Covers all Exam Objectives for CEHv6

Includes Real-World Scenarios, Hands-On Exercises, and Leading-Edge Exam Prep Software Featuring:

- Custom Test Engine
- Hundreds of Sample Questions
- Electronic Flashcards
- Entire Book in PDF

CEH™

Certified Ethical Hacker

STUDY GUIDE

Exam 312-50
Exam EC0-350

Kimberly Graves

SYBEX | SERIOUS SKILLS.

Table of Contents

Chapter 13

Bypassing Network Security: Evading IDSs, Honeypots, and Firewalls

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

- ✓ List the types of intrusion detection systems and evasion techniques
- ✓ List firewall types and honeypot evasion techniques

Intrusion detection systems (IDS), firewalls, and honeypots are all security measures used to ensure a hacker is not able to gain access to a network or target system. An IDS and a firewall are both essentially packet filtering devices and are used to monitor traffic based on a predefined set of rules. A honeypot is a fake target system used to lure hackers away from the more valuable targets. As with other security mechanisms, IDSs, firewalls, and honeypots are only as good as their design and implementation. It is important to be familiar with how these devices operate and provide security as they are commonly subjects of attack.

Types of IDSs and Evasion Techniques

Intrusion detection systems (IDSs) inspect traffic and look for known signatures of attacks or unusual behavior patterns. A *packet sniffer* views and monitors traffic and is a built-in component of an IDS. An IDS alerts a command center or system administrator by pager, email, or cell phone when an event appearing on the company's security event list is triggered. *Intrusion prevention systems* (IPSs) initiate countermeasures such as blocking traffic when suspected traffic flow is detected. IPSs automate the response to an intrusion attempt and allow you to automate the deny-access capability.

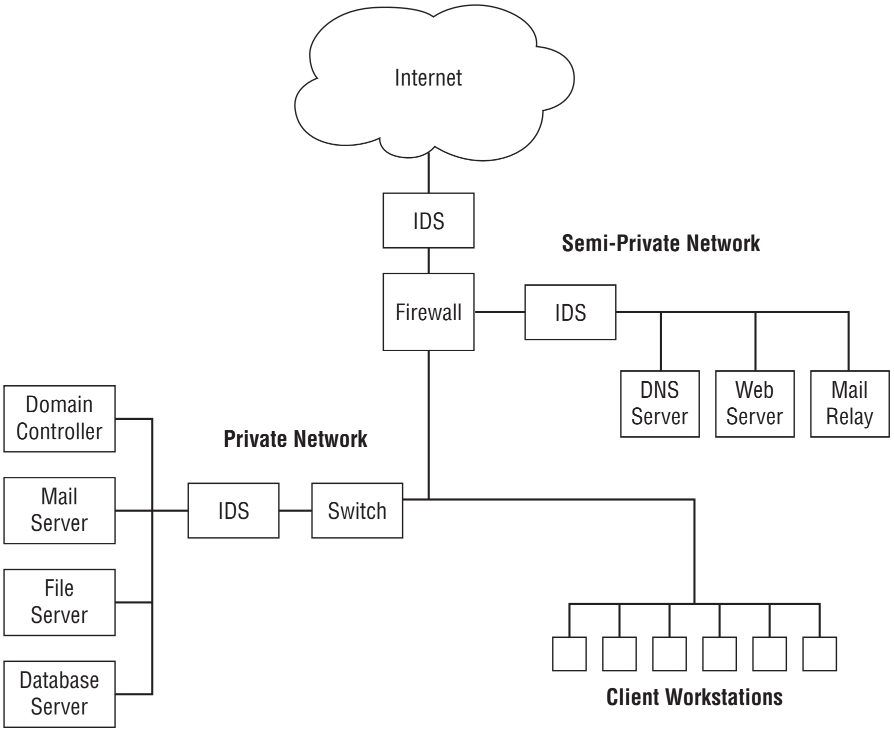
There are two main types of IDS:

Host Based Host-based IDSs (HIDSs) are applications that reside on a single system or host and filter traffic or events based on a known signature list for that specific operating system. HIDSs include Norton Internet Security and Cisco Security Agent (CSA). Many worms and Trojans can turn off an HIDS. HIDSs can also be installed directly on servers to detect attacks against corporate resources and applications.

Network Based Network-based IDSs (NIDSs) are software-based appliances that reside on the network. They're used solely for intrusion detection purposes to detect all types of malicious network traffic and computer usage that can't be detected by a conventional firewall. This includes network attacks against vulnerable services; data attacks on applications; host-based attacks such as privilege escalation, unauthorized logins, and access to sensitive files; and malware. NIDSs are *passive* systems: the IDS sensor detects a potential security breach, logs the information, and signals an alert on the console.

The location of a network-based IDS in a network architecture is depicted in Figure 13.1. A network IDS sensor can be located as a first point of detection between the firewall and the Internet or on the semi-private DMZ, detecting attacks on the organization's servers. Finally, a network IDS can be located on the internal private network, with the corporate servers detecting possible attacks on those servers.

FIGURE 13.1 Network-based IDS



An IDS can perform either signature analysis or anomaly detection to determine if the traffic is a possible attack. Signature detection IDSs match traffic with known signatures and patterns of misuse. A *signature* is a pattern used to identify either a single packet or a series of packets that, when combined, execute an attack. An IDS that employs anomaly detection looks for intrusion attempts based on a person's normal business patterns and alerts when there is an anomaly in the behavior of access to systems, files, logins, and so on.

A hacker can evade an IDS by changing the traffic so that it does not match a known signature. This may involve using a different protocol such as UDP instead of TCP or HTTP instead of ICMP to deliver an attack. Additionally, a hacker can break an attack up into several smaller packets to pass through an IDS but, when reassembled at the receiving station, will result in a compromise of the system. This is known as session splicing. Other methods of evading detection involve inserting extra data, obfuscating addresses or data by using encryption, or desynchronizing and taking over a current client's session.

Hacking Tool

ADMmutate takes an attack script and creates a different—but functionally equivalent—script to perform the attack. The new script isn't in the database of known attack signatures and therefore can bypass the IDS.

Understanding Snort Rules and Output

For the CEH exam, you should be familiar with Snort rules and output. You may need to read a Snort rule or output and answer a question pertaining to what the rule is doing or what type of attack is indicated by the output.

Snort is a real-time packet sniffer, HIDS, and traffic-logging tool deployed on Linux and Windows systems. Snort can analyze protocols, perform content searching/matching, and detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. You can configure Snort and the IDS rules in the `snort.conf` file. The command to install and run Snort is:

```
snort –l c:\snort\log –c C:\snort\etc\snort.conf –A console
```

Snort consists of two major components:

Snort Engine An IDS detection engine that utilizes a modular plug-in architecture

Snort Rules A flexible rule language to describe traffic to be collected

The Snort Engine is distributed both as source code and binaries for popular Linux distributions and Windows. It's important to note that the Snort Engine and Snort rules are distributed separately. The Snort IDS Engine and rules can be downloaded from `snort.org`. The installation methods and software dependencies vary by OS, so this chapter does not include a lab on installing Snort. Detailed installation instructions can be found at `snort.org`.

Configuring Snort

Snort has one configuration file: `snort.conf`. It usually resides in `/etc/snort`. The file contains variables that need to be modified for your specific installation and customized to the events you want to alert on. The file variables are organized in the following sections:

- Network variables
- Preprocessors
- Postprocessors
- Rules

The `snort.conf` file network variables that need to be customized to your network are listed in Table 13.1.

TABLE 13.1 Snort variables

Variable	Meaning
HOME_NET	Local IP address space
EXTERNAL_NET	External IP address space
SMTP	Your SMTP servers
HTTP_SERVERS	Your web servers
SQL_SERVERS	Your SQL Servers
DNS_SERVERS	Your DNS servers
RULE_PATH	The directory that contains your rule files

Here is a sample Snort configuration file using the 192.168.1.0 network as the home network:

```
var HOME_NET 192.168.1.0/24
var EXTERNAL_NET any
var SMTP $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var DNS_SERVERS $HOME_NET
var RULE_PATH /etc/snort/rules
```

The following are the rule locations identified in the config file:

```
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
```

Snort Rules

Snort rules are used to generate alerts based on the traffic that is viewed by the IDS processing engine.

All rules have a rule header composed of the following fields:

- `<rule action>`
- `<protocol>`
- `<src address & port>`
- `<dest address & port>`

Here's an example of a Snort rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 23
```

This rule says to generate an alert (and a log message) for any TCP packet coming from an external address space (and any port) destined to the local address space (and port 23).

The Snort rule header is followed by rule options, which are a delimited list of features to use in Snort. Here are some rule options and explanations. The line

```
msg:"TELNET SGI telnetd format bug"
```

specifies to the logging and alerting engines what message to print. The line

```
flags: A+
```

matches the TCP `ACK` flag (plus any other set flag). The line

```
content: "bin/sh"
```

matches the given string in the packet's payload. The line

```
classtype:attempted-admin
```

associates a high priority to this alert by giving it an *attack class* of `attempted-admin` (attempted administrator privilege gain).

Snort Output

For the CEH exam, it is important to understand a Snort output report. Here is an example of a Snort alert. First, here is the timestamp:

```
04/21-19:26:37.353790
```

These are the source and destination MAC addresses:

```
0:8:2:FB:36:C6 -> 0:6:5B:57:A6:3F
```

The type of Ethernet frame (0x800 means Ethernet) and the length are next:

```
type:0x800 len:0x3C
```

This line specifies the source IP 202.185.44.43 to the destination IP 202.185.44.28 and source port 445 and destination port 2202:

```
202.185.44.43:445 -> 202.185.44.28:2202
```

This line states that the protocol is TCP and the Time To Live (TTL) is 128:

```
TCP TTL:128
```

Next is the type of service, the ID, the IP length, and the datagram length:

```
TOS:0x0 ID:17467 IpLen:20 DgmLen:41 DF
```

The ***A**** means the ACK flag is on, so the packet is an acknowledgment of a previous packet:

```
***A****
```

In this line, Seq is the sequence number, and Ack is the numbered response to the previous packet:

```
Seq: 0x9D08DD67 Ack: 0x83EB1E02
```

Finally, in the following line Win is the window size and the TCP length is 2000:

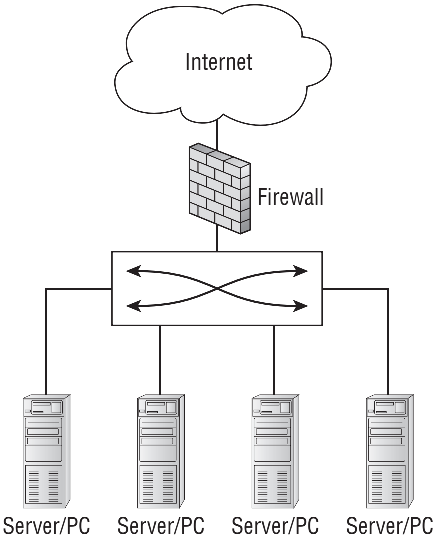
```
Win: 0x3FE1 TcpLen: 2000
```

NOTE In many cases, reading and interpreting Snort output reports on the CEH exam is just a matter of knowing the TCP flags and TCP well-known port numbers.

Firewall Types and Honeypot Evasion Techniques

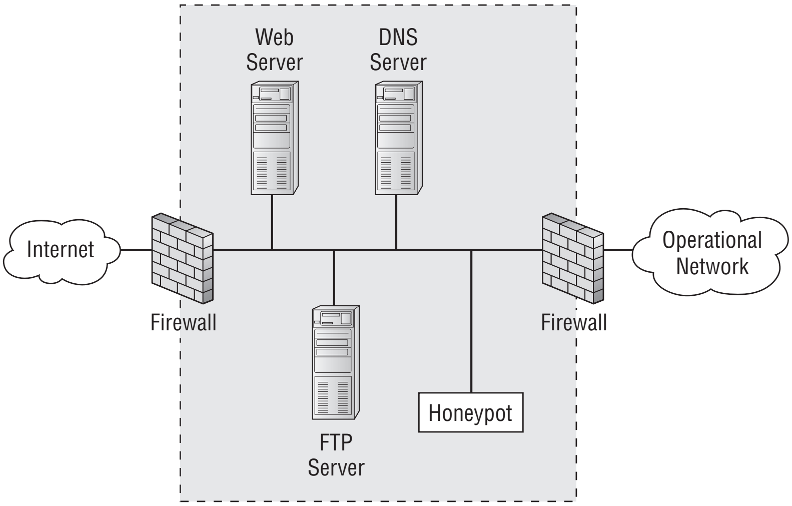
A *firewall* is a software program or hardware appliance that allows or denies access to a network and follows rules set by an administrator to direct where packets are allowed to go on the network. A *perimeter hardware firewall* appliance (Figure 13.2) is set up either at the network edge where a trusted network connects to an untrusted network, such as the Internet, or between networks. A *software firewall* protects a personal computer, a system, or a host from unwanted or malicious packets entering the network interface card (NIC) from the network.

FIGURE 13.2 Perimeter hardware firewall



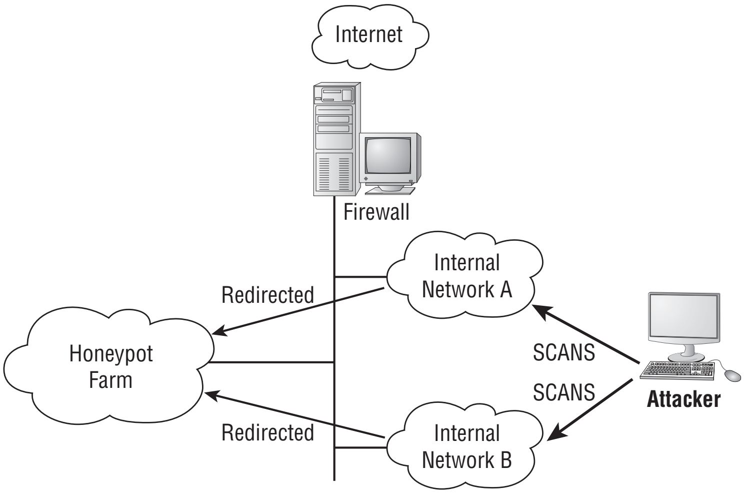
A *honeypot* (Figure 13.3) is a decoy box residing inside your network demilitarized zone (DMZ), set up by a security professional to trap or aid in locating hackers, or to draw them away from the real target system.

FIGURE 13.3 Honeypot Location



The honeypot is a decoy system that a malicious attacker might try to attack; software on the system can log information about the attacker such as the IP address. This information can be used to try to locate the attacker either during or after the attack. The best location for a honeypot is in front of the firewall on the DMZ, making it attractive to hackers. A honeypot with a static address is designed to look like a real production server (see Figure 13.4). Exercise 13.1 walks you through installing and using a honeypot.

FIGURE 13.4 Honeypot



Real World Scenario

Finding a Honeypot

I was performing a wireless network security audit for a large corporation a few years ago. I drove around the corporate campus scanning for open access points (APs), and I was a bit surprised at how many open unsecured APs could be seen by my wireless scanning sniffer. I found over 30 APs to which I could connect and gain network access.

Of course, the next step after connecting to the APs was to scan the network. So, as part of the security audit, I connected from outside the building and ran a port scan against the entire network range; I found several systems with open ports. There was a mail server and a couple of web servers, as well as a Domain Controller that was not totally patched. As per the scope of the audit, I was just to report the vulnerabilities I found and not attempt to exploit the services I found running on the systems. I was surprised that such a large organization would have vulnerabilities so easily found on the open wireless network. I documented all the target systems and the vulnerable ports and services in my security auditing report.

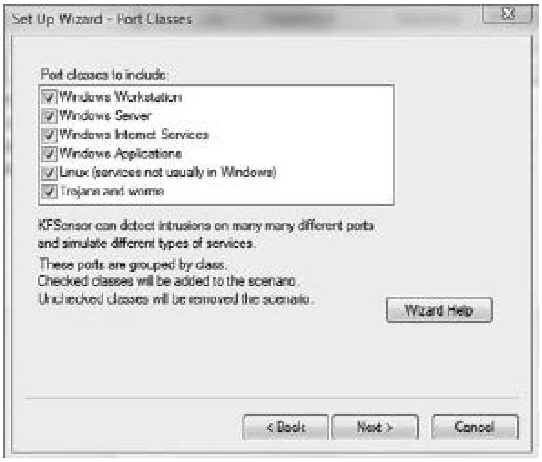
When I presented my report to the customer the following day, the IT manager simply said, “Good, you found our honeynet, now go find the real systems.” They had taken all the rogue APs discovered on the network and shunted them to a separate VLAN. Then on the shunted VLAN they had created fake systems, or honeypots, to attract potential hackers. These honeypots can keep a hacker busy trying to attack the honeypot system with no real data while the real services are untouched.

EXERCISE 13.1

Installing and Using KFSensor as a Honeypot

1. Download and install a trial version of KFSensor from `www.keyfocus.net`.
2. Open and run KFSensor. A pop-up window will appear to start the configuration wizard. Click Next to continue.

EXERCISE 13.1 *(continued)*



Set Up Wizard - Port Classes

Port classes to include:

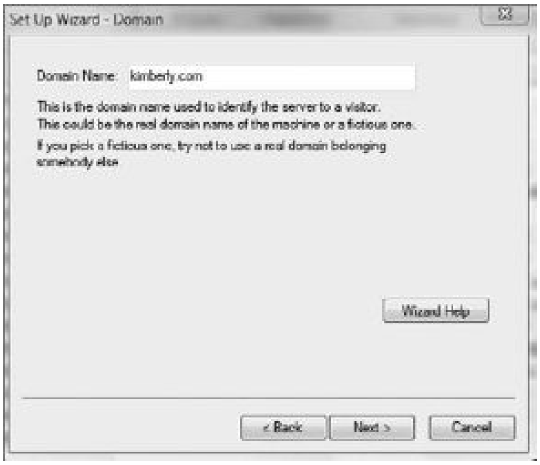
☑ Windows Workstation
☑ Windows Server
☑ Windows Internet Services
☑ Windows Applications
☑ Linux (services not usually in Windows)
☑ Trojans and worms

KFSensor can detect intrusions on many many different ports and simulate different types of services.

These ports are grouped by class.
Checked classes will be added to the scenario.
Unchecked classes will be removed the scenario.

3. Click Next to select all ports.

4. Type ***`your name`*** **`.com`** (or another domain name of your choosing) in the Domain Name field and click Next.



Set Up Wizard - Domain

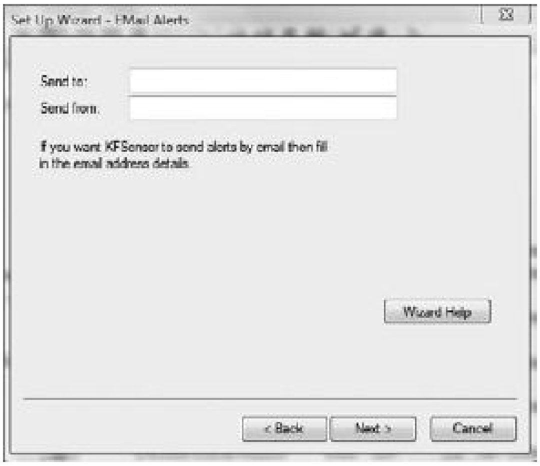
Domain Name: kimberly.com

This is the domain name used to identify the server to a visitor.
This could be the real domain name of the machine or a fictious one.

If you pick a fictious one, try not to use a real domain belonging somebody else

EXERCISE 13.1 *(continued)*

5. Type your email address in the Send To and Send From fields to receive email alerts from KFSensor.



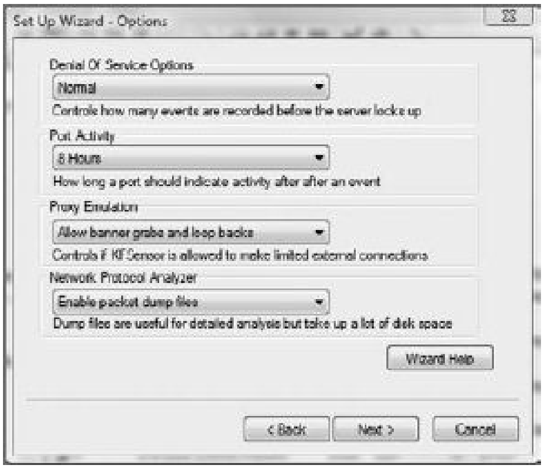
Set Up Wizard - EMail Alerts

Send to:

Send from:

If you want KFSensor to send alerts by email then fill in the email address details

6. From the Port Activity drop-down, select 8 hours. Choose Enable Packet Dump Files from the Network Protocol Analyzer drop-down. Other options can remain at their defaults.



Set Up Wizard - Options

Denial Of Service Options: Normal
Controls how many events are recorded before the server locks up

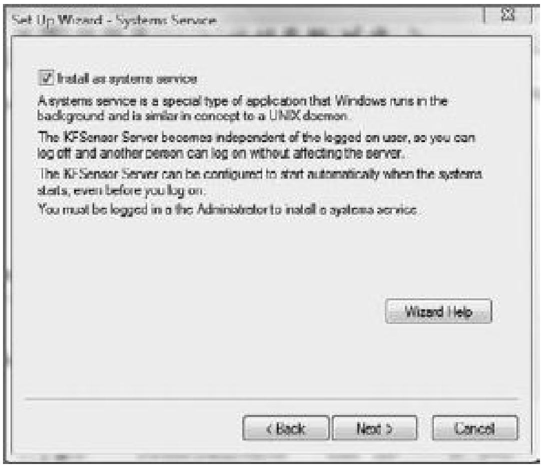
Port Activity: 8 Hours
How long a port should indicate activity after after an event

Proxy Emulation: Allow banner grabs and loop backs
Controls if KFSensor is allowed to make limited external connections

Network Protocol Analyzer: Enable packet dump files
Dump files are useful for detailed analysis but take up a lot of disk space

EXERCISE 13.1 *(continued)*

7. Click Next to accept the default to install as a system service.



☑ Install as systems service

A systems service is a special type of application that Windows runs in the background and is similar in concept to a UNIX daemon.

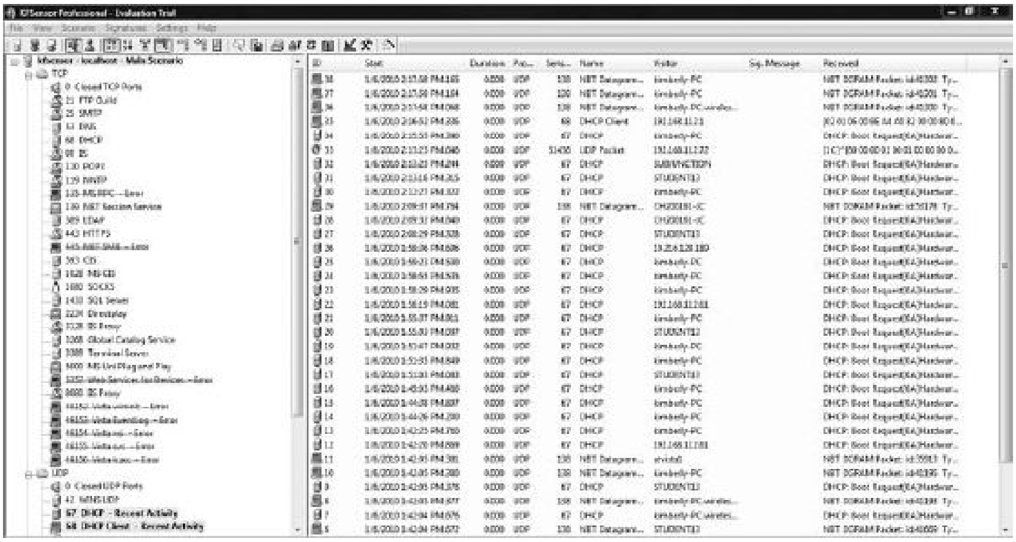
The KFSensor Server becomes independent of the logged on user, so you can log off and another person can log on without affecting the server.

The KFSensor Server can be configured to start automatically when the systems starts, even before you log on.

You must be logged in a the Administrator to install a systems service.

8. Click Finish to complete the wizard configuration.

9. The Main scenario for KFSensor should appear on the left. You may receive a message indicating that some of the ports have been disabled because they are in use by the system services; the strikeout text indicates the ports are not available in KFSensor.



KFSensor Professional - Evaluation Trail
File View Scenario Signatures Settings Help

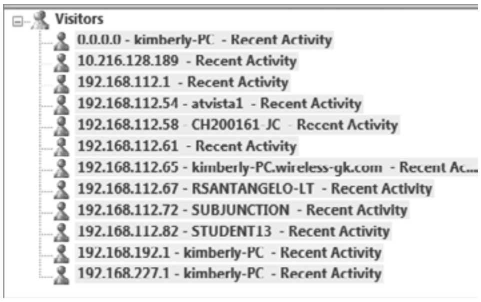
kfsensor - localhost - Main Scenario
- TCP
  - 0 Closed TCP Ports
  - 21 FTP Guild
  - 25 SMTP
  - 53 DNS
  - 68 DHCP
  - 80 IIS
  - 110 POP3
  - 119 NNTP
  - 135 MS RPC -- Error
  - 139 NBT Session Service
  - 389 LDAP
  - 443 HTTPS
  - ~~445 NBT SMB -- Error~~
  - 593 CIS
  - 1028 MS CIS
  - 1080 SOCKS
  - 1433 SQL Server
  - 2234 Directplay
  - 3128 IIS Proxy
  - 3268 Global Catalog Service
  - 3389 Terminal Server
  - 5000 MS Uni Plug and Play
  - ~~5357 Web Services for Devices -- Error~~
  - 8080 IIS Proxy
  - ~~49152 Vista wininit -- Error~~
  - ~~49153 Vista Eventlog -- Error~~
  - ~~49154 Vista svc -- Error~~
  - ~~49155 Vista svc -- Error~~
  - ~~49156 Vista svc -- Error~~
- UDP
  - 0 Closed UDP Ports
  - 42 WINS UDP
  - **67 DHCP - Recent Activity**
  - **68 DHCP Client - Recent Activity**

ID	Start	Duration	Pro...	Sens...	Name	Visitor	Sig. Message	Received
38	1/6/2010 2:17:58 PM.165	0.000	UDP	138	NBT Datagram...	kimberly-PC		NBT DGRAM Packet: id:40392 Ty...
37	1/6/2010 2:17:58 PM.164	0.000	UDP	138	NBT Datagram...	kimberly-PC		NBT DGRAM Packet: id:40391 Ty...
36	1/6/2010 2:17:48 PM.068	0.000	UDP	138	NBT Datagram...	kimberly-PC.wireles...		NBT DGRAM Packet: id:40390 Ty...
35	1/6/2010 2:16:52 PM.236	0.000	UDP	68	DHCP Client	192.168.11.21		[02 01 06 00 86 A4 68 82 00 00 80 0...
34	1/6/2010 2:15:53 PM.380	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
33	1/6/2010 2:13:23 PM.045	0.000	UDP	51436	UDP Packet	192.168.11.22		[1C]^[00 00 00 01 00 01 00 00 00 0...
32	1/6/2010 2:12:23 PM.244	0.000	UDP	67	DHCP	SUBWVICTION		DHCP: Boot Request(6A)Hardwar...
31	1/6/2010 2:11:16 PM.215	0.000	UDP	67	DHCP	STUDENT12		DHCP: Boot Request(6A)Hardwar...
30	1/6/2010 2:11:27 PM.322	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
29	1/6/2010 2:09:51 PM.794	0.000	UDP	138	NBT Datagram...	CH20018-JC		NBT DGRAM Packet: id:36178 Ty...
28	1/6/2010 2:09:31 PM.049	0.000	UDP	67	DHCP	CH20018-JC		DHCP: Boot Request(6A)Hardwar...
27	1/6/2010 2:08:29 PM.328	0.000	UDP	67	DHCP	STUDENT13		DHCP: Boot Request(6A)Hardwar...
26	1/6/2010 1:58:06 PM.606	0.000	UDP	67	DHCP	19.216.128.180		DHCP: Boot Request(6A)Hardwar...
25	1/6/2010 1:56:21 PM.509	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
24	1/6/2010 1:58:54 PM.526	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
23	1/6/2010 1:58:29 PM.035	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
22	1/6/2010 1:58:19 PM.081	0.000	UDP	67	DHCP	192.168.11.101		DHCP: Boot Request(6A)Hardwar...
21	1/6/2010 1:55:37 PM.011	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
20	1/6/2010 1:55:30 PM.037	0.000	UDP	67	DHCP	STUDENT12		DHCP: Boot Request(6A)Hardwar...
19	1/6/2010 1:53:47 PM.032	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
18	1/6/2010 1:52:53 PM.849	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
17	1/6/2010 1:51:30 PM.083	0.000	UDP	67	DHCP	STUDENT13		DHCP: Boot Request(6A)Hardwar...
16	1/6/2010 1:45:03 PM.480	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
15	1/6/2010 1:44:28 PM.837	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
14	1/6/2010 1:44:26 PM.280	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
13	1/6/2010 1:42:25 PM.765	0.000	UDP	67	DHCP	kimberly-PC		DHCP: Boot Request(6A)Hardwar...
12	1/6/2010 1:42:20 PM.699	0.000	UDP	67	DHCP	192.168.11.101		DHCP: Boot Request(6A)Hardwar...
11	1/6/2010 1:42:05 PM.381	0.000	UDP	138	NBT Datagram...	vhvista0		NBT DGRAM Packet: id:35913 Ty...
10	1/6/2010 1:42:05 PM.280	0.000	UDP	138	NBT Datagram...	kimberly-PC		NBT DGRAM Packet: id:40395 Ty...
9	1/6/2010 1:42:05 PM.376	0.000	UDP	67	DHCP	STUDENT13		DHCP: Boot Request(6A)Hardwar...
8	1/6/2010 1:42:05 PM.677	0.000	UDP	138	NBT Datagram...	kimberly-PC.wireles...		NBT DGRAM Packet: id:40388 Ty...
7	1/6/2010 1:42:04 PM.676	0.000	UDP	67	DHCP	kimberly-PC.wireles...		DHCP: Boot Request(6A)Hardwar...
6	1/6/2010 1:42:04 PM.672	0.000	UDP	138	NBT Datagram...	STUDENT12		NBT DGRAM Packet: id:40389 Ty...

Perform a port scan against the system running KFSensor to identify the services.

EXERCISE 13.1 *(continued)*

10. Attempt to connect to a service running on the KFSensor system.
11. View the visitor to the KFSensor Honeypot by clicking the View menu and choosing Visitors.



```
Visitors
  0.0.0.0 - kimberly-PC  - Recent Activity
  10.216.128.189  - Recent Activity
  192.168.112.1  - Recent Activity
  192.168.112.54 - atvista1  - Recent Activity
  192.168.112.58 - CH200161-JC  - Recent Activity
  192.168.112.61  - Recent Activity
  192.168.112.65 - kimberly-PC.wireless-gk.com  - Recent Ac...
  192.168.112.67 - RSANTANGELO-LT  - Recent Activity
  192.168.112.72 - SUBJUNCTION  - Recent Activity
  192.168.112.82 - STUDENT13  - Recent Activity
  192.168.192.1 - kimberly-PC  - Recent Activity
  192.168.227.1 - kimberly-PC  - Recent Activity
```

12. Click the IP address of a visitor to view the connections.
13. KFSensor will continue to run even when the program is closed. To stop the servers completely, right-click the KFSensor icon in the system tray and choose Stop Server.

The easiest way to bypass a firewall is to compromise a system on the trusted or internal side of the firewall. The compromised system can then connect through the firewall, from the trusted to the untrusted side, to the hacker's system. A common method of doing this is to make the compromised system connect to the hacker with destination port 80, which looks just like a web client connecting to a web server through the firewall. This is referred to as a *reverse WWW shell*.

NOTE This attack works because most firewalls permit outgoing connections to be made to port 80 by default.

Using a tunnel to send HTTP traffic, the hacker bypasses the firewall and makes the attack look innocuous to the firewall; such attacks are virtually untraceable by system administrators. Hacking programs can create covert channels, which let the attack traffic travel down an allowed path such as an Internet Control Message Protocol (ICMP) ping request or reply. Another method of utilizing a covert channel tunnels the attack traffic as a TCP acknowledgment.

To evade the trap set by a honeypot, a hacker can run anti-honeypot software, which tries to determine whether a honeypot is running on the target system and warn the hacker about it. In this way, a hacker can attempt to evade detection by not attacking a honeypot. Most anti-honeypot software checks the software running on the system against a known list of honeypots such as honeyd.

Hacking Tools

007 Shell is a shell-tunneling program that lets a hacker use a covert channel for the attack and thus bypass firewall rules.

ICMP Shell is a program similar to telnet that a hacker uses to make a connection to a target system using just ICMP commands, which are usually allowed through a firewall.

AckCmd is a client/server program that communicates using only TCP ACK packets, which can usually pass through a firewall.

Covert_TCP is a program that a hacker uses to send a file through a firewall one byte at a time by hiding the data in the IP header.

Send-Safe Honeypot Hunter is a honeypot-detection tool that checks against a proxy server for honeypots.

Countermeasures

Specter is a honeypot system that can automatically capture information about a hacker's machine while they're attacking the system.

Honeyd is an open source honeypot that creates virtual hosts on a network that is then targeted by hackers.

KFSensor is a host-based IDS that acts as a honeypot and can simulate virtual services and Trojan installations.

Sobek is a data-capturing honeypot tool that captures an attacker's keystrokes.

The Nessus vulnerability scanner (`www.nessus.org`) can also be used to detect honeypots.

Summary

Intrusion detection systems can be either network or host based. It is important to implement both types to protect valuable data on servers from attack. In both cases it is critical to keep the rules and definitions up-to-date to ensure the IDS has the latest attack vectors to compare traffic. Firewalls can also be network or host based, and in many cases network appliances' and systems software will perform both IDS detection and firewalling actions. Just because a firewall and IDS are implemented on a network or server, you should not be lulled into a false sense of security; tunneling and encryption can defeat both IDSs and firewalls because the real traffic headers and data cannot be read by the appliance. A CEH uses such techniques in an attempt to bypass the protection of firewalls and IDSs.

Exam Essentials

Know the two main types of IDSs. IDSs can be either host based or network based. A host-based IDS is operating system specific and protects a single system. A network-based IDS can protect the entire network.

Be able to define a honeypot. A honeypot resides in a DMZ as a vulnerable host and advertises services and software to entice a hacker to hack the system.

Be able to define a firewall. A firewall is a packet-filtering device that compares traffic to a list of rules and filters traffic from an untrusted network to a trusted network.

Understand how to detect a honeypot. A honeypot can be detected by comparing the system information to a known list of honeypots in a proxy server.

Understand how an IDS works. An IDS can either perform anomaly analysis or signature-based detection.

Know how to perform firewall evasion techniques. Firewall evasion can be performed by using a protocol such as ICMP or HTTP to carry attack traffic. Another technique is to split the packets into several smaller packets so the entire attack string cannot be detected.

Review Questions

1. What is a system that performs attack recognition and alerting for a network?

   A. HIDS

   B. NIDS

   C. Anomaly detection HIDS

   D. Signature-based NIDS

2. Which of the following tools bypasses a firewall by sending one byte at a time in the IP header?

   A. Honeyd

   B. Nessus

   C. Covert_TCP

   D. 007 Shell

   E. TCP to IP Hide

3. Which of the following is a honeypot-detection tool?

   A. Honeyd

   B. Specter

   C. KFSensor

   D. Sobek

4. Which of the following is a system designed to attract and identify hackers?

   A. Honeypot

   B. Firewall

   C. Honeytrap

   D. IDS

5. Which of the following is a tool used to modify an attack script to bypass an IDS's signature detection?

   A. ADMmutate

   B. Script Mutate

   C. Snort

   D. Specter

6. What is a reverse WWW shell?

A. A web server making a reverse connection to a firewall

B. A web client making a connection to a hacker through the firewall

C. A web server connecting to a web client through the firewall

D. A hacker connecting to a web server through a firewall

7. A reverse WWW shell connects to which port on a hacker's system?

A. 80

B. 443

C. 23

D. 21

8. What is the command used to install and run Snort?

A. `snort -l c:\snort\log -c C:\snort\etc\snort.conf -A console`

B. `snort -c C:\snort\etc\snort.conf -A console`

C. `snort -c C:\snort\etc\snort.conf console`

D. `snort -l c:\snort\log -c -A`

9. What type of program is Snort?

A. NIDS

B. Sniffer, HIDS, and traffic-logging tool

C. Sniffer and HIDS

D. NIDS and sniffer

10. What are the ways in which an IDS is able to detect intrusion attempts? (Choose all that apply.)

A. Signature detection

B. Anomaly detection

C. Traffic identification

D. Protocol analysis

11. You are viewing a snort output report and see an entry with the following address information: `168.175.44.80:34913 -> 142.155.44.28:443`. What type of server is the destination address?

A. HTTP

B. FTP

C. SSL

D. HTTPS

12. What is the `snort.conf` file variable for the local IP subnet?

    A. `INTERNAL_NET`

    B. `DESTINATION_NETWORK`

    C. `SOURCE_NET`

    D. `HOME_NET`

13. How is the rule location identified in the `snort.conf` file?

    A. `RULE_PATH`

    B. `RULE_DIR`

    C. `RULES`

    D. `RULE_NET`

14. Which field is *not* located in the rule header in a Snort rule?

    A. Rule Action

    B. Protocol

    C. Source Address

    D. `HOME_NET`

15. Which Snort rule option would associate a high priority to an alert?

    A. `class:attempted-admin`

    B. `classtype:High`

    C. `classtype:attempted-admin`

    D. `class:admin`

16. What are the two components needed when installing Snort?

    A. Snort rules

    B. Snort signatures

    C. Snort Engine

    D. Snort processor

17. What is an attack signature in an IDS?

    A. A pattern of packets that indicates an attack

    B. The first packet that indicates the start of an attack

    C. The TCP header that indicates an attack

    D. The confirmation that an attack has occurred

18. What is a method used to defeat an IDS signature match?

    A. Anomaly detection

    B. Tunneling

    C. Packet smashing

    D. Buffer overflows

19. You are reviewing a Snort output report with the following content:

```
10/17-20:28:15.014784 0:10:5A:1:D:5B -> 0:2:B3:87:84:25 type:0x800 len:0x3C
192.168.1.4:1244 -> 192.168.1.67:443 TCP TTL:128 TOS:0x0 ID:39235
IpLen:20 DgmLen:40 DF
***A**** Seq: 0xA18BBE Ack: 0x69749F36 Win: 0x2238 TcpLen: 20
0x0000: 00 02 B3 87 84 25 00 10 5A 01 0D 5B 08 00 45 00 .....%..Z..[..E.
0x0010: 00 28 99 43 40 00 80 06 DD F4 C0 A8 01 04 C0 A8 .(.C@..........
0x0020: 01 43 04 DC 01 BB 00 A1 8B BE 69 74 9F 36 50 10 .C........it.6P.
0x0030: 22 38 6E 63 00 00 00 00 00 00 00 00 "8nc........
```

What TCP flags are set in the packet?

A. ACK

B. SYN

C. FIN

D. RST

20. A Snort file has been retrieved with the following output:

```
10/17-20:28:15.080091 0:2:B3:87:84:25 -> 0:10:5A:1:D:5B type:0x800 len:0x13B
192.168.1.67:443 -> 192.168.1.4:1244 TCP TTL:64 TOS:0x0 ID:6664
IpLen:20 DgmLen:301 DF
***AP*** Seq: 0x6974A4F2 Ack: 0xA18F51 Win: 0x1E51 TcpLen: 20
0x0000: 00 10 5A 01 0D 5B 00 02 B3 87 84 25 08 00 45 00 ..Z..[.....%..E.
0x0010: 01 2D 1A 08 40 00 40 06 9C 2B C0 A8 01 43 C0 A8 .-..@.@..+...C..
0x0020: 01 04 01 BB 04 DC 69 74 A4 F2 00 A1 8F 51 50 18 ......it.....QP.
0x0030: 1E 51 5B AF 00 00 17 03 01 01 00 9D 6D 31 27 DB .Q[.........m1'.
0x0040: 5C 57 B7 39 48 C5 FE 3C 92 77 65 E4 95 49 F4 C5 \W.9H..<.we..I..
0x0050: 5B 98 CB A2 A5 F9 DF C1 F1 6D A2 1A 22 04 E4 DB [........m.."...
0x0060: 4A 1F 18 A9 F8 11 54 57 E6 AF 9A 6C 55 43 8D 37 J.....TW...lUC.7
0x0070: 76 E9 DB 61 2C 62 63 3C 7D E0 F4 08 E0 44 96 03 v..a,bc<}....D..
0x0080: 72 72 16 0C 87 B9 BC FF 08 52 C1 41 22 59 D7 B9 rr.......R.A"Y..
0x0090: 8E 4B 77 DE B8 11 AE AF B2 CB 8D 01 92 E8 26 4A .Kw...........&J
0x00A0: 8C 24 00 8E C3 07 36 7F 84 9F 08 AF 2B 83 F8 13 .$....6.....+...
0x00B0: 1F 61 93 A8 2E 9D 5E 11 A1 DE CF 5E CF 1A 69 1B .a....^....^..i.
0x00C0: 24 F9 A8 B1 CF C7 6C 08 69 ED BF 75 0A 46 C6 63 $.....l.i..u.F.c
0x00D0: CF D2 29 5B 2D 25 C1 44 0E 3F 4C 40 8D 30 75 74 ..)[-%.D.?L@.0ut
0x00E0: A4 C3 06 90 45 65 AC 73 0C C8 CD 4E 0E 22 DD C3 ....Ee.s...N."..
0x00F0: 37 48 FD 8B E6 77 02 9C 76 84 3F E9 7C 0E 9F 28 7H...w..v.?.|..(
0x0100: 06 C1 07 B8 88 4D 22 F2 D0 EF EA B4 37 40 F4 6D .....M".....7@.m
0x0110: F8 79 47 25 85 AC 12 BB 92 94 0E 66 D9 2C 88 53 .yG%.......f.,.S
0x0120: F7 25 D7 DE 44 BF FF F2 54 4F 5B EF AB 6E E1 A0 .%..D...TO[..n..
0x0130: 38 BB DD 36 BF 5B 26 65 58 F8 8A 8..6.[&eX..
```

What is the web client's port number?

A. 443

B. 1244

C. 64

D. 080091

Answers to Review Questions

1. B. An NIDS performs attack recognition for an entire network.
2. C. Covert_TCP passes through a firewall by sending one byte at a time of a file in the IP header.
3. D. Sobek is a honeypot-detection tool.
4. A. A honeypot is a system designed to attract and identify hackers.
5. A. ADMmutate is a tool used to modify an attack script to bypass an IDS's signature detection.
6. B. A reverse WWW shell occurs when a compromised web client makes a connection back to a hacker's computer and is able to pass through a firewall.
7. A. The hacker's system, which is acting as a web server, uses port 80.
8. A. Use the command `snort –l c:\snort\log –c C:\snort\etc\snort.conf –A console` to install and run the Snort program.
9. B. Snort is a sniffer, HIDS, and traffic-logging tool.
10. A, B. Signature analysis and anomaly detection are the ways an IDS detects instruction attempts.
11. D. The destination port 443 indicates the traffic destination is an HTTPS server.
12. D. The `HOME_NET` variable is used in a `snort.conf` file to identify the local network.
13. A. The rule location is identified by the `RULE_PATH` variable in a `snort.conf` file.
14. D. Rule Action, Protocol, Source Address, and Destination Address are all included in a Snort rule header. `HOME_NET` is the variable to define the Internal Network in the `snort.conf` file.
15. C. This Snort option associates a high priority to this alert by giving it an *attack class* of `attempted-admin`.
16. A, C. Snort rules and the Snort Engine need to be installed separately during installation of Snort.
17. A. An attack *signature* is a pattern used to identify either a single packet or a series of packets that, when combined, execute an attack.
18. B. Tunneling is a method used to defeat an IDS signature match.
19. A. ***A**** indicates the ACK flag is set.
20. B. The destination address is 192.168.1.4:1244 and 1244 indicates the client port number. The source port of 443 indicates an HTTPS server.