixed

exposure value, which the iris is responsible for maintaining. The other answers are
true statements about CCTV lenses.
8. A. A transponder is a type of proximity-based access control device that does not require the user to slide a card through a reader. The reader and card communicate directly. The card and reader have a receiver, transmitter, and battery. The reader sends signals to the card to request information. The card sends the reader an access code.
9. A. Although many effective physical security mechanisms are on the market today,
none can look at a situation, make a judgment about it, and decide what the next step should be. A security guard is employed when an organization needs to have a countermeasure that can think and make decisions in different scenarios.
10. B. An electrostatic IDS creates an electrostatic field, which is just an electric field
associated with static electric charges. The IDS creates a balanced electrostatic
field between itself and the object being monitored. If an intruder comes within a certain range of the monitored object, there is capacitance change. The IDS can
detect this change and sound an alarm.
11. C. This type of system is sensitive to sounds and vibrations and detects the changes in the noise level of an area it is placed within. This level of sensitivity
can cause many false alarms. These devices do not emit any waves; they only listen for sounds within an area and are considered passive devices.
12. C. Every physical security program should have delaying mechanisms, which have
the purpose of slowing down an intruder so security personnel can be alerted and arrive at the scene. A warning sign is a deterrence control, not a delaying control.
13. D. A user-activated device requires the user to do something: swipe the card through the reader and/or enter a code. A system sensing device recognizes the presence of the card and communicates with it without the user needing to carry out any activity.
14. B. Intrusion detection systems are expensive, require someone to respond when
they set off an alarm, and, because of their level of sensitivity, can cause several
false alarms. Like any other type of technology or device, they have their own vulnerabilities that can be exploited and penetrated.
15. D. Cipher locks, also known as programmable locks, use keypads to control access into an area or facility. The lock can require a swipe card and a specific
combination that’s entered into the keypad.
16. A. A security guard would want to be alerted when a door has been open for an extended period. It may be an indication that something is taking place other than a person entering or exiting the door. A security system can have a threshold
set so that if the door is open past the defined time period, an alarm sounds.

Security Operations

This chapter presents the following:

- The security operations center (SOC)
- Preventive and detective measures
- Logging and monitoring

There are two types of companies in the world: those that know they've been hacked, and those that don't.
—Misha Glenny

Security operations pertains to everything that takes place to keep networks, computer systems, applications, and environments up and running in a secure and protected manner. But even if you take great care to ensure you are watching your perimeters (both virtual and physical) and ensuring that you provision new services and retire unneeded ones in a secure manner, odds are that some threat source will be able to compromise your information systems. What then? Security operations also involves the detection, containment, eradication, and recovery that is required to ensure the continuity of business operations.

Most of the necessary operational security issues have been addressed in earlier chapters. They were integrated with related topics and not necessarily pointed out as actual operational security issues. So instead of repeating what has already been stated, this chapter reviews and points out the operational security topics that are important for organizations and CISSP candidates.

The Security Operations Center

The security operations center (SOC) is the nerve center of security operations in organizations with a mature information security management system (ISMS). The SOC encompasses the people, processes, and technology that support logging and monitoring of preventive controls, detection of security events, and incident response. By integrating them together in the SOC, an organization streamlines the process of detecting and responding to threats, thereby minimizing organizational losses. In the aftermath of a security incident, lessons learned can be uniformly applied to better mitigate future threats. As defensive processes evolve, they can be rehearsed easily because everyone is on the same team.

Elements of a Mature SOC

Figure 21-1 shows a high-level view of the core elements of a typical mature SOC. More important than the specific components is the fact that they are integrated so that security tasks are performed in a coordinated manner. Still, it's hard to have a SOC that doesn't have at least the three platforms shown in the figure. The endpoint detection and response (EDR) tool is deployed on all endpoints and monitors user and process behaviors. Anything like suspicious activities or suspected malware is reported to a central management system, which is typically the security information and event management (SIEM) platform. Of course, the EDR can't tell what is going on across the networks, so we need a tool to monitor those for suspicious activity. This is the role of the network detection and response (NDR) system, which similarly reports its findings to the SIEM solution. The SIEM solution aggregates these (and many other) data feeds and provides a holistic view into all the security-related information in the organizational environment.

Tier 1 security analysts spend most of their time monitoring security tools and other technology platforms for suspicious activity. For all their sophistication, these tools tend to generate a lot of false positives (that is, false alarms), so we need people to go through and verify the alerts generated by these tools. These analysts are typically the least experienced, so their job is to triage alerts, handling the more mundane and passing on the more complex and dangerous ones to the more experienced staff in the SOC. Tier 2 analysts can dig deeper into the alerts to determine if they constitute security incidents. If they do, these analysts can then coordinate with incident responders and intelligence analysts to further investigate, contain, and eradicate the threats.

The key to a good SOC is to have the policies and procedures in place to ensure the platforms are well tuned, the team is trained and working together, and the context of the organization's business is considered in every action taken. This business context

People

Technology
Endpoint
detection and
response

Tier 1
analyst

Process
Intelligence
analyst

Policies
Procedures

Business

Network
detection and
response

Figure 21-1

Security
information
and event
management

Partners

Tier 2
analyst

Core elements of a mature SOC

Incident
responder

Government

includes partners and customers, because the SOC needs to understand the entire ecosystem within which the organization operates. Occasionally, liaising with appropriate
government organizations will also be needed, and the SOC must be prepared to do so.
Examples of this are scenarios that require reporting cybercrimes and exchanging threat
intelligence with the appropriate agencies.

Threat Intelligence

One of the key capabilities of a SOC is to consume (and, ideally, develop) threat intelligence. Gartner defines threat intelligence as “evidence-based knowledge…about an
existing or emerging menace or hazard to assets. This intelligence can be used to inform

decisions regarding the subject's response to that menace or hazard." In other words,
threat intelligence is information about our adversaries' past, present, and future actions
that allows us to respond or prevent them from being successful. From this definition
flow four essential characteristics of good intelligence, known by the acronym CART:

- Complete Sufficient to detect or prevent the threat from being realized
- Accurate Factual and free from errors
- Relevant Useful to detect and prevent the threat from being realized
- Timely Received and operationalized fast enough to make an impact

It is important to keep in mind that threat intelligence is meant to help decision-makers
choose what to do about a threat. It answers a question that these leaders may have. For
example, a C-suite executive may ask a strategic question like, "What cyberthreats will
be targeting our industry in the next year?" That person probably doesn't care about (or
understand) the technical details of the tools the SOC is using. The SOC director, on
the other hand, is interested in tactical issues and may need to know the technical details
in order to respond to ongoing threats. The SOC director may ask what command
and control infrastructure a particular threat actor is using. So, all good intelligence is,
essentially, an answer to a question asked by a decision-maker in the organization. These
questions are the requirements that drive the intelligence cycle shown in Figure 21-2.

Figure 21-2 The intelligence cycle

Once the requirements are known and prioritized, the intelligence analyst can get to
work collecting data that can help answer those questions. The next section discusses
the different data sources that analysts can use, but for now, the important

point to consider is that intelligence analysts shouldn't have to start from scratch when it comes to identifying the data sources. A good collection management framework (CMF) allows an organization to determine where data lives that can answer the questions that are being asked by its leaders and identify informational "blind spots" that need to be addressed by developing new data sources.

The data that is collected still needs to be analyzed before it yields intelligence products. The analysis step involves integrating the data and evaluating it. Intelligence analysts may reach out to subject matter experts to ensure specific data items are reliable, valid, and relevant, and to help put them into a broader context. Sometimes the data items will contradict each other, so these conflicts need to be resolved before drawing final conclusions.

The final step in the intelligence cycle is to share the finished intelligence with the appropriate decision-makers. Because the intelligence requirement was meant to answer a question from a given individual (or class of individuals), the analyst already knows how to phrase the report. If the report is going to an executive, it should be written in a nontechnical manner (but, ideally, with a more technical appendix that explains where the conclusions come from). If the report is going to cybersecurity professionals, it requires a lot more technical data.

Typically, one full iteration of the intelligence cycle leads to further questions that must be answered. These questions feed the next cycle by becoming (or contributing to) new intelligence requirements.

Threat Data Sources

Let's get back to the threat data sources that are needed to address intelligence requirements. Numerous third parties offer free or paid threat data feeds. These are subscription services that constantly (or periodically) feed information such as indicators of compromise (IOCs); an IOC is technical data that is characteristic of malicious activity. For example, we may know that a particular domain name is being used to deliver ransomware to compromised targets, so that domain name is an IOC for that particular threat.

Unless the question that drove an intelligence requirement was "What is one domain used in ransomware attacks?" this IOC, by itself, would not be an intelligence product. Rather, it is an example of the first of three types of data sources commonly

used in
cyberthreat intelligence: third-party data feeds.
Another important type of data source is called open-source intelligence (OSINT),
which is the name collectively given to any source that is freely available on the Internet. Often, we can get the information we need simply by doing a web search for it. Of
course, there are also tools that make this process much easier by integrating queries
against multiple open sources. Over time, intelligence analysts assemble lists of URLs
that prove useful to their specific intelligence needs.
The third type of commonly used data source, and in many ways the most important,
is internal sources. These are sources under the direct control of the organization and that

can be tasked to collect data. For example, you may task your DNS server to provide all
the domain names for which clients in your organization are requesting resolution. This
would likely be a very large list full of repeated entries, particularly for popular domains.
From it, however, you could gather data such as newly observed domains (NODs) or domains with a small community of interest (COI) in your organization. Either could be
an early indicator of an attack, though it would be fraught with false positives.

Cyberthreat Hunting

EXAM TIP Threat hunting involves proactively searching for malicious activities that were not detected by other means. If you already know there’s been an incident, then you are reactively responding to it. This key difference between threat hunting and incident response is important to remember.

PART VII

If you have a threat intelligence program in your organization, you can use it to stay one
step ahead of the adversaries (or at least just one step behind). Cyberthreat hunting is the
practice of proactively looking for threat actors in your networks. In other words, instead
of waiting for an alert from your SIEM system to start investigating an incident, you
develop a hypothesis of what an adversary may be up to (informed by threat intelligence,
of course) and then set about proving or negating that hypothesis.
For example, suppose that threat intelligence reveals that other organizations in your

sector are being targeted by attackers who are enabling the Remote Desktop Protocol (RDP) to move laterally across the environment. RDP is normally disabled in your organization except for on a handful of jump boxes (hardened hosts that act as a secure entry point or gateway into a sensitive part of a network). From these two facts, you develop the hypothesis that an adversary is enabling RDP on regular workstations to move laterally over your organization's networks. Your hunt operation will be centered on proving your hypothesis (by finding evidence that this is going on) or negating it (by finding no workstations with RDP inappropriately enabled). Your hunt would involve checking the registry of every Windows endpoint in your environment, examining the Windows Registry keys that enable Remote Desktop Services (RDS). Hopefully, you would write a script that does this for you automatedly, so you don't have to manually check every endpoint. Suppose you find several endpoints with RDS enabled. You now narrow your hunt to those systems and determine whether they are a) legitimately authorized to use RDS, b) authorized for RDS but didn't follow the configuration management process, or c) evidence of adversarial activities.

This is the crux of threat hunting: you develop a hypothesis of adversarial action based on threat intelligence, and then you prove or negate your hypothesis. Threat hunting is inherently proactive and based on intelligence, whereas incident response is reactive and based on alerts. Because threat hunting requires the skills of intelligence analysts, cybersecurity analysts (typically tier 2), and incident responders, many organizations stand up hunt teams with one or more members from each of these three roles. The team may run a hunt campaign consisting of multiple related hunt operations, and then return to their daily jobs until they're needed for the next campaign.

Preventive and Detective Measures

As exciting and effective as cyberthreat hunting can be, relatively few organizations have the resources to engage in this effort consistently. Even in organizations that do have the resources, most of the efforts of security operations are focused on preventing and detecting security incidents. A good way to reduce the likelihood of contingencies and

disasters is to ensure that your organization’s defensive architectures include the right set of tools. These technical controls need to be carefully considered in the context of your organization’s own conditions to determine which are useful and which aren’t. Regardless of the tools you employ, there is an underlying process that drives their operation in a live environment. The steps of this generalized process are described here:

1. Understand the risk. Chapter 2 presented the risk management process that organizations should use. The premise of this process is that you can’t ever eliminate all risks and should therefore devote your scarce resources to mitigating the most dangerous risks to a point where their likelihood is acceptable to the senior leaders. If you don’t focus on that set of risks, you will likely squander your resources countering threats that are not the ones your CEO is really concerned about.
2. Use the right controls. Once you are focused on the right set of risks, you can more easily identify the controls that will appropriately mitigate them. The relationship between risks and controls is many to many, since a given risk can have multiple controls assigned to it and a given control can be used to mitigate multiple risks. In fact, the number of risks mitigated by one control should give you an indicator of the value of that control to the organization. On the other hand, having multiple controls mitigating a risk may be less efficient, but may provide resiliency.
3. Use the controls correctly. Selecting the right tools is only part of the battle. You also need to ensure they are emplaced and configured correctly. The network architectures covered in Chapter 7 place some very significant limitations on the effectiveness of tools based on where they are plugged in. If an IDS is deployed on the wrong subnet, it may not be able to monitor all the traffic from the threat sources against which it is supposed to defend. Similarly, that same IDS with the wrong configuration or rule set could well become an expensive ornament on the network.
4. Manage your configuration. One of the certainties in life is that, left alone, every configuration is guaranteed to become obsolete at some point in the future. Even if it is not left alone, making unauthorized or undocumented changes will introduce risk at best and at worst quietly render your network vulnerable to an immediate threat. Properly done, configuration management will ensure you have ground truth about your network so that you can better answer the questions that are typically asked when doing security operations.

5. Assess your operation. You should constantly (or at least periodically) be

looking
at your defensive plan, comparing it with your latest threat and risk assessments,
and asking yourself, “Are we still properly mitigating the risks?” You should test
your controls using cases derived from your risk assessment. This verifies that you
are correctly mitigating those risks. However, you should also occasionally test
your controls against an unconstrained set of threats in order to validate that you
are mitigating the correct risks. A good penetration test (pen test) can both verify
and validate the controls.

This process can yield a huge number of possible preventive controls. There are some
controls, however, that are so pervasive that every information security professional
should be able to incorporate them into a defensive architecture. In the
following sections, we describe the most important ones.

Firewalls

PART VII

Firewalls are used to restrict access to one network from another network. Most
organizations use firewalls to restrict access to their networks from the Internet. They may also
use firewalls to restrict one internal network segment from accessing another internal
segment. For example, if the security administrator wants to make sure unauthorized
employees cannot access the research and development network, he would place a
firewall between the R&D network and all other networks and configure the firewall to
allow only the type of traffic he deems acceptable.
A firewall device supports and enforces the organization’s network security policy.
An organizational security policy provides high-level directives on acceptable and
unacceptable actions as they pertain to protecting critical assets. The firewall has a more
defined and granular security policy that dictates what services are allowed to be accessed,
what IP addresses and ranges are to be restricted, and what ports can be accessed. The
firewall is described as a “choke point” in the network because all communications should
flow through it, and this is where traffic is inspected and restricted.
A firewall may be a server running a firewall software product or a specialized
hardware appliance. In either case, the firewall monitors packets coming into and out of
the network it is protecting. It can discard packets, repackage them, or redirect them,

depending upon the firewall configuration. Packets are filtered based on their source
and destination addresses, and ports by service, packet type, protocol type, header information, sequence bits, and much more. Many times, organizations set up firewalls to
construct a demilitarized zone (DMZ), which is a network segment located between the
protected and unprotected networks. The DMZ provides a buffer zone between the dangerous Internet and the goodies within the internal network that the organization is
trying to protect. As shown in Figure 21-3, two firewalls are usually installed to form the
DMZ. The DMZ usually contains web, mail, and DNS servers, which must be hardened systems because they would be the first in line for attacks. Many DMZs also have an IDS
sensor that listens for malicious and suspicious behavior.

DMZ

Public
servers

Internal LAN

Enterprise
servers

Internet
Work
stations

Figure 21-3

At least two firewalls, or firewall interfaces, are generally used to construct a DMZ.

Many different types of firewalls are available, because each environment may have
unique requirements and security goals. Firewalls have gone through an evolution of
their own and have grown in sophistication and functionality. The following sections
describe the various types of firewalls.
The types of firewalls we will review are

- Packet filtering
- Stateful
- Proxy
- Next-generation

We will then dive into the three main firewall architectures, which are

• Screened host
• Multihome
• Screened subnet

NOTE Recall that we discussed another type of firewall, web application firewalls (WAFs), in Chapter 4.

Packet-Filtering Firewalls

Packet filtering is a firewall technology that makes access decisions based upon networklevel protocol header values. The device that is carrying out packet-filtering processes is configured with access control lists (ACLs), which dictate the type of traffic that is allowed into and out of specific networks.

Packet filtering was the technology used in the first generation of firewalls, and it is the most rudimentary type of all of the firewall technologies. The filters only have the capability of reviewing protocol header information at the network and transport layers and carrying out permit or deny actions on individual packets. This means the filters can make access decisions based upon the following basic criteria:

• Source and destination IP addresses
• Source and destination port numbers
• Protocol types
• Inbound and outbound traffic direction

Packet filtering is built into a majority of the firewall products today and is a capability that many routers perform. The ACL filtering rules are enforced at the network interface of the device, which is the doorway into or out of a network. As an analogy, you could have a list of items you look for before allowing someone into your office premises through your front door. Your list can indicate that a person must be 18 years or older, have an access badge, and be wearing shoes. When someone knocks on the door, you grab your list, which you will use to decide if this person can or cannot come inside. So your front door is one interface into your office premises. You can also have a list that outlines who can exit your office premises through your back door, which is another interface. As shown in Figure 21-4, a router has individual interfaces with their own unique addresses, which provide doorways into and out of a network. Each interface can have its own ACL values, which indicate what type of traffic is allowed in and out of that

specific interface.
Figure 21-4
ACLs are
enforced at
the network
interface level.

Router D
10.10.13.2
f0/0
10.10.13.1
f2/0

Router C
f1/0
10.10.12.2

10.10.10.1
f0/0

10.10.12.1
f0/0
f1/0
10.10.11.1

Router A

f1/0
10.10.11.2
Router B

PART VII

f0/0
10.10.10.2

We will cover some basic ACL rules to illustrate how packet filtering is implemented
and enforced. The following router configuration allows SMTP traffic to travel from
system 10.1.1.2 to system 172.16.1.1:
permit tcp host 10.1.1.2 host 172.16.1.1 eq smtp

This next rule permits UDP traffic from system 10.1.1.2 to 172.16.1.1:
permit udp host 10.1.1.2 host 172.16.1.1

If you want to ensure that no ICMP traffic enters through a certain interface, the
following ACL can be configured and deployed:
deny icmp any any

If you want to allow standard web traffic (that is, to a web server listening on port 80)
from system 1.1.1.1 to system 5.5.5.5, you can use the following ACL:
permit tcp host 1.1.1.1 host 5.5.5.5 eq www

NOTE Filtering inbound traffic is known as ingress filtering. Outgoing traffic can also be filtered using a process referred to as egress filtering.

So when a packet arrives at a packet-filtering device, the device starts at the top of
its ACL and compares the packet's characteristics to each rule set. If a successful match
(permit or deny) is found, then the remaining rules are not processed. If no matches are
found when the device reaches the end of the list, the traffic should be denied, but each
product is different. So if you are configuring a packet-filtering device, make sure that if
no matches are identified, then the traffic is denied.
Packet filtering is also known as stateless inspection because the device does not understand the context that the packets are working within. This means that the device does
not have the capability to understand the "full picture" of the communication that is
taking place between two systems, but can only focus on individual packet characteristics. As we will see in the next section, stateful firewalls understand and keep track of
a full communication session, not just the individual packets that make it up. Stateless
firewalls make their decisions for each packet based solely on the data contained in that
individual packet.
The lack of sophistication in packet filtering means that an organization should not
solely depend upon this type of firewall to protect its infrastructure and assets, but it does
not mean that this technology should not be used at all. Packet filtering is commonly
carried out at the edge of a network to strip out all of the obvious "junk" traffic. Since the
rules are simple and only header information is analyzed, this type of filtering can take
place quickly and efficiently. After traffic is passed through a packet-filtering device, it is
usually then processed by a more sophisticated firewall, which digs deeper into the packet
contents and can identify application-based attacks.

Some of the weaknesses of packet-filtering firewalls are as follows:

• They cannot prevent attacks that employ application-specific vulnerabilities

or
functions.
• They have limited logging functionality.
• Most packet-filtering firewalls do not support advanced user authentication schemes.
• They may not be able to detect packet fragmentation attacks.

The advantages to using packet-filtering firewalls are that they are scalable, they are not application dependent, and they have high performance because they do not carry out extensive processing on the packets. They are commonly used as the first line of defense to strip out all the network traffic that is obviously malicious or unintended for a specific network. The network traffic usually then has to be processed by more sophisticated firewalls that will identify the not-so-obvious security risks.

Stateful Firewalls

PART VII

When packet filtering is used, a packet arrives at the firewall, and the firewall runs through its ACLs to determine whether this packet should be allowed or denied. If the packet is allowed, it is passed on to the destination host, or to another network device, and the packet-filtering device forgets about the packet. This is different from stateful inspection, which remembers and keeps track of what packets went where until each particular connection is closed.

A stateful firewall is like a nosy neighbor who gets into people's business and conversations. She keeps track of the suspicious cars that come into the neighborhood, who is out of town for the week, and the postman who stays a little too long at the neighbor lady's house. This can be annoying until your house is burglarized. Then you and the police will want to talk to the nosy neighbor, because she knows everything going on in the neighborhood and would be the one most likely to know something unusual happened. A stateful-inspection firewall is nosier than a regular filtering device because it keeps track of what computers say to each other. This requires that the firewall maintain a state table, which is like a score sheet of who said what to whom.

Keeping track of the state of a protocol connection requires keeping track of many variables. Most people understand the three-step handshake a TCP connection goes through (SYN, SYN/ACK, ACK), but what does this really mean? If Quincy's system wants to communicate with your system using TCP, it sends your system a packet

with
the SYN flag value in the TCP header set to 1. This makes this packet a SYN packet.
If your system accepts Quincy's system's connection request, it sends back a packet that
has both the SYN and ACK flags within the packet header set to 1. This is a SYN/ACK
packet. Finally, Quincy's system confirms your system's SYN with its own ACK packet.
After this three-way handshake, the TCP connection is established.
While many people know about these three steps of setting up a TCP connection, they are not always familiar with all of the other items that are being negotiated at this
time. For example, your system and Quincy's system will agree upon sequence numbers,

Byte
Offset
0

1

0

2

3

Source Port

Destination Port

4

Sequence Number

8

Acknowledgment Number

12
16

Offset

Window

CEUA PRSF

Urgent Pointer

Checksum

20

Bit

TCP Flags

Offset Reserved

20
Bytes

TCP Options (variable length, optional)
1
2
3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 89 0 1
Nibble

Byte

Word

TCP Flags

Congestion Notification

CEUAPRSF
Congestion Window
C 0x80 Reduced (CWR)
E 0x40 ECN Echo (ECE)
U 0x20 Urgent
A 0x10 Ack
P 0x08 Push
R 0x04 Reset
S 0x02 Syn
F 0x01 Fin

ECN (Explicit Congestion
Notification). See RFC
3168 for full details, valid
states below.

TCP Options

0 End of Options List
1 No Operation (NOP, Pad)
2 Maximum segment size
3 Window Scale
Packet State DSB ECN bits 4 Selective ACK ok
11
Syn 00
8 Timestamp

Syn-Ack
Ack

00
01

01
00

No Congestion
No Congestion

01
10

00
00

Congestion
Receiver Response
Sender Response

11
11
11

00
01
11

Offset

Number of 32-bit words
in TCP header, minimum
value of 5. Multiply by 4
to get byte count.

Checksum

Checksum of entire TCP
segment and pseudo
header (parts of IP
header)

Figure 21-5 TCP header

how much data to send at a time (window size), how potential transmission errors will be identified (CRC values), and so forth. Figure 21-5 shows all of the values that make up a TCP header. So, a lot of information is going back and forth between your systems just in this one protocol—TCP. There are other protocols that are involved with networking

that a stateful firewall has to be aware of and keep track of.
So “keeping state of a connection” means to keep a scorecard of all the various protocol
header values as packets go back and forth between systems. The values not only have to
be correct—they have to happen in the right sequence. For example, if a stateful firewall
receives a packet that has all TCP flag values turned to 1, something malicious is taking
place. Under no circumstances during a legitimate TCP connection should all of these
values be turned on like this. Attackers send packets with all of these values turned to
1 with the hopes that the firewall does not understand or check these values and just
forwards the packets onto the target system.

In another situation, if Gwen’s system sends your system a SYN/ACK packet and
your system did not first send a SYN packet to Gwen’s system, this, too, is against the
protocol rules. The protocol communication steps have to follow the proper sequence.
Attackers send SYN/ACK packets to target systems in an attempt to get the firewall to
interpret this as an already established connection and just allow the packets to go to
the destination system without inspection. A stateful firewall will not be fooled by such
actions because it keeps track of each step of the communication. It knows how protocols
are supposed to work, and if something is out of order (incorrect flag values, incorrect
sequence, etc.), it does not allow the traffic to pass through.
When a connection begins between two systems, the firewall investigates all elements
of the packet (all headers, payload, and trailers). All of the necessary information about
the specific connection is stored in the state table (source and destination IP addresses,
source and destination ports, protocol type, header flags, sequence numbers, timestamps,
etc.). Once the initial packets go through this in-depth inspection and everything is
deemed safe, the firewall then just reviews the network and transport header portions
for the rest of the session. The values of each header for each packet are compared to
the values in the current state table, and the table is updated to reflect the

progression of
the communication process. Scaling down the inspection of the full packet to just the
headers for each packet is done to increase performance.
TCP is considered a connection-oriented protocol, and the various steps and states this
protocol operates within are very well defined. A connection progresses through a series
of states during its lifetime. The states are LISTEN, SYN-SENT, SYN-RECEIVED,
ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LASTACK, TIME-WAIT,
and the fictional state CLOSED. A stateful firewall keeps track of
each of these states for each packet that passes through, along with the corresponding
acknowledgment and sequence numbers. If the acknowledgment and/or sequence
numbers are out of order, this could imply that a replay attack is underway, and the
firewall will protect the internal systems from this activity.
Nothing is ever simple in life, including the standardization of network protocol
communication. While the previous statements are true pertaining to the states of a TCP
connection, in some situations an application layer protocol has to change these basic
steps. For example, FTP uses an unusual communication exchange when initializing its
data channel compared to all of the other application layer protocols. FTP basically sets
up two sessions just for one communication exchange between two computers. The states
of the two individual TCP connections that make up an FTP session can be tracked in
the normal fashion, but the state of the FTP connection follows different rules. For a
stateful device to be able to properly monitor the traffic of an FTP session, it must be
able to take into account the way that FTP uses one outbound connection for the control
channel and one inbound connection for the data channel. If you were configuring a
stateful firewall, you would need to understand the particulars of some specific protocols
to ensure that each is being properly inspected and controlled.
Since TCP is a connection-oriented protocol, it has clearly defined states during the
connection establishment, maintenance, and tearing-down stages. UDP is a connectionless
protocol, which means that none of these steps take place. UDP holds no state, which
makes it harder for a stateful firewall to keep track of. For connectionless protocols,

Stateful-Inspection Firewall Characteristics
The following lists some important characteristics of a stateful-inspection firewall:

- Maintains a state table that tracks each and every communication session
- Provides a high degree of security and does not introduce the performance hit that application proxy firewalls introduce
- Is scalable and transparent to users
- Provides data for tracking connectionless protocols such as UDP and ICMP
- Stores and updates the state and context of the data within the packets

a stateful firewall keeps track of source and destination addresses, UDP header values, and some ACL rules. This connection information is also stored in the state table and tracked.
Since the protocol does not have a specific tear-down stage, the firewall will just time out the connection after a period of inactivity and remove the data being kept pertaining to that connection from the state table.
An interesting complexity of stateful firewalls and UDP connections is how ICMP comes into play. Since UDP is connectionless, it does not provide a mechanism to allow the receiving computer to tell the sending computer that data is coming too fast. In TCP, the receiving computer can alter the Window value in its header, which tells the sending computer to reduce the amount of data that is being sent. The message is basically, “You are overwhelming me and I cannot process the amount of data you are sending me. Slow down.” UDP does not have a Window value in its header, so instead the receiving computer sends an ICMP packet that provides the same function. But now this means that the stateful firewall must keep track of and allow associated ICMP packets with specific UDP connections. If the firewall does not allow the ICMP packets to get to the sending system, the receiving system could get overwhelmed and crash. This is just one example of the complexity that comes into play when a firewall has to do more than just packet filtering. Although stateful inspection provides an extra step of protection, it also adds more complexity because this device must now keep a dynamic state table and remember connections.
Stateful-inspection firewalls, unfortunately, have been the victims of many types of DoS attacks. Several types of attacks are aimed at flooding the state table with bogus information. The state table is a resource, similar to a system’s hard drive space, memory, and CPU. When the state table is stuffed full of bogus information, a poorly

designed
device may either freeze or reboot.

Proxy Firewalls
A proxy is a middleman. It intercepts and inspects messages before delivering them to the
intended recipients. Suppose you need to give a box and a message to the president of
the United States. You couldn’t just walk up to the president and hand over these items.

Instead, you would have to go through a middleman, likely a Secret Service agent, who
would accept the box and message and thoroughly inspect the box to ensure nothing
dangerous is inside. This is what a proxy firewall does—it accepts messages either entering or leaving a network, inspects them for malicious information, and, when it decides
the messages are okay, passes the data on to the destination computer.
A proxy firewall stands between a trusted network and an untrusted network and makes the connection, each way, on behalf of the source. What is important is that a
proxy firewall breaks the communication channel; there is no direct connection between
the two communicating devices. Where a packet-filtering device just monitors traffic
as it is traversing a network connection, a proxy ends the communication session and
restarts it on behalf of the sending system. Figure 21-6 illustrates the steps of a proxybased firewall. Notice that the firewall does not simply apply ACL rules to the traffic; it
stops the user connection at the internal interface of the firewall itself and then starts a
new session on behalf of this user on the external interface. When the external web server
replies to the request, this reply goes to the external interface of the proxy firewall and
ends. The proxy firewall examines the reply information and, if it is deemed safe, starts
a new session from itself to the internal system. This is just like our analogy of what the
Secret Service agent does between you and the president.
A proxy technology can actually work at different layers of a network stack. A proxybased firewall that works at the lower layers of the OSI model is referred to as a circuitlevel proxy. A proxy-based firewall that works at the application layer is, strangely enough,
called an application-level proxy.

User with web
browser configured
Once page is

Firewall\HTTP
to use firewall proxy
cached, proxy server
proxy requests
server requests web page
sends the user the
web page on
from Internet website.
behalf of end user. requested web page.

3

5

1

Internet

Workstation

Web server
2

Web server responds to
HTTP request from proxy
server, unaware the request
is coming from a user behind a proxy.

Firewall\HTTP proxy
accepts connection request.

Figure 21-6

Proxy firewall breaks connection

A circuit-level proxy creates a connection (circuit) between the two communicating systems. It works at the session layer of the OSI model and monitors traffic from a network-based view. This type of proxy cannot “look into” the contents of a packet; thus, it does not carry out deep-packet inspection. It can only make access decisions based upon protocol header and session information that is available to it. While this means that a circuit-level proxy cannot provide as much protection as an applicationlevel proxy, because it does not have to understand application layer

protocols, it is
considered application independent. So, it cannot provide the detail-oriented protection
that a proxy working at a higher level can, but this allows it to provide a broader range of
protection where application layer proxies may not be appropriate or available.
NOTE Traffic sent to the receiving computer through a circuit-level proxy appears to have originated from the firewall instead of the sending system. This is useful for hiding information about the internal computers on the network the firewall is protecting.

Application-level proxies inspect the packet up through the application layer. Where
a circuit-level proxy only has insight up to the session layer, an application-level proxy
understands the packet as a whole and can make access decisions based on the content of
the packets. Application-level proxies understand various services and protocols and the
commands that are used by them. An application-level proxy can distinguish between
an FTP GET command and an FTP PUT command, for example, and make access
decisions based on this granular level of information; on the other hand, packet-filtering
firewalls and circuit-level proxies can allow or deny FTP requests only as a whole, not by
the commands used within FTP.
An application-level proxy firewall has one proxy per protocol. A computer can have
many types of protocols (FTP, NTP, SMTP, HTTP, and so on). Thus, one applicationlevel proxy per protocol is required. This does not mean one proxy firewall per service is
required, but rather that one portion of the firewall product is dedicated to understanding
how a specific protocol works and how to properly filter it for suspicious data.
Providing application-level proxy protection can be a tricky undertaking. The proxy
must totally understand how specific protocols work and what commands within that
protocol are legitimate. This is a lot to know and look at during the transmission of
data. As an analogy, picture a screening station at an airport that is made up of many
employees, all with the job of interviewing people before they are allowed into the airport
and onto an airplane. These employees have been trained to ask specific questions and
detect suspicious answers and activities, and have the skill set and authority to detain
suspicious individuals. Now, suppose each of these employees speaks a different language
because the people they interview come from different parts of the world. So, one
employee who speaks German could not understand and identify suspicious answers

of
a person from Italy because they do not speak the same language. This is the same for an
application-level proxy firewall. Each proxy is a piece of software that has been designed
to understand how a specific protocol “talks” and how to identify suspicious data within
a transmission using that protocol.

NOTE If the application-level proxy firewall does not understand a certain protocol or service, it cannot protect this type of communication. In this scenario, a circuit-level proxy is useful because it does not deal with such complex issues. An advantage of a circuit-level proxy is that it can handle a wider variety of protocols and services than an application-level proxy can, but the downfall is that the circuit-level proxy cannot provide the degree of granular control that an application-level proxy provides. Life is just full of compromises.

A circuit-level proxy works similarly to a packet filter in that it makes access decisions
based on address, port, and protocol type header values. It looks at the data within the
packet header rather than the data at the application layer of the packet. It does not know
whether the contents within the packet are safe or unsafe; it only understands the traffic
from a network-based view.
An application-level proxy, on the other hand, is dedicated to a particular protocol
or service. At least one proxy is used per protocol because one proxy could not properly
interpret all the commands of all the protocols coming its way. A circuit-level proxy
works at a lower layer of the OSI model and does not require one proxy per protocol
because it does not look at such detailed information.

Application-Level Proxy Firewalls
Application-level proxy firewalls, like all technologies, have their pros and cons.
It is important to fully understand all characteristics of this type of firewall before
purchasing and deploying this type of solution.
Characteristics of application-level proxy firewalls:

• They have extensive logging capabilities due to the firewall being able to examine the entire network packet rather than just the network addresses and ports.
• They are capable of authenticating users directly, as opposed to packetfiltering firewalls and stateful-inspection firewalls, which can usually only

carry out system authentication.
• Since they are not simply layer 3 devices, they can address spoofing attacks and other sophisticated attacks.

• They are not generally well suited to high-bandwidth or real-time applications.
• They tend to be limited in terms of support for new network applications and protocols.
• They create performance issues because of the necessary per-packet processing requirements.

PART VII

Disadvantages of using application-level proxy firewalls:

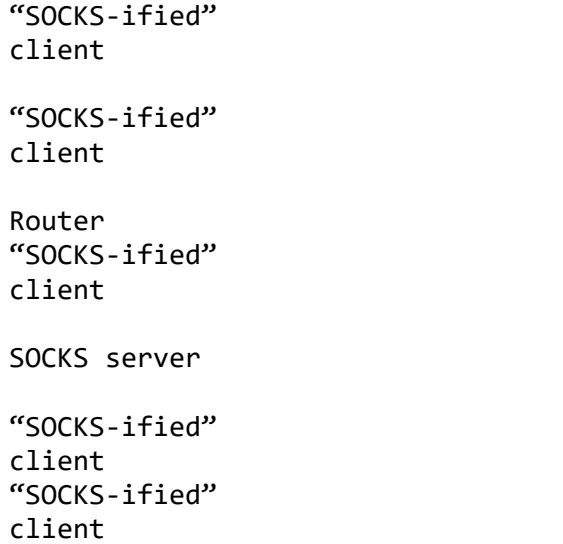

Figure 21-7

Circuit-level proxy firewall

SOCKS is an example of a circuit-level proxy gateway that provides a secure channel between two computers. When a SOCKS-enabled client sends a request to access a computer on the Internet, this request actually goes to the network’s SOCKS proxy firewall, as shown in Figure 21-7, which inspects the packets for malicious information and checks its policy rules to see whether this type of connection is allowed. If the packet is acceptable and this type of connection is allowed, the SOCKS firewall sends the message to the destination computer on the Internet. When the computer on the Internet responds, it sends its packets to the SOCKS firewall, which again inspects the

data and then passes the packets on to the client computer.
The SOCKS firewall can screen, filter, audit, log, and control data flowing in and out
of a protected network. Because of its popularity, many applications and protocols have
been configured to work with SOCKS in a manner that takes less configuration on the
administrator’s part, and various firewall products have integrated SOCKS software to
provide circuit-based protection.
NOTE Remember that whether an application- or circuit-level proxy firewall is being used, it is still acting as a proxy. Both types of proxy firewalls deny actual end-to-end connectivity between the source and destination systems. In attempting a remote connection, the client connects to and communicates with the proxy; the proxy, in turn, establishes a connection to the destination system and makes requests to it on the client’s behalf. The proxy maintains two independent connections for every one network transmission. It essentially turns a two-party session into a four-party session, with the middle process emulating the two real systems.

Application-Level vs. Circuit-Level Proxy Firewall Characteristics
Characteristics of application-level proxy firewalls:

- Each protocol that is to be monitored must have a unique proxy.
- They provide more protection than circuit-level proxy firewalls.
- They require more processing per packet and thus are slower than circuitlevel proxy firewalls.

Characteristics of circuit-level proxy firewalls:

- They do not require a proxy for each and every protocol.
- They do not provide the deep-inspection capabilities of an application-level proxy firewall.
- They provide security for a wider range of protocols.

Next-Generation Firewalls

NOTE Firewall technology has evolved as attack types have evolved. The first-generation firewalls could only monitor network traffic. As attackers moved from just carrying out network-based attacks (DoS, fragmentation, spoofing, etc.) to conducting software-based attacks (buffer overflows, injections, malware, etc.), new generations of firewalls were developed to monitor for these types of attacks.

A next-generation firewall (NGFW) combines the best attributes of the previously discussed firewalls, but adds a number of important improvements. Most importantly, it
incorporates a signature-based and/or behavioral analysis IPS engine. This means that,
in addition to ensuring that the traffic is behaving in accordance with the

rules of the
applicable protocols, the firewall can look for specific indicators of attack
even in otherwise well-behaved traffic. Some of the most advanced NGFWs include
features that
allow them to share signatures with a cloud-based aggregator so that once a new
attack is
detected by one firewall, all other firewalls manufactured by that vendor become
aware
of the attack signature.
Another characteristic of an NGFW is its ability to connect to external data
sources
such as Active Directory, whitelists, blacklists, and policy servers. This
feature allows
controls to be defined in one place and pulled by every NGFW on the network,
which
reduces the chances of inconsistent settings on the various firewalls that
typically exist in
large networks.
For all their power, NGFWs are not appropriate for every organization. The
typical
cost of ownership alone tends to make these infeasible for small or even
medium-sized
networks. Organizations need to ensure that the correct firewall technology is
in place to
monitor specific network traffic types and protect unique resource types. The
firewalls
also have to be properly placed; we will cover this topic in the next section.

Firewall Type	OSI Layer	Characteristics
Packet filtering	Network layer	Looks at destination and source addresses, ports, and services requested. Typically routers using ACLs to control and monitor network traffic.
Stateful	Network layer	Looks at the state and context of packets. Keeps track of each conversation using a state table.
Application-level proxy		

Application layer	Looks deep into packets and makes granular access control decisions. It requires one proxy per protocol.
Circuit-level proxy	Session layer	Looks only at the header packet information. It protects a wider range of protocols and services than an application-level proxy, but does not provide the detailed level of control available to an application-level proxy.
Next-generation firewall	Multiple layers	Very fast and supportive of high bandwidth. Built-in IPS. Able to connect to external services like Active Directory.

Table 21-1

Comparison of Different Types of Firewalls

Table 21-1 lists the important concepts and characteristics of the firewall types discussed in the preceding sections. Although various firewall products can provide a mix of these services and work at different layers of the OSI model, it is important you understand the basic definitions and functionalities of these firewall types.

Appliances

A firewall may take the form of either software installed on a regular computer using a regular operating system or a dedicated hardware appliance that has its own operating system. The second choice is usually more secure, because the vendor uses a stripped-down version of an operating system (usually Linux or BSD Unix). Operating systems are full of code and functionality that are not necessary for a firewall. This extra complexity opens the doors for vulnerabilities. If a hacker can exploit and bring down a company's firewall, then the company is very exposed and in danger. In today's jargon, dedicated hardware devices that have stripped-down operating systems and limited and focused software capabilities are called appliances. Where an operating system has to provide a vast array of functionality, an appliance

provides
very focused functionality—as in just being a firewall.
If a software-based firewall is going to run on a regular system, then the unnecessary user accounts should be disabled, unnecessary services deactivated, unused subsystems disabled, unneeded ports closed, and so on. If firewall software
is going to run on a regular system and not a dedicated appliance, then the system
needs to be fully locked down.

Firewall Architecture
Firewalls can be placed in a number of areas on a network to meet particular needs. They
can protect an internal network from an external network and act as a choke point for
all traffic. A firewall can be used to segment and partition network sections and enforce
access controls between two or more subnets. Firewalls can also be used to provide a
DMZ architecture. And as covered in the previous section, the right firewall type needs
to be placed in the right location. Organizations have common needs for firewalls; hence,
they keep them in similar places on their networks. We will see more on this topic in the
following sections.

Screened Host A screened host is a firewall that communicates directly with a perimeter
router and the internal network. Figure 21-8 shows this type of architecture.
Traffic received from the Internet is first filtered via packet filtering on the outer router.
The traffic that makes it past this phase is sent to the screened-host firewall, which applies

Dual-Homed Firewall Dual-homed refers to a device that has two interfaces: one connected to one network and the other connected to a different network. If firewall
software is installed on a dual-homed device—and it usually is—the underlying operating
system should have packet forwarding and routing turned off for security reasons. If they
are enabled, the computer may not apply the necessary ACLs, rules, or other restrictions
required of a firewall. When a packet comes to the external NIC from an untrusted
network on a dual-homed firewall and the operating system has forwarding enabled, the
operating system forwards the traffic instead of passing it up to the firewall

software for
inspection.
Many network devices today are multihomed, which just means they have several NICs
that are used to connect several different networks. Multihomed devices are commonly
used to house firewall software, since the job of a firewall is to control the traffic as it goes
from one network to another. A common multihomed firewall architecture allows an
organization to have several DMZs. One DMZ may hold devices that are shared between
organizations in an extranet, another DMZ may house the organization's DNS and mail
servers, and yet another DMZ may hold the organization's web servers. Different DMZs
are used for two reasons: to control the different traffic types (for example, to ensure
HTTP traffic only goes toward the web servers and ensure DNS requests go toward the
DNS server), and to ensure that if one system on one DMZ is compromised, the other
systems in the rest of the DMZs are not accessible to this attacker.
If a company depends solely upon a multihomed firewall with no redundancy, this
system could prove to be a single point of failure. If it goes down, then all traffic flow
stops. Some firewall products have embedded redundancy or fault-tolerance capabilities.
If a company uses a firewall product that does not have these capabilities, then the
network should have redundancy built into it.
Along with potentially being a single point of failure, another security issue that is posed
by relying on a single firewall is the lack of defense in depth. If the company depends on
just one firewall, no matter what architecture is being used or how many interfaces the
device has, there is only one layer of protection. If an attacker can compromise the one
firewall, then she can gain direct access to company network resources.

Internal network

Screening
device

Figure 21-8

Screened
host

A screened host is a firewall that is screened by a router.

more rules to the traffic and drops the denied packets. Then the traffic moves to the internal destination hosts. The screened host (the firewall) is the only device that receives traffic directly from the router. No traffic goes directly from the Internet, through the router, and to the internal network. The screened host is always part of this equation.

If the firewall is an application-based system, protection is provided at the network layer by the router through packet filtering, and at the application layer by the firewall. This arrangement offers a high degree of security, because for an attacker to be successful, she would have to compromise two systems.

What does the word “screening” mean in this context? As shown in Figure 21-8, the router is a screening device and the firewall is the screened host. This just means there is a layer that scans the traffic and gets rid of a lot of the “junk” before the traffic is directed toward the firewall. A screened host is different from a screened subnet, which is described next.

Screened Subnet A screened-subnet architecture adds another layer of security to the screened-host architecture. The external firewall screens the traffic entering the DMZ network. However, instead of the firewall then redirecting the traffic to the internal network, an interior firewall also filters the traffic. The use of these two physical firewalls creates a DMZ.

In an environment with only a screened host, if an attacker successfully breaks through the firewall, nothing lies in her way to prevent her from having full access to the internal network. In an environment using a screened subnet, the attacker would have to hack through another firewall to gain access. In this layered approach to security, the more layers provided, the better the protection. Figure 21-9 shows a simple example of a screened subnet.

The examples shown in the figures are simple in nature. Often, more complex networks and DMZs are implemented in real-world systems. Figures 21-10 and 21-11 show some other possible architectures of screened subnets and their configurations.

Internal network

Screened subnet

Screening
device

Firewall

DNS
server
Firewall
Mail
server

Figure 21-9 With a screened subnet, two firewalls are used to create a DMZ.

Exterior
router

Perimeter network 1

Firewall 1

Firewall 2
Perimeter network 2

Interior
router

Figure 21-10 A screened subnet can have different networks within it and different firewalls that
filter for specific threats.

PART VII

Internal netwrok

Border
router 1

Backup

Border
router 2

Firewall

Perimeter netwrok

Web
server

Mail 1

DNS

Perimeter network

Perimeter network
Interior
router 1

Interior
router 2
Internal network

Figure 21-11 Some architectures have separate screened subnets with different server types
in each.

The screened-subnet approach provides more protection than a stand-alone firewall or
a screened-host firewall because three devices are working together and an attacker must
compromise all three devices to gain access to the internal network. This architecture also
sets up a DMZ between the two firewalls, which functions as a small network isolated
among the trusted internal and untrusted external networks. The internal users usually

Firewall Architecture Characteristics
It is important to understand the following characteristics of these firewall architecture
types:
Dual-homed:

- A single computer with separate NICs connected to each network.
- Used to divide an internal trusted network from an external untrusted network.
- Must disable a computer's forwarding and routing functionality so the two networks are truly segregated.

Screened host:

- A router filters (screens) traffic before it is passed to the firewall.

Screened subnet:

- An external router filters (screens) traffic before it enters the subnet. Traffic
headed toward the internal network then goes through two firewalls.

have limited access to the servers within this area. Web, e-mail, and other public servers

often are placed within the DMZ. Although this solution provides the highest security,
it also is the most complex. Configuration and maintenance can prove to be difficult
in this setup, and when new services need to be added, three systems may need to be
reconfigured instead of just one.
TIP Sometimes a screened-host architecture is referred to as a singletiered configuration and a screened subnet is referred to as a two-tiered configuration. If three firewalls create two separate DMZs, this may be called a three-tiered configuration.

PART VII

Organizations used to deploy a piece of hardware for every network function needed
(DNS, mail, routers, switches, storage, web), but today many of these items run within
virtual machines on a smaller number of hardware machines. This reduces software and
hardware costs and allows for more centralized administration, but these components
still need to be protected from each other and external malicious entities. As an analogy,
let’s say that 15 years ago each person lived in their own house and a police officer
was placed between each house so that the people in the houses could not attack each
other. Then last year, many of these people moved in together so that now at least five

people live in the same physical house. These people still need to be protected from each
other, so some of the police officers had to be moved inside the houses to enforce the
laws and keep the peace. Analogously, virtual firewalls have “moved into” the virtualized
environments to provide the necessary protection between virtualized entities.
As illustrated in Figure 21-12, a network can have a traditional physical firewall on the
physical network and virtual firewalls within the individual virtual environments.
Virtual firewalls can provide bridge-type functionality in which individual traffic
links are monitored between virtual machines, or they can be integrated within the
hypervisor. The hypervisor is the software component that carries out virtual machine
management and oversees guest system software execution. If the firewall is embedded
within the hypervisor, then it can “see” and monitor all the activities taking

place within
the system.

Web servers

VLAN1

VLAN2

vNIC

Applications

VLAN1

Database

VLAN2

vNIC

Hypervisor 1

VLAN1

VLAN2

vNIC

Hypervisor 2

Hypervisor 3

Access
Switch
Aggregation
Switch

Firewall
Border router

Figure 21-12 Virtual firewalls

Bastion Host

A system is considered a bastion host if it is a highly exposed device that is most likely to be targeted by attackers. The closer any system is to an untrusted network, such as the Internet, the more it is considered a target candidate since it has a smaller number of layers of protection guarding it. If a system is on the public side of

a
DMZ or is directly connected to an untrusted network, it is considered a bastion host; thus, it needs to be extremely locked down.
The system should have all unnecessary services disabled, unnecessary accounts disabled, unneeded ports closed, unused applications removed, unused subsystems and administrative tools removed, and so on. The attack surface of the system needs
to be reduced, which means the number of potential vulnerabilities needs to be reduced as much as possible.
A bastion host does not have to be a firewall—the term just relates to the position
of the system in relation to an untrusted environment and its threat of attack. Different systems can be considered bastion hosts (mail, web, DNS, etc.) if they are
placed on the outer edges of networks.

The “Shoulds” of Firewalls

PART VII

The default action of any firewall should be to implicitly deny any packets not explicitly
allowed. This means that if no rule states that the packet can be accepted, that packet
should be denied, no questions asked. Any packet entering the network that has a source
address of an internal host should be denied. Masquerading, or spoofing, is a popular
attacking trick in which the attacker modifies a packet header to have the source address
of a host inside the network she wants to attack. This packet is spoofed and illegitimate.
There is no reason a packet coming from the Internet should have an internal source
network address, so the firewall should deny it. The same is true for outbound traffic. No
traffic should be allowed to leave a network that does not have an internal source address.
If this occurs, it means someone, or some program, on the internal network is spoofing
traffic. This is how zombies work—the agents used in distributed DoS (DDoS) attacks.
If packets are leaving a network with different source addresses, these packets are spoofed
and the network is most likely being used as an accomplice in a DDoS attack.
Firewalls should reassemble fragmented packets before sending them on to their destination. In some types of attacks, the hackers alter the packets and make them seem
to be something they are not. When a fragmented packet comes to a firewall, the firewall
is seeing only part of the picture. It makes its best guess as to whether this piece of a packet
is malicious or not. Because these fragments contain only a part of the full packet, the

firewall is making a decision without having all the facts. Once all fragments are allowed through to a host computer, they can be reassembled into malicious packages that can cause a lot of damage. A firewall should accept each fragment, assemble the fragments

into a complete packet, and then make an access decision based on the whole packet.

The drawback to this, however, is that firewalls that do reassemble packet fragments before allowing them to go on to their destination computer cause traffic delay and more overhead. It is up to the organization to decide whether this configuration is necessary and whether the added traffic delay is acceptable.

Many organizations choose to deny network entrance to packets that contain source routing information, which was mentioned earlier. Source routing means that the packet decides how to get to its destination, not the routers in between the source and destination computer. Source routing moves a packet throughout a network on a predetermined path. The sending computer must know about the topology of the network and how to route data properly. This is easier for the routers and connection mechanisms in between, because they do not need to make any decisions on how to route the packet. However, it can also pose a security risk. When a router receives a packet that contains source routing information, the router assumes the packet knows what needs to be done and passes the packet on. In some cases, not all filters may be applied to the packet, and a network administrator may want packets to be routed only through a certain path and not the route a particular packet dictates. To make sure none of this misrouting happens, many firewalls are configured to check for source routing information within the packet and deny it if it is present.

Firewalls are not effective “right out of the box.” You really need to understand the type of firewall being implemented and its configuration ramifications. For example, a firewall may have implied rules, which are used before the rules you configure. These implied rules might contradict your rules and override them. In this case, you may think that a certain traffic type is being restricted, but the firewall allows that

type of traffic into
your network by default.
The following list addresses some of the issues that need you need to understand as
they pertain to firewalls:

• Most of the time a distributed approach needs to be used to control all network
access points, which cannot happen through the use of just one firewall.
• Firewalls can present a potential bottleneck to the flow of traffic and a single
point of failure threat.
• Some firewalls do not provide protection from malware and can be fooled by the more sophisticated attack types.
• Firewalls do not protect against sniffers or rogue wireless access points and provide little protection against insider attacks.

The role of firewalls is becoming more and more complex as they evolve and take on more functionality and responsibility. At times, this complexity works against
security professionals because it requires them to understand and properly implement
additional functionality. Without an understanding of the different types of firewalls and
architectures available, many more security holes can be introduced, which lays out the
welcome mat for attackers.

Intrusion Detection and Prevention Systems

The options for intrusion detection and prevention include host-based intrusion detection systems (HIDSs), network-based intrusion detection systems (NIDSs), and wireless intrusion detection systems (WIDSs). Each may operate in detection or prevention
mode depending on the specific product and how it is employed. As a refresher, the main
difference between an intrusion detection system (IDS) and an intrusion prevention
system (IPS) is that an IDS only detects and reports suspected intrusions, while an IPS
detects, reports, and stops suspected intrusions. How do they do this? There are two basic
approaches: rule-based or anomaly-based.

Rule-Based IDS/IPS

Rule-based intrusion detection and prevention is the simplest and oldest technology.
Essentially, we write rules (or subscribe to a service that writes them for us) and load
those onto the system. The IDS/IPS monitors the environment in which it is placed,
looking for anything that matches a rule. For example, suppose you have a

signature for
a particular piece of malware. You could create a rule that looks for any data that matches
that signature and either raise an alert (IDS) or drop the data and generate the alert (IPS).
Rule-based approaches are very effective when we know the telltale signs of an attack. But
what if the attacker changes tools or procedures?
The main drawback of rule-based approaches to detecting attacks is that we need to have a rule that accurately captures the attack. This means someone got hacked,
investigated the compromise, generated the rule, and shared it with the community.
This process takes time and, until the rule is finalized and loaded, the system won't be
effective against that specific attack. Of course, there's nothing stopping the adversary
from slightly modifying tools or techniques to bypass your new rule either.

Anomaly-Based IDS/IPS

PART VII

Anomaly-based intrusion detection and prevention uses a variety of approaches to detect
things that don't look right. One basic approach is to observe the environment for some
time to figure out what "normal" looks like. This is called the training mode. Once it has
created a baseline of the environment, the IDS/IPS can be switched to testing mode, in
which it compares observations to the baselines created earlier. Any observation that is
significantly different generates an alert. For example, a particular workstation has a pattern of behavior during normal working hours and never sends more than, say, 10MB of
data to external hosts during a regular day. One day, however, it sends out 100MB. That
is pretty anomalous, so the IDS/IPS raises an alert (or blocks the traffic). But what if that
was just the annual report being sent to the regulators?
The main challenge with anomaly-based approaches is that of false positives; that
is, detecting intrusions when none happened. False positives can lead to fatigue and
desensitizing the personnel who need to examine each of these alerts. Conversely, false
negatives are events that the system incorrectly classifies as benign, delaying the response
until the intrusion is detected through some other means. Obviously, both are bad
outcomes.

EDR, NDR, and XDR

HIDS and antimalware features are increasingly being bundled into comprehensive endpoint detection and response (EDR) platforms. Similarly, NIDSs are evolving into network detection and response (NDR) products. These newer solutions do everything that HIDSs and NIDSs do, but also offer a host of other features such as combining rule-based and anomaly detection capabilities. Extended detection and response (XDR) platforms take this one step further by correlation of events across multiple sensors, both in the cloud and on premises, to get a more holistic view of what is going on in an environment.

Perhaps the most important step toward reducing errors is to baseline the system. Baselining is the process of establishing the normal patterns of behavior for a given network or system. Most of us think of baselining only in terms of anomaly-based IDSs because these typically have to go through a period of learning before they can determine what is anomalous. However, even rule-based IDSs should be configured in accordance with whatever is normal for an organization. There is no such thing as a one-size-fits-all set of IDS/IPS rules, though some individual rules may very well be applicable to all (e.g., detecting a known specimen of malware).

NOTE The term “perimeter” has lost some of its importance of late. While it remains an important concept in terms of security architecting, it can mislead some into imagining a wall separating us from the bad guys. A best practice is to assume the adversaries are already “inside the wire,” which downplays the importance of a perimeter in security operations.

Whitelisting and Blacklisting

One of the most effective ways to tune detection platforms like IDS/IPS is to develop lists of things that are definitely benign and those that are definitely malicious. The platform, then, just has to figure out the stuff that is not on either list. A whitelist (more inclusively called an allow list) is a set of known-good resources such as IP addresses, domain names, or applications. Conversely, a blacklist (also known as a deny list) is a set of known-bad resources. In a perfect world, you would only want to use whitelists, because nothing outside of them would ever be allowed in your environment. In reality, we end up using them in specific cases in which we have complete knowledge of the acceptable resources.

For example, whitelisting applications that can execute on a computer is an

effective
control because users shouldn’t be installing arbitrary software on their own. Similarly,
we can whitelist devices that are allowed to attach to our networks.
Things are different when we can’t know ahead of time all the allowable resources.
For example, it is a very rare thing for an organization to be able to whitelist websites for
every user. Instead, we would rely on blacklists of domain and IP addresses. The problem
with blacklists is that the Internet is such a dynamic place that the only thing we can

be sure of is that our blacklist will always be incomplete. Still, blacklisting is better than
nothing, so we should always try to use whitelists first, and then fall back on blacklists
when we have no choice.

Antimalware Software

Traditional antimalware software uses signatures to detect malicious code. Signatures,
sometimes referred to as fingerprints, are created by antimalware vendors. A signature is
a set of code segments that a vendor has extracted from a malware sample. Similar to how
our bodies have antibodies that identify and go after specific pathogens by matching segments of their genetic codes, antimalware software has an engine that scans files, e-mail
messages, and other data passing through specific protocols and then compares them to
its database of signatures. When there is a match, the antimalware software carries out
whatever activities it is configured to do, which can be to quarantine the item, attempt
to clean it (remove the malware), provide a warning message dialog box to the user, and/
or log the event.
Signature-based detection (also called fingerprint detection) is a reasonably effective way
to detect conventional malware, but it has a delayed response time to new threats. Once
malware is detected in the wild, the antimalware vendor must study it, develop and test a
new signature, release the signature, and all customers must download it. If the malicious
code is just sending out silly pictures to all of your friends, this delay is not so critical.

If the malicious software is a new variant of TrickBot (a versatile Trojan behind many ransomware attacks), this amount of delay can be devastating.

Since new malware is released daily, it is hard for the signature-based vendors to keep up. Another technique that almost all antimalware software products use is referred to as heuristic detection. This approach analyzes the overall structure of the malicious code, evaluates the coded instructions and logic functions, and looks at the type of data within the virus or worm. So, it collects a bunch of information about this piece of code and assesses the likelihood of it being malicious in nature. It has a type of "suspiciousness counter," which is incremented as the program finds more potentially malicious attributes. Once a predefined threshold is met, the code is officially considered dangerous and the antimalware software jumps into action to protect the system. This allows antimalware software to detect unknown malware, instead of just relying on signatures.

As an analogy, let's say Barney is the town cop who is employed to root out the bad guys and lock them up (quarantine). If Barney uses a signature method, he compares a stack of photographs of bad actors to each person he sees on the street. When he sees a match, he quickly throws the bad guy into his patrol car and drives off. By contrast, if he uses a heuristic method, he watches for suspicious activity. So if someone with a ski mask is standing outside a bank, Barney assesses the likelihood of this being a bank robber against it just being a cold guy in need of some cash.

Some antimalware products create a simulated environment, called a virtual machine or sandbox, and allow some of the logic within the suspected code to execute in the protected environment. This allows the antimalware software to see the code in question in action, which gives it more information as to whether or not it is malicious.

NOTE The virtual machine or sandbox is also sometimes referred to as an emulation buffer. They are all the same thing—a piece of memory that is segmented and protected so that if the code is malicious, the system is protected.

Reviewing information about a piece of code is called static analysis, while allowing a portion of the code to run in a virtual machine is called dynamic analysis. They

are both
considered heuristic detection methods.
Now, even though all of these approaches are sophisticated and effective, they are not
100 percent effective because malware writers are crafty. It is a continual cat-and-mouse
game that is carried out every day. The antimalware industry comes out with a new way
of detecting malware, and the very next week the malware writers have a way to get
around this approach. This means that antimalware vendors have to continually increase
the intelligence of their products and you have to buy a new version every year.
The next phase in the antimalware software evolution is referred to as behavior blockers.
Antimalware software that carries out behavior blocking actually allows the suspicious
code to execute within the operating system unprotected and watches its interactions
with the operating system, looking for suspicious activities. The antimalware software
watches for the following types of actions:

- Writing to startup files or the Run keys in the Windows registry
- Opening, deleting, or modifying files
- Scripting e-mail messages to send executable code
- Connecting to network shares or resources
- Modifying an executable logic
- Creating or modifying macros and scripts
- Formatting a hard drive or writing to the boot sector

If the antimalware program detects some of these potentially malicious activities, it can
terminate the software and provide a message to the user. The newer-generation behavior
blockers actually analyze sequences of these types of operations before determining the
system is infected. (The first-generation behavior blockers only looked for individual
actions, which resulted in a large number of false positives.) The newer-generation
software can intercept a dangerous piece of code and not allow it to interact with other
running processes. They can also detect rootkits. In addition, some of these antimalware
programs can allow the system to roll back to a state before an infection took place so the
damages inflicted can be “erased.”
While it sounds like behavior blockers might bring us our well-deserved bliss and
utopia, one drawback is that the malicious code must actually execute in real time;
otherwise, our systems can be damaged. This type of constant monitoring also requires a
high level of system resources. We just can’t seem to win.

EXAM TIP Heuristic detection and behavior blocking are considered proactive and can detect new malware, sometimes called “zero-day” attacks. Signature-based detection cannot detect new malware.

Most antimalware vendors use a blend of all of these technologies to provide as much protection as possible. The individual antimalware attack solutions are shown in Figure 21-13.

NOTE Another antimalware technique is referred to as reputation-based protection. An antimalware vendor collects data from many (or all) of its customers’ systems and mines that data to search for patterns to help identify good and bad files. Each file type is assigned a reputation metric value, indicating the probability of it being “good” or “bad.” These values are used by the antimalware software to help it identify “bad” (suspicious) files.

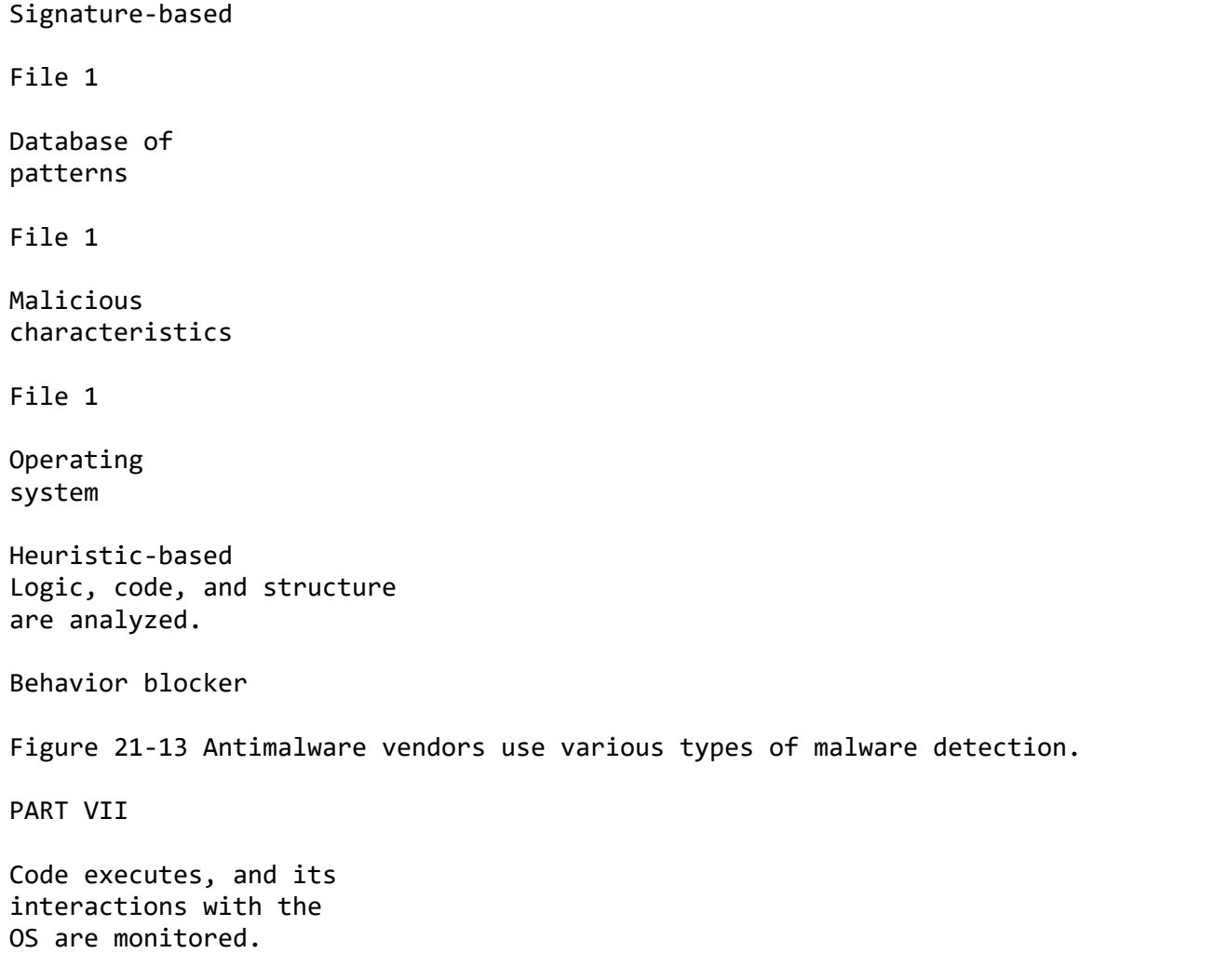

Figure 21-13 Antimalware vendors use various types of malware detection.

Detecting and protecting an enterprise from the long list of malware requires

more
than just rolling out antimalware software. Just as with other pieces of a security program,
certain administrative, physical, and technical controls must be deployed and maintained.
The organization should either have a stand-alone antimalware policy or have one incorporated into an existing security policy. It should include standards outlining what
type of antimalware software and antispyware software should be installed and how they
should be configured.
Antimalware information and expected user behaviors should be integrated into the
security-awareness program, along with who users should contact if they discover a virus. A
standard should cover the do's and don'ts when it comes to malware, which are listed next:

• Every workstation, server, and mobile device should have antimalware software installed.
• An automated way of updating malware signatures should be deployed on each device.
• Users should not be able to disable antimalware software.
• A preplanned malware eradication process should be developed and a contact person designated in case of an infection.
• All external disks (USB drives and so on) should be scanned automatically.
• Backup files should be scanned.
• Antimalware policies and procedures should be reviewed annually.
• Antimalware software should provide boot malware protection.
• Antimalware scanning should happen at a gateway and on each device.
• Virus scans should be automated and scheduled. Do not rely on manual scans.
• Critical systems should be physically protected so malicious software cannot be
installed locally.
Since malware has cost organizations millions of dollars in operational costs and productivity hits, many have implemented antimalware solutions at network entry points.
The scanning software can be integrated into a mail server, proxy server, or firewall.
(The solutions are sometimes referred to as virus walls.) This software scans incoming
traffic, looking for malware so it can be detected and stopped before entering the network. These products can scan Simple Mail Transport Protocol (SMTP), HTTP, FTP,
and possibly other protocol types, but what is important to realize is that the product
is only looking at one or two protocols and not all of the incoming traffic. This is the
reason each server and workstation should also have antimalware software installed.

Sandboxing
A sandbox is an application execution environment that isolates the executing code from

the operating system to prevent security violations. To the code, the sandbox looks just
like the environment in which it would expect to run. For instance, when we sandbox

an application, it behaves as if it were communicating directly with the OS. In reality, it
is interacting with another piece of software whose purpose is to ensure compliance with
security policies. Another instance is that of software (such as helper objects) running in
a web browser. The software acts as if it were communicating directly with the browser,
but those interactions are mediated by a policy enforcer of some sort. The power
of sandboxes is that they offer an additional layer of protection when running code that we are
not certain is safe to execute.
With Sandbox
Sandbox
Process

Policy
enforcement
mechanism

Operating
system

Without Sandbox
Process

Operating
system

Outsourced Security Services
Nearly all of the preventive and detective measures we've discussed in the preceding
subsections can be outsourced to an external service provider. Why would we want to
do that? Well, for starters, many small and midsized organizations lack the resources
to provide a full team of experienced security professionals. We are
experiencing workforce shortages that are not likely to be solved in the near term. This means that hiring,
training, and retaining qualified personnel is not feasible in many cases. Instead, many
organizations have turned to managed security services providers (MSSPs) for third-party
provided security services.
EXAM TIP Outsourced security services are what (ISC)2 refers to as third-party provided security.

• Requirements Before you start interviewing potential MSSPs, make sure you know your requirements. You can outsource the day-to-day activities, but you can’t outsource your responsibility to understand your own security needs.

PART VII

MSSPs typically offer a variety of services ranging from point solutions to taking over
the installation, operation, and maintenance of all technical (and some cases physical)
security controls. (Sorry, you still have to provide policies and many administrative
controls.) Your costs will vary depending on what you need but, in many cases, you’ll
get more than you could’ve afforded if you were to provide these services in-house. Still,
there are some issues that you should consider before hiring an MSSP:

• Understanding Does the MSSP understand your business processes? Are they asking the right questions to get there? If your MSSP doesn’t know what it is that
your organization does (and how), they will struggle to provide usable security. Likewise, you need to understand their qualifications and processes. Trust is a two-way street grounded on accurate information.
• Reputation It is hard to be a subpar service provider and not have customers complain about you. When choosing an MSSP, you need to devote some time
to reading online reviews and asking other security professionals about their experiences with specific companies.
• Costing You may not be able to afford the deluxe version of the MSSP’s services, so you will likely have to compromise and address only a subset of your
requirements. When you have trimmed down your requirements, is it still more cost-effective to go with this provider? Should you go with another? Should you just do it yourself?
• Liability Any reasonable MSSP will put limits on their liability if your organization is breached. Read the fine print on the contract and consult your attorneys, particularly if you are in an industry that is regulated by the government.

Honeypots and Honeynets
A honeypot is a network device that is intended to be exploited by attackers, with the
administrator’s goal being to gain information on the attackers’ tactics, techniques, and
procedures (TTPs). Honeypots can work as early detection mechanisms, meaning that
the network staff can be alerted that an intruder is attacking a honeypot system, and they
can quickly go into action to make sure no production systems are vulnerable to that specific attack type. A honeypot usually sits in the screened subnet, or

DMZ, and attempts
to lure attackers to it instead of to actual production computers. Think of honeypots as
marketing devices; they are designed to attract a segment of the market, get them to buy
something, and keep them coming back. Meanwhile, threat analysts are keeping tabs on
their adversaries' TTPs.
To make a honeypot system alluring to attackers, administrators may enable services
and ports that are popular to exploit. Some honeypot systems emulate services, meaning
the actual services are not running but software that acts like those services is available.
Honeypot systems can get an attacker's attention by advertising themselves as easy targets
to compromise. They are configured to look like the organization's regular systems so
that attackers will be drawn to them like bears are to honey.
Another key to honeypot success is to provide the right kind of bait. When someone
attacks your organization, what is it that they are after? Is it credit card information,
patient files, intellectual property? Your honeypots should look like systems that would
allow the attacker to access the assets for which they are searching. Once compromised,
the directories and files containing this information must appear to be credible. It should
also take a long time to extract the information, so that we maximize the contact time
with our "guests."

A honeynet is an entire network that is meant to be compromised. While it may be
tempting to describe honeynets as networks of honeypots, that description might be
a bit misleading. Some honeynets are simply two or more honeypots used together.
However, others are designed to ascertain a specific attacker's intent and dynamically
spawn honeypots that are designed to be appealing to that particular attacker. As you can
see, these very sophisticated honeynets are not networks of preexisting honeypots, but
rather adaptive networks that interact with the adversaries to keep them engaged (and
thus under observation) for as long as possible.
NOTE Black holes are sometimes confused with honeynets, when in reality
they are almost the opposite of them. Black holes typically are routers
with rules that silently drop specific (typically malicious) packets without
notifying the source. They normally are used to render botnet and other
known-bad traffic useless. Whereas honeypots and honeynets allow us to