UNIT - V

UNIT - V: Wireless Network sniffers& Penetration Test

Using Wireless Sniffers to Locate SSIDs, MAC Filters and MAC Spoofing, Rogue Access Points Intrusion detection systems, Types of IDS and Evasion Techniques, Firewall Types and Honeypot Evasion Techniques. Cryptography Attacks, Performing a Penetration Test

Using Wireless Sniffers to Locate SSIDs

A common attack on a WLAN involves eavesdropping or sniffing. This is an easy attack to perform and usually occurs at hotspots or with any default installation access point (AP), because packets are generally sent unencrypted across the WLAN. Passwords for network access protocols such as FTP, POP3, and SMTP can be captured in clear text (unencrypted) by a hacker on an unencrypted WLAN.

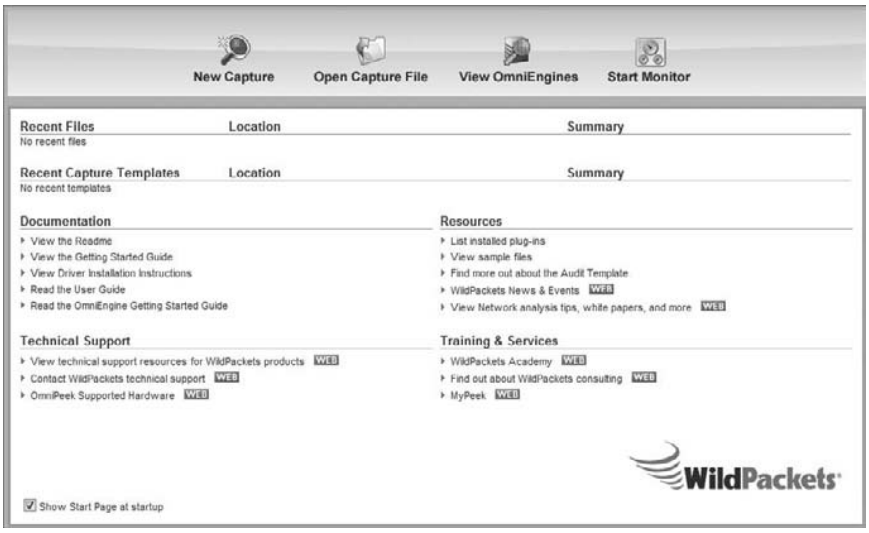
The *Service Set Identifier (SSID)* is the name of the WLAN and can be located in beacon frames and probe response frames. If two wireless networks are physically close, the SSIDs are used to identify and differentiate the respective networks. The SSID is usually sent in the clear in a beacon frame as well as other frames, such as probe response frames. Most APs allow the WLAN administrator to hide the SSID. However, this isn't a robust security mechanism because some tools can read the SSID from other packets, such as probe requests and other client-side packets.

Exercise 10.1 walks you through installing and using a WLAN sniffer tool called Omnipeek.

ExErCisE 10.1

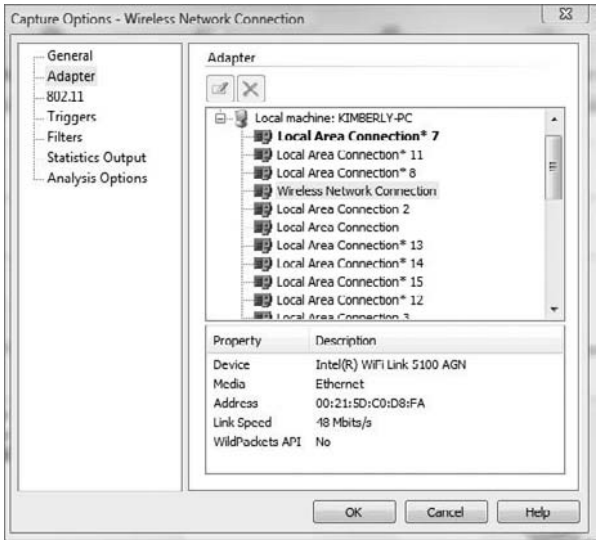
installing and using a WlaN sniffer tool

1. Download a trial version of Omnipeek from www.wildpackets.com. You will need to have a wireless LAN adapter that is supported by Omnipeek in promiscuous mode for Omnipeek to properly capture all the traffic on a wireless LAN. Check for the sup- ported wireless LAN adapters and supporting drivers from www.wildpackets.com.
2. Start a new capture by clicking the New Capture button on the Omnipeek start screen.

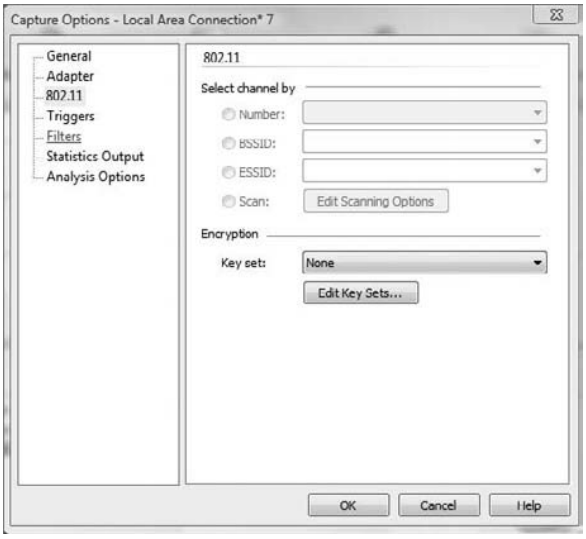


3. Select the wireless adapter from the capture options.

   Note: On the Adapter tab, the WildPackets API must list a description of Yes or the adapter will not work properly in Omnipeek, as shown here:



4. Click the 802.11 tab and choose initially to scan all channels. Later, once you have identified a specific WLAN to monitor, you can choose to only capture traffic on that one channel.



ExErCisE 10.1 ***(continued)***

5. Click OK to start the capture. The capture window will show frames being captured. Double-click a frame to see more detail.
6. Click the stop capture button to stop capturing. Select the Display filter drop down button (it looks like a funnel) from the toolbar just above the frames. Select POP from the filter drop down list. Only POP email frames will be displayed. You can use a dis- play filter to show only certain types of frames. POP, SMTP, FTP, TELNET, and HTTP frames all carry clear text data. Passwords and other information can be gathered from those frames.
7. To find Access Points (AP) and Stations that are connected, click on the WLAN menu on the left side of the screen. The APs BSSID, STA MAC, Channel, and SSID can all be located on the WLAN screen of Omnipeek. APs not broadcasting the SSID will show 0x00 for the SSID until a station connects and Omnipeek can determine the SSID from the probe frames. Once Omnipeek can determine the SSID, it will be displayed on the WLAN screen.

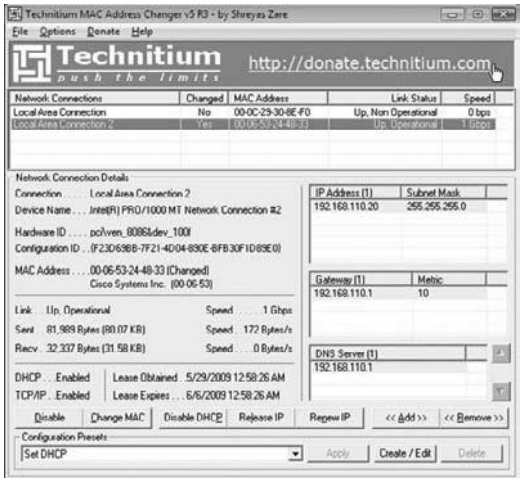
MAC Filters and MAC Spoofing

An early security solution in WLAN technology used MAC address filters: a network administrator entered a list of valid MAC addresses for the systems allowed to associate with the AP. MAC filters are cumbersome to configure and aren't scalable for an enterprise network because they must be configured on each AP. MAC spoofing is easy to perform (as you'll see in Exercise 10.2) and negates the effort required to implement MAC filters. A hacker can identify a valid MAC address because the MAC headers are never encrypted.

ExErCisE 10.2

MAC address spoofing

1. Download and install TMAC from www.technitium.com.
2. Select the wireless adapter from the list of network connections in TMAC. Click the Change MAC button.



3. Type **00:11:22:33:44:55** as the MAC address; click the Change Now button and confirm the changes to be made to the MAC address.
4. Open a command prompt and type **IPCONFIG /ALL** to confirm the MAC address of the wireless adapter has been changed to 00:11:22:33:44:55.
5. To restore the original MAC address of the network adapter, select the adapter within TMAC, click the Change MAC button, and click the Original MAC button.
6. Configure an access point to allow only the MAC address 00:11:22:33:44:55 to connect to the WLAN. (This step will vary depending on the type of access point—refer to the user guide for your access point to configure the MAC address filtering.)
7. Test the wireless client connecting using the original MAC address. The client should not connect to the AP with the MAC filtering applied. Change the MAC to 00:11:22:33:44:55 using TMAC and attempt to connect again to the AP. It should be able to connect to the AP using the Spoofed MAC address.

\

Hacking tool

SMAC is a MAC spoofing tool that a hacker can use to spoof a valid user's address and gain access to the network.

Rogue Access Points

Rogue access points are WLAN access points that aren't authorized to connect to a net- work. Rogue APs open a wireless hole into the network. A hacker can plant a rogue AP, or an employee may unknowingly create a security hole by plugging an access point into the network. The resulting rogue AP can be used by anyone who can connect to the AP, including a hacker, giving them access to the wired LAN. This is why it's critical for organizations to scan for rogue access points. Even organizations that have a "no wireless" policy need to perform wireless scanning to ensure no rogue APs are connected to the network.

Rogue APs are probably the most dangerous wireless threat that exists because they give a potential hacker direct access to the wired LAN. Clients connecting to rogue access points will usually receive an IP address directly from the network or from the AP and then the traffic is bridged directly on the wired LAN. From there a hacker can perform scanning, enumeration, and system hacking against targets on the wired LAN. Countermeasures to detect and remove rogue access points exist and should be implemented by all organizations.

Many enterprise WLAN controller based management solutions have the ability to perform rogue access point detection. These controller-based solutions include the ability to monitor the air using either access points or sensors/monitors, or both. Access points by nature must remain on a channel while clients are connected in order to service those clients, whereas sensors and monitors are able to continually scan the air on all channels in the frequency band to capture possible rogue access point wireless transmissions. These wireless MAC addresses are compared to addresses received on the wire to determine if
the AP is connected to the same LAN as the wireless intrusion detection system (WIDS) or wireless intrusion prevention system (WIPS). Some WIPSs can also keep clients from connecting to rogue access points by sending spoofed deauthentication frames to any client attempting to connect to the rogue AP thus keeping clients from sending data through the rogue AP. Overlay WIDS/WIPS systems can also be helpful in detecting rogue access points by triangulating the position of the rogue AP.

Enterprise WLAN WIPS and overlay WIPS are only temporary detection and containment options. The primary goal should be to locate the rogue AP and remove it from the network.

Evil Twin or AP Masquerading

Hackers can use a software-based AP to create an AP that looks like a real Access Point. This is known as the Evil Twin attack or AP Masquerading.

Intrusion detection systems

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.

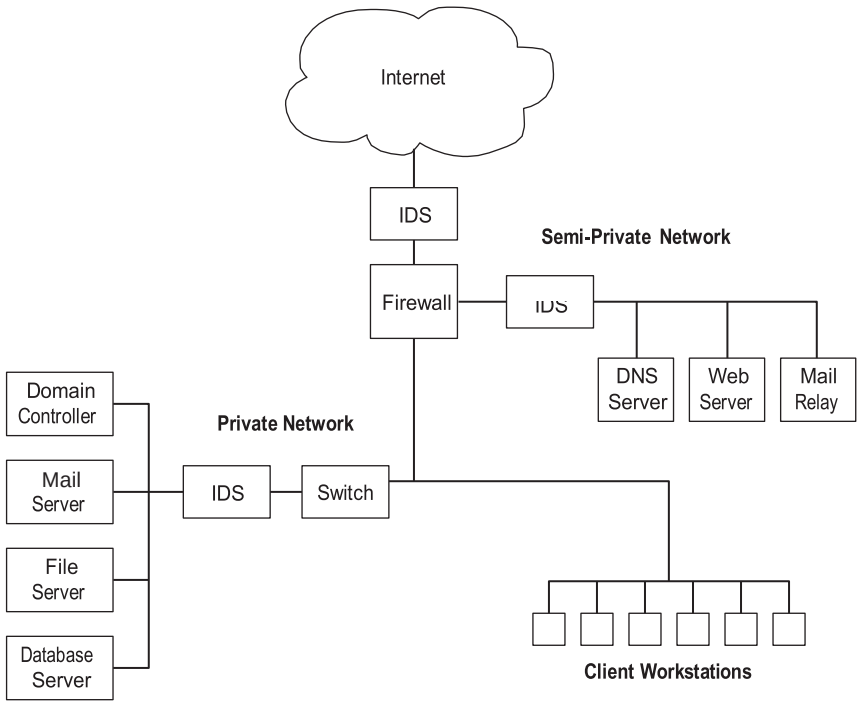
There are two main types of IDS:

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

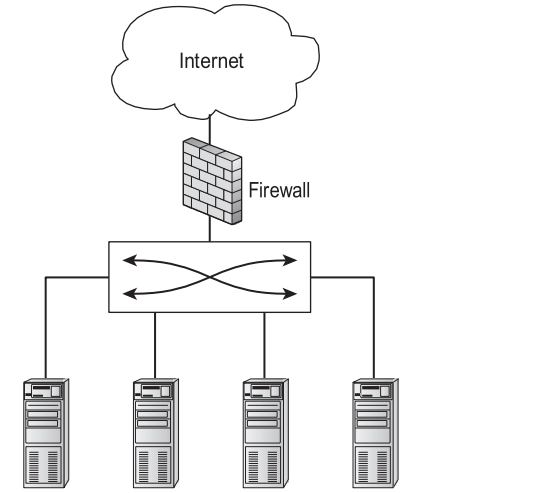
Finally, in the following line Win is the window size and the TCP length is 2000:
Win: 0x3FE1 TcpLen: 2000

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

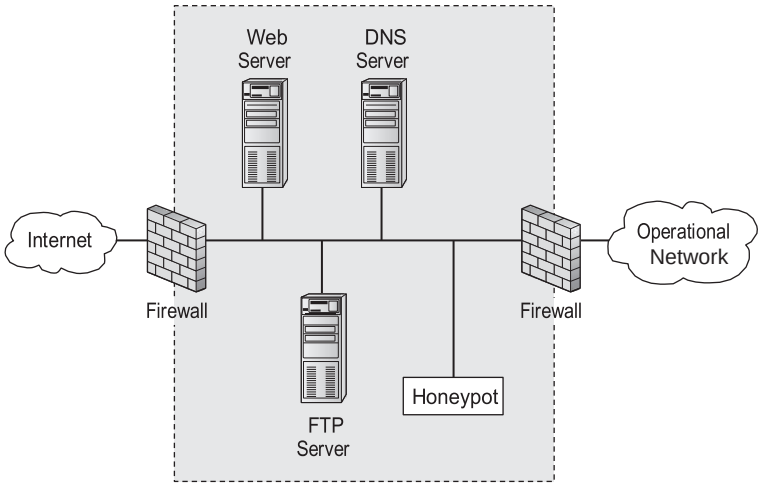
A *firewall* is a software program or hardware appliance that allows or denies access to a network and follows rules set by an administrator to direct where packets are allowed to go on the network. A *perimeter hardware firewall* appliance (Figure 13.2) is set up either at the network edge where a trusted 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



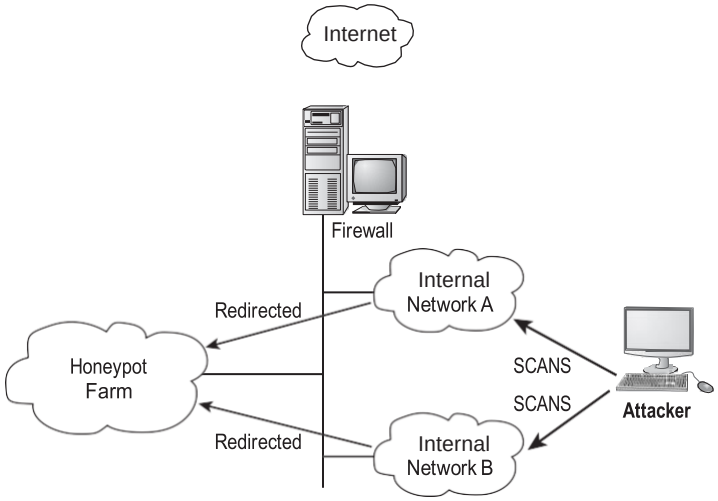
A *honeypot* (Figure 13.3) is a decoy box residing inside your network demilitarized zone (DMZ), set up by a security professional to trap or aid in locating hackers, or to draw them away from the real target system.

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 Scenarlo

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 scan- ning 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 hack- ers. 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.

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.

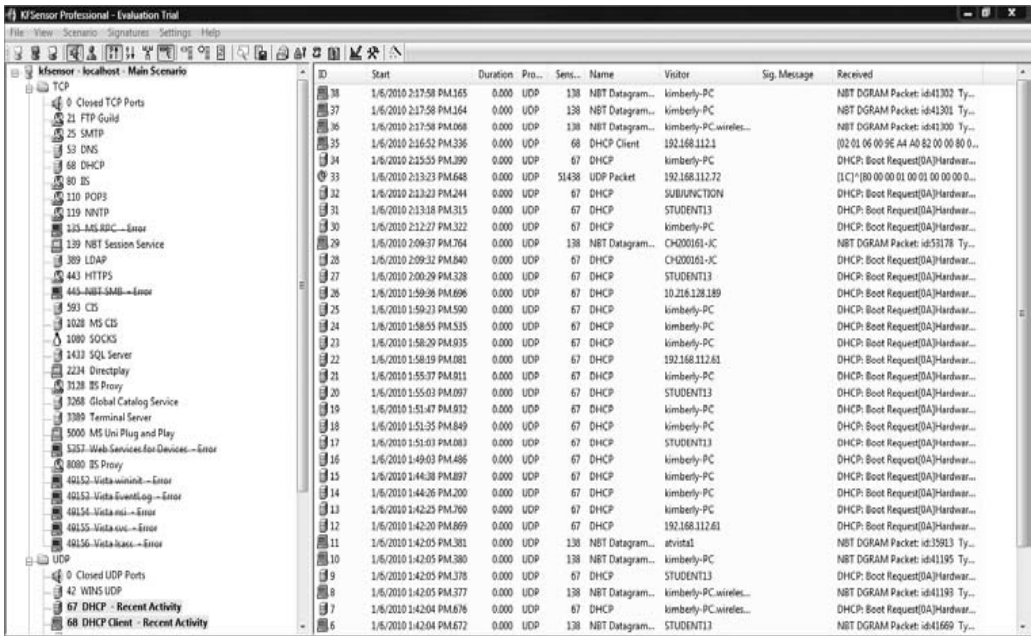
5. Type your email address in the Send To and Send From fields to receive email alerts from KFSensor.

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.

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



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

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.

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*.

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 tar- get 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.

Cryptography Attacks

Cryptographic attacks are methods of evading the security of a cryptographic system by finding weaknesses in the cipher, protocol, or key management. The following are cryptographic attacks that can be performed by an attacker:

Cipher Text–Only Attack This attack requires the attacker to obtain several messages encrypted using the same encryption algorithm. The key indicators of a cipher text–only attack are the following:

- The attacker does not have the associated plain text.
- The attacker attempts to crack the code by looking for patterns and using statistical analysis.

Known–Plain Text Attack This attack requires the attacker to have the plain text and cipher text of one or more messages. The goal is to discover the key. This attack can be used if you know a portion of the plain text of a message.

Chosen–Plain Text Attack This type of attack is carried out when an attacker has the plain text messages of their choosing encrypted. An attacker can analyze the cipher text output of the encryption.

Chosen–Cipher Text Attack This type of attack is carried out when the attacker can decrypt portions of the cipher text message of their choosing. The attacker can use the decrypted portion of the message to discover the key.

A replay attack occurs when the attacker can intercept cryptographic keys and reuse them at a later date to either encrypt or decrypt messages to which they may not have access.

A brute-force attack involves trying all possible combinations (such as keys or pass- words) until the correct solution is identified. Brute-force attacks are usually successful but require time and are usually costly.

Performing a Penetration Test

A penetration test simulates methods that intruders use to gain unauthorized access to an organization's network and systems and to compromise them.

The purpose of a penetration test is to test the security implementations and security policy of an organization. The goal is to see if the organization has implemented security measures as specified in the security policy.

A hacker whose intent is to gain unauthorized access to an organization's network is dif- ferent from a professional penetration tester. The professional tester lacks malice and intent and uses their skills to improve an organization's network security without causing a loss of service or a disruption to the business.

In this chapter, we'll look at the aspects of penetration testing (pen testing) that you must know as a CEH.

Defining Security Assessments

A *penetration tester* assesses the security posture of the organization as a whole to reveal the potential consequences of a real attacker compromising a network or application. Security assessments can be categorized as security audits, vulnerability assessments, or penetration testing. Each security assessment requires that the people conducting the assessment have different skills based on the scope of the assessment.

A *security audit* and a *vulnerability assessment* scan IP networks and hosts for known security weaknesses with tools designed to locate live systems, enumerate users, and identify operating systems and applications, looking for common security configuration mistakes and vulnerabilities.

A vulnerability or security assessment only identifies the potential vulnerabilities whereas a *pen test* tries to gain access to the network. An example of a security assessment is looking at a door and thinking if that door is unlocked it could allow someone to gain unauthorized access, whereas a pen test tries to open the door to see where it leads. A pen test is usually a better indication of the weaknesses of the network or systems but is more invasive and therefore has more potential to cause disruption to network service.

Penetration Testing

There are two types of security assessments: external and internal assessments. An *external assessment* tests and analyzes publicly available information, conducts network scanning and enumeration, and runs exploits from outside the network perimeter, usually via the Internet. An *internal assessment* is performed on the network from within the organization, with the tester acting either as an employee with some access to the network or as a black hat with no knowledge of the environment.

A black-hat penetration test usually involves a higher risk of encountering unexpected problems. The team is advised to make contingency plans in order to effectively utilize time and resources.

You can outsource your penetration test if you don't have qualified or experienced testers or if you're required to perform a specific assessment to meet audit requirements, such as the Health Insurance Portability and Accountability Act (HIPAA).

An organization employing an assessment term must specify the scope of the assessment, including what is to be tested and what is not to be tested. For example, a pen test may be a targeted test limited to the first 10 systems in a demilitarized zone (DMZ) or a comprehensive assessment uncovering as many vulnerabilities as possible. In the scope of work, a service-level agreement (SLA) should be defined to determine any actions that will be taken in the event of a serious service disruption.

Other terms for engaging an assessment team can specify a desired code of conduct, the procedures to be followed, and the interaction or lack of interaction between the organization and the testing team.

A security assessment or pen test can be performed manually with several tools, usually freeware or shareware, though the test may also include sophisticated fee-based software. A different approach is to use more expensive automated tools. Assessing the security posture of your organization using a manual test is sometimes a better option than just using an automated tool based on a standard template. The company can benefit from the expertise of an experienced professional who analyzes the information. While the automated approach may be faster and

easier, something may be missed during the audit. However, a manual approach requires planning, scheduling, and diligent documentation.

The only difference between true "hacking" and pen testing is permission. It is critical that a person performing a penetration test get written consent to perform the pen testing.

Penetration Testing Steps

Penetration testing includes three phases:

Preattack phase

Attack phase

Postattack phase

The *preattack phase* involves reconnaissance or data gathering. This is the first step for a pen tester. Gathering data from Whois, DNS, and network scanning can help you map a target network and provide valuable information regarding the operating system and applications running on the systems. The pen test involves locating the IP block and using Whois domain name lookup to find personnel contact information, as well as enumerating information about hosts. This information can then be used to create a detailed network diagram and identify targets. You should also test network filtering devices to look for legitimate traffic, stress-test proxy servers, and check for default installation of firewalls
to ensure that default users IDs, passwords, and guest passwords have been disabled or changed and no remote login is allowed.

Next is the *attack phase*, and during this phase tools can range from exploitive to responsive. They're used by professional hackers to monitor and test the security of systems and the network. These activities include but aren't limited to the following:

Penetrating the Perimeter This activity includes looking at error reports, checking access control lists by forging responses with crafted packets, and evaluating protocol filtering rules by using various protocols such as SSH, FTP, and telnet. The tester should also test for buffer overflows, SQL injections, bad input validation, output sanitization, and DoS attacks. In addition to performing software testing, you should allocate time to test internal web applications and wireless configurations, because the insider threat is the greatest security threat today.

Acquiring the Target This set of activities is more intrusive and challenging than a vulnerability scan or audit. You can use an automated exploit tool like CORE IMPACT or attempt to access the system through legitimate information obtained from social engineering. This activity also includes testing the enforcement of the security policy, or using password cracking and privilege escalation tools to gain greater access to protected resources.

Escalating Privileges Once a user account has been acquired, the tester can attempt to give the user account more privileges or rights to systems on the network. Many hacking tools are able to exploit a vulnerability in a system and create a new user account with administrator privileges.

Executing, Implanting, and Retracting This is the final phase of testing. Your hacking skills are challenged by escalating privileges on a system or network while not disrupting business processes. *Leaving a mark* can show where you were able to gain greater access to protected resources. Many companies don't want you to leave marks or execute arbitrary code, and such limitations are identified and agreed upon prior to starting your test.

The *postattack* phase involves restoring the system to normal pretest configurations, which includes removing files, cleaning Registry entries if vulnerabilities were created, and removing shares and connections.

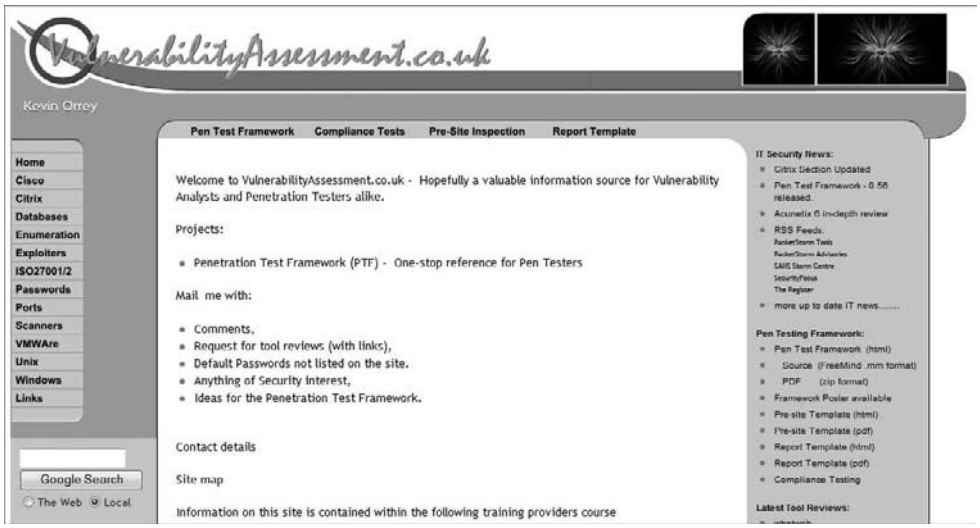
Finally, you analyze all the results and create two copies of the security assessment reports, one for your records and one for management. These reports include your objectives, your observations, all activities undertaken, and the results of test activities, and may recommend fixes for vulnerabilities.

Exercise 15.1 shows a framework for a comprehensive penetration test.

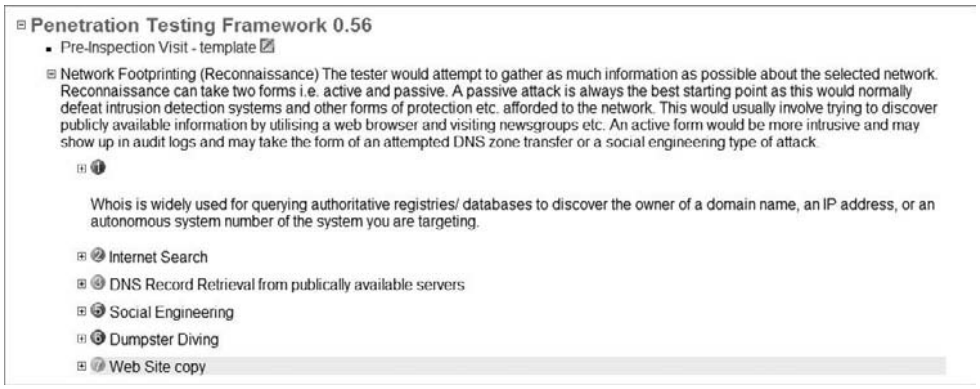
ExErCisE 15.1

viewing a Pen Testing Framework of Tools

1. Open a web browser to www.vulnerabilityassessment.co.uk.



2. Click the Pen Test Framework link near the top.
3. Expand the Network Footprinting section and view the subheadings.



4. Continue down the major heading, expanding each of the subheadings for the pen test framework. You can use this list to locate all the tools necessary in each step of the pen testing process.

The Pen Test Legal Framework

A penetration tester must be aware of the legal ramifications of hacking a network, even in an ethical manner. We explored the laws applicable to hacking in Chapter 1. The documents that an ethical hacker performing a penetration test must have signed with the client are as follows:

Scope of work, to identify what is to be tested

Nondisclosure agreement, in case the tester sees confidential information

Liability release, releasing the ethical hacker from any actions or disruption of service caused by the pen test

Automated Penetration Testing Tools

A 2006 survey of the hackers mailing list created a top-10 list of vulnerability scanning tools; more than 3,000 people responded. Fyodor (http://insecure.org/fyodor/), who created the list, says, "Anyone in the security field would be well advised to go over the list and investigate tools they are unfamiliar with." The following should be considered the top pen testing tools in a hacker's toolkit:

Nessus This freeware network vulnerability scanner has more than 11,000 plug-ins avail- able. Nessus includes remote and local security checks, a client/server architecture with a GTK graphical interface, and an embedded scripting language for writing your own plug- ins or understanding the existing ones.

GFI LANguard This is a commercial network security scanner for Windows. GFI LANguard scans IP networks to detect what machines are running. It can determine the host operating system, what applications are running, what Windows service packs are installed, whether any security patches are missing, and more.

Retina This is a commercial vulnerability assessment scanner from eEye. Like Nessus, Retina scans all the hosts on a network and reports on any vulnerabilities found.

CORE IMPACT CORE IMPACT is an automated pen testing product that is widely con- sidered to be the most powerful exploitation tool available (it's also very costly). It has a large, regularly updated database of professional exploits. Among its features, it can exploit one machine and then establish an encrypted tunnel through that machine to reach and exploit other machines.

ISS Internet Scanner This is an application-level vulnerability assessment. Internet Scanner can identify more than 1,300 types of networked devices on your network, including desk- tops, servers, routers/switches, firewalls, security devices, and application routers.

X-Scan X-Scan is a general multithreaded plug-in-supported network vulnerability scanner. It can detect service types, remote operating system types and versions, and weak usernames and passwords.

SARA Security Auditor's Research Assistant (SARA) is a vulnerability assessment tool derived from the System Administrator Tool for Analyzing Networks (SATAN) scanner. Updates are typically released twice a month.

Qualys Guard This is a web-based vulnerability scanner. Users can securely access QualysGuard through an easy-to-use web interface. It features more than 5,000 vulnerability checks, as well as an inference-based scanning engine.

SAINT Security Administrator's Integrated Network Tool (SAINT) is a commercial vulnerability assessment tool.

MBSA Microsoft Baseline Security Analyzer (MBSA) is built on the Windows Update Agent and Microsoft Update infrastructure. It ensures consistency with other Microsoft products and, on average, scans more than 3 million computers each week.

In addition to this list, you should be familiar with the following vulnerability exploitation tools:

Metasploit Framework This is an open source software product used to develop, test, and use exploit code.

Canvas Canvas is a commercial vulnerability exploitation tool. It includes more than 150 exploits.

Pen Test Deliverables

The main deliverable at the end of a penetration test is the pen testing report. The report should include the following:

A list of your findings, in order of highest risk

An analysis of your findings

A conclusion or explanation of your findings

Remediation measures for your findings
Log files from tools that provide supporting evidence of your findings
An executive summary of the organization's security posture
The name of the tester and the date testing occurred
Any positive findings or good security implementations