Catalog

5.1. Social Engineering

Social engineering is a nontechnical method of breaking into a system or network. It's the process of deceiving users of a system and convincing them to perform acts useful to the hacker, such as giving out information that can be used to defeat or bypass security mechanisms. Social engineering is important to understand because hackers can use it to attack the human element of a system and circumvent technical security measures. This method can be used to gather information before or during an attack.

A social engineer commonly uses the telephone or Internet to trick people into revealing sensitive information or to get them to do something that is against the security policies of the organization. By this method, social engineers exploit the natural tendency of a person to trust their word, rather than exploiting computer security holes. It's generally agreed that users are the weak link in security; this principle is what makes social engineering possible.

The following is an example of social engineering recounted by Kapil Raina, currently a security expert at VeriSign, based on an actual workplace experience with a previous employer:

> One morning a few years back, a group of strangers walked into a large shipping firm and walked out with access to the firm's entire corporate network. How did they do it? By obtaining small amounts of access, bit by bit, from a number of different employees in that firm. First, they did research about the company for two days before even attempting to set foot on the premises. For example, they learned key employees' names by calling HR. Next, they pretended to lose their key to the front door, and a man let them in. Then they "lost" their identity badges when entering the third floor secured area, smiled, and a friendly employee opened the door for them.
>
> The strangers knew the CFO was out of town, so they were able to enter his office and obtain financial data off his unlocked computer. They dug through the corporate trash, finding all kinds of useful documents. They asked a janitor for a garbage pail in which to place their contents and carried all of this data out of the building in their hands. The strangers had studied the CFO's voice, so they were

able to phone, pretending to be the CFO, in a rush, desperately in need of his network password. From there, they used regular technical hacking tools to gain super-user access into the system.

In this case, the strangers were network consultants performing a security audit for the CFO without any other employees' knowledge. They were never given any privileged information from the CFO but were able to obtain all the access they wanted through social engineering.

The most dangerous part of social engineering is that companies with authentication processes, firewalls, virtual private networks, and network-monitoring software are still wide open to attacks, because social engineering doesn't assault the security measures directly. Instead, a social-engineering attack bypasses the security measures and goes after the human element in an organization.

The Art of Manipulation

Social engineering includes the acquisition of sensitive information or inappropriate access privileges by an outsider, based on the building of inappropriate trust relationships. The goal of a social engineer is to trick someone into providing valuable information or access to that information. Social engineering preys on qualities of human nature, such as the desire to be helpful, the tendency to trust people, and the fear of getting in trouble. Hackers who are able to blend in and appear to be a part of the organization are the most successful at social-engineering attacks. This ability to blend in is commonly referred to as the *art of manipulation*.

People are usually the weakest link in the security chain. A successful defense depends on having good policies in place and teaching employees to follow the policies. Social engineering is the hardest form of attack to defend against because a company can't protect itself with hardware or software alone.

Types of Social Engineering-Attacks

Social engineering can be broken into two common types:

Human-Based Human-based social engineering refers to person-to-person interaction to retrieve the desired information. An example is calling the help desk and trying to find out a password.

Computer-Based Computer-based social engineering refers to having computer software that attempts to retrieve the desired information. An example is sending a user an email and asking them to reenter a password in a web page to confirm it. This social-engineering attack is also known as *phishing*.

Human-Based Social Engineering

Human-based social engineering techniques can be broadly categorized as follows:

Impersonating an Employee or Valid User In this type of social-engineering attack, the hacker pretends to be an employee or valid user on the system. A hacker can gain physical access by pretending to be a janitor, employee, or contractor. Once inside the facility, the hacker gathers information from trashcans, desktops, or computer systems.

Posing as an Important User In this type of attack, the hacker pretends to be an important user such as an executive or high-level manager who needs immediate assistance to gain access to a computer system or files. The hacker uses intimidation so that a lower-level employee such as a help desk worker will assist them in gaining access to the system. Most low-level employees won’t question someone who appears to be in a position of authority.

Using a Third Person Using the third-person approach, a hacker pretends to have permission from an authorized source to use a system. This attack is especially effective if the supposed authorized source is on vacation or can’t be contacted for verification.

Calling Technical Support Calling tech support for assistance is a classic social-engineering technique. Help desk and technical support personnel are trained to help users, which makes them good prey for social-engineering attacks.

Shoulder Surfing Shoulder surfing is a technique of gathering passwords by watching over a person's shoulder while they log in to the system. A hacker can watch a valid user log in and then use that password to gain access to the system.

Dumpster Diving Dumpster diving involves looking in the trash for information written on pieces of paper or computer printouts. The hacker can often find passwords, filenames, or other pieces of confidential information.

A more advanced method of gaining illicit information is known as *reverse social engineering*. Using this technique, a hacker creates a persona that appears to be in a position of authority so that employees ask the hacker for information, rather than the other way around. For example, a hacker can impersonate a help desk employee and get the user to give them information such as a password.

Computer-Based Social Engineering

Computer-based social-engineering attacks can include the following:

- ✓ Email attachments
- ✓ Fake websites
- ✓ Pop-up windows

Insider Attacks

If a hacker can't find any other way to hack an organization, the next best option is to infiltrate the organization by getting hired as an employee or finding a disgruntled employee to assist in the attack. Insider attacks can be powerful because employees have physical access and are able to move freely about the organization. An example might be someone posing as a delivery person by wearing a uniform and gaining access to a delivery room or loading dock. Another possibility is someone posing as a member of the cleaning crew who has access to the inside of the building and is usually able to move about the offices. As a last resort, a hacker might bribe or otherwise coerce an employee to participate in the attack by providing information such as passwords.

Identity Theft

A hacker can pose as an employee or steal the employee's identity to perpetrate an attack. Information gathered in dumpster diving or shoulder surfing in combination with creating fake ID badges can gain the hacker entry into an organization. Creating a persona that can enter the building unchallenged is the goal of identity theft.

Phishing Attacks

Phishing involves sending an email, usually posing as a bank, credit card company, or other financial organization. The email requests that the recipient confirm banking information or reset passwords or PINs. The user clicks the link in the email and is redirected to a fake website. The hacker is then able to capture this information and use it for financial gain or to perpetrate other attacks. Emails that claim the senders have a great amount of money but need your help getting it out of the country are examples of phishing attacks. These attacks prey on the common person and are aimed at getting them to provide bank account access codes or other confidential information to the hacker.

Online Scams

Some websites that make free offers or other special deals can lure a victim to enter a username and password that may be the same as those they use to access their work system. The hacker can use this valid username and password once the user enters the information in the website form.

Mail attachments can be used to send malicious code to a victim's system, which could automatically execute something like a software keylogger to capture passwords. Viruses, Trojans, and worms can be included in cleverly crafted emails to entice a victim to open the attachment. Mail attachments are considered a computer-based social-engineering attack.

Here is an example of an email that which tries to convince the receiver to open an unsafe attachment:

Mail server report.

Our firewall determined the e-mails containing worm copies are being sent from your computer.

Nowadays it happens from many computers, because this is a new virus type (Network Worms).

URL Obfuscation

The URL (uniform resource locator) is commonly used in the address bar of a web browser to access a particular website. In lay terms, it is the website address. URL obfuscation consists of hiding a fake URL in what appear to be a legitimate website address. For example, a website of 204.13.144.2/Citibank may appear to be a legitimate web address for Citibank but in fact is not. URL obfuscation is used in phishing attacks and some online scams to make the scam seem more legitimate. A website address may be seen as an actual financial institution name or logo, but the link leads to a fake website or IP address. When users click the link, they're redirected to the hacker's site.

Social-Engineering Countermeasures

Knowing how to combat social engineering is critical for any certified ethical hacker. There are a number of ways to do this.

Documented and enforced security policies and security awareness programs are the most critical component in any information security program. Good policies and procedures aren't effective if they aren't taught and reinforced to employees. The policies need to be communicated to employees to emphasize their importance and then enforced by management. After receiving security awareness training, employees will be committed to supporting the security policies of the organization.

The corporate security policy should address how and when accounts are set up and terminated, how often passwords are changed, who can access what information, and how policy violations are to be handled. Also, the policy should spell out help desk procedures for the previous tasks as well as a process for identifying employees—for example, using an employee number or other information to validate a password change. The destruction of paper documents and physical access restrictions are additional areas the security policy should address. Lastly, the policy should address technical areas, such as use of modems and virus control.

One of the advantages of a strong security policy is that it removes the responsibility of employees to make judgment calls regarding a hacker's request. If the requested action is prohibited by the policy, the employee has guidelines for denying it.

The most important countermeasure for social engineering is employee education. All employees should be trained on how to keep confidential data safe. Management teams are involved in the creation and implementation of the security policy so that they fully understand it and support it throughout the organization. The company security awareness policy should require all new employees to go through a security orientation. Annual classes should be required to provide refreshers and updated information for employees.

Another way to increase involvement is through a monthly newsletter with security awareness articles.

5.2. Denial of Service

A DoS attack is an attempt by a hacker to flood a user's or an organization's system. As a

CEH, you need to be familiar with the types of DoS attacks and should understand how DoS and DDoS attacks work. You should also be familiar with robots (BOTs) and robot networks (BOTNETs), as well as smurf attacks and SYN flooding. Finally, as a CEH, you need to be familiar with various DoS and DDoS countermeasures.

There are two main categories of DoS attacks:

- Attacks sent by a single system to a single target (simple DoS)
- Attacks sent by many systems to a single target (distributed denial of service, or DDoS).

The goal of DoS isn't to gain unauthorized access to machines or data, but to prevent legitimate users of a service from using it. A DoS attack may do the following:

❖ Flood a network with traffic, thereby preventing legitimate network traffic.

❖ Disrupt connections between two machines, thereby preventing access to a service.

❖ Prevent a particular individual from accessing a service.

❖ Disrupt service to a specific system or person.

Different tools use different types of traffic to flood a victim, but the result is the same: a service on the system or the entire system is unavailable to a user because it's kept busy trying to respond to an exorbitant number of requests.

A DoS attack is usually an attack of last resort. It's considered an unsophisticated attack because it doesn't gain the hacker access to any information but rather annoys the target and interrupts their service. DoS attacks can be destructive and have a substantial impact when sent from multiple systems at the same time (DDoS attacks).

DDoS attacks can be perpetrated by BOTs and BOTNETs, which are compromised systems that an attacker uses to launch the attack against the end victim. The system or network that has been compromised is a secondary victim, whereas the DoS and DDoS attacks flood the primary victim or target.

How DDoS Attacks Work

DDoS is an advanced version of the DoS attack. Like DoS, DDoS tries to deny access to services running on a system by sending packets to the destination system in a way that the destination system can't handle. The key of a DDoS attack is that it relays attacks from many different hosts (which must first be compromised), rather than from a single host like DoS. DDoS is a large-scale, coordinated attack on a victim system.

The services under attack are those of the primary victim; the compromised systems used to launch the attack are secondary victims. These compromised systems, which send the DDoS to the primary victim, are sometimes called *zombies* or *BOTs*. They're usually compromised through another attack and then used to launch an attack on the primary victim at a certain time or under certain conditions. It can be difficult to track the source of the attacks because they originate from several IP addresses.

Normally, DDoS consists of three parts:

- Master/handler
- Slave/secondary victim/zombie/agent/BOT/BOTNET
- Victim/primary victim

The *master* is the attack launcher. A *slave* is a host that is compromised by and controlled by the master. The *victim* is the target system. The master directs the slaves to launch the attack on the victim system.

DDoS is done in two phases. In the intrusion phase, the hacker compromises weak systems in different networks around the world and installs DDoS tools on those compromised slave systems. In the DDoS attack phase, the slave systems are triggered to cause them to attack the primary victim.

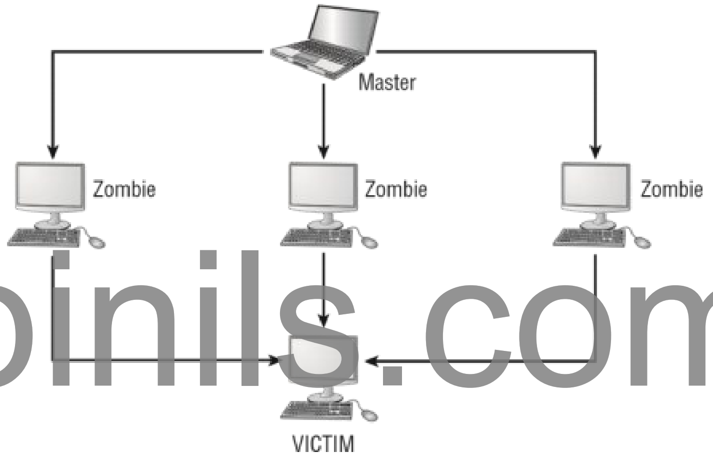


Fig: Bots or Zombie systems

How BOTs/BOTNETs Work

A BOT is short for *web robot* and is an automated software program that behaves intelligently. Spammers often use BOTs to automate the posting of spam messages on newsgroups or the sending of emails. BOTs can also be used as remote attack tools. Most often, BOTs are web software agents that interface with web pages. For example, web crawlers (spiders) are web robots that gather web page information.

The most dangerous BOTs are those that covertly install themselves on users' computers for malicious purposes.

Some BOTs communicate with other users of Internet-based services via instant messaging, Internet Relay Chat (IRC), or another web interface. These BOTs allow IRQ users to ask questions in plain English and then formulate a proper response. Such BOTs can often handle many tasks, including reporting weather; providing zip code information; listing sports scores; converting units of measure, such as currency; and so on.

A BOTNET is a group of BOT systems. BOTNETs serve various purposes, including DDoS attacks; creation or misuse of Simple Mail Transfer Protocol (SMTP) mail relays for spam; Internet marketing fraud; and the theft of application serial numbers, login IDs, and financial information such as credit card numbers. Generally a BOTNET refers to a group of compromised systems running a BOT for the purpose of launching a coordinated DDoS attack. See Figure 7.3.

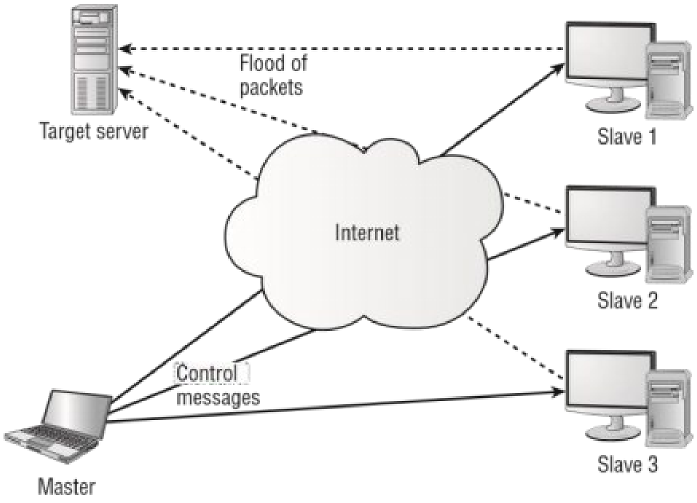


Fig: Anatomy of a Distributed DoS Attack

Smurf and SYN Flood Attacks

A *smurf* attack sends a large amount of ICMP Echo (ping) traffic to a broadcast IP address with the spoofed source address of a victim. Each secondary victim's host on that IP network replies to the ICMP Echo request with an Echo reply, multiplying the traffic by the number of hosts responding. On a multiaccess broadcast network, hundreds of machines might reply to each packet. This creates a magnified DoS attack of ping replies, flooding the primary victim. IRC servers are the primary victim of smurf attacks on the Internet.

A *SYN flood* attack sends TCP connection requests faster than a machine can process them. The attacker creates a random source address for each packet and sets the SYN flag to request a new connection to the server from the spoofed IP address. The victim responds to the spoofed IP address and then waits for the TCP confirmation that never arrives. Consequently, the victim's connection table fills up waiting for replies; after the table is full, all new connections are ignored. Legitimate users are ignored as well and can't access the server.

A SYN flood attack can be detected through the use of the netstat command. An example of the netstat output from a system under a SYN flood is shown in Figure 7.4.

Here are some of the methods used to prevent SYN flood attacks:

SYN Cookies SYN cookies ensure the server does not allocate system resources until a successful three-way handshake has been completed.

RST Cookies Essentially the server responds to the client SYN frame with an incorrect

SYN ACK. The client should then generate an RST packet telling the server that something is wrong. At this point, the server knows the client is valid and will now accept incoming connections from that client normally.

Micro Blocks Micro blocks prevent SYN floods by allocating only a small space in memory for the connection record. In some cases, this memory allocation is as small as 16 bytes.

Stack Tweaking This method involves changing the TCP/IP stack to prevent SYN floods. Techniques of stack tweaking include selectively dropping incoming connections or reducing the timeout when the stack will free up the memory allocated for a connection.

DoS/DDoS Countermeasures

There are several ways to detect, halt, or prevent DoS attacks. The following are common security features:

Network-Ingress Filtering All network access providers should implement networkingress filtering to stop any downstream networks from injecting packets with faked or spoofed addresses into the Internet. Although this doesn't stop an attack from occurring, it does make it much easier to track down the source of the attack and terminate the attack quickly. Most IDS, firewalls, and routers provide network-ingress filtering capabilities.

Rate-Limiting Network Traffic A number of routers on the market today have features that let you limit the amount of bandwidth some types of traffic can consume. This is sometimes referred to as *traffic shaping*.

Intrusion Detection Systems Use an intrusion detection system (IDS) to detect attackers who are communicating with slave, master, or agent machines. Doing so lets you know whether a machine in your network is being used to launch a known attack but probably won't detect new variations of these attacks or the tools that implement them. Most IDS vendors have signatures to detect Trinoo, TFN, or Stacheldraht network traffic.

Automated Network-Tracing Tools Tracing streams of packets with spoofed addresses through the network is a time-consuming task that requires the cooperation of all networks carrying the traffic and that must be completed while the attack is in progress.

Host-Auditing and Network-Auditing Tools File-scanning tools are available that attempt to detect the existence of known DDoS tool client and server binaries in a system. Network scanning tools attempt to detect the presence of DDoS agents running on hosts on your network.

5.3. Session Hijacking

Session hijacking is when a hacker takes control of a user session after the user has successfully authenticated with a server. Session hijacking involves an attack identifying the current session IDs of a client/server communication and taking over the client's session. Session hijacking is made possible by tools that perform sequence-number prediction.

Spoofing attacks are different from hijacking attacks. In a spoofing attack, the hacker performs sniffing and listens to traffic as it's passed along the network from sender to receiver. The hacker then uses the information gathered to spoof or uses an address of a legitimate system. Hijacking involves actively taking another user offline to perform the attack. The attacker relies on the legitimate user to make a connection and authenticate.

After that, the attacker takes over the session, and the valid user's session is disconnected. Session hijacking involves the following three steps to perpetuate an attack:

Tracking the Session The hacker identifies an open session and predicts the sequence number of the next packet.

Desynchronizing the Connection The hacker sends the valid user's system a TCP reset (RST) or finish (FIN) packet to cause them to close their session.

Injecting the Attacker's Packet The hacker sends the server a TCP packet with the predicted sequence number, and the server accepts it as the valid user's next packet.

Hackers can use two types of session hijacking: active and passive. The primary difference between active and passive hijacking is the hacker's level of involvement in the session. In an active attack, an attacker finds an active session and takes over the session by using tools that predict the next sequence number used in the TCP session.

In a passive attack, an attacker hijacks a session and then watches and records all the traffic that is being sent by the legitimate user. Passive session hijacking is really no more than sniffing. It gathers information such as passwords and then uses that information to authenticate as a separate session.

Sequence Prediction

TCP is a connection-oriented protocol, responsible for reassembling streams of packets into their original intended order. Every packet has to be assigned a unique session number that enables the receiving machine to reassemble the stream of packets into their original and intended order; this unique number is known as a *sequence number*. If the packets arrive out of order, as happens regularly over the Internet, then the SN is used to stream the packets correctly. As just illustrated, the system initiating a TCP session transmits a packet with the SYN bit set. This is called a *synchronize packet* and includes the client's ISN. The ISN is a pseudo-randomly generated number with over 4 billion possible combinations, yet it is statistically possible for it to repeat.

When the ACK packet is sent, each machine uses the SN from the packet being acknowledged, plus an increment. This not only properly confirms receipt of a specific packet, but also tells the sender the next expected TCP packet SN. Within the three-way handshake, the increment value is 1. In normal data communications, the increment value equals the size of the data in bytes (for example, if you transmit 45 bytes of data, the ACK responds using the incoming packet's SN plus 45).

Fig: Sequence numbers and acknowledgment during the TCP three-way handshake

Hacking tools used to perform session hijacking do sequence number prediction. To successfully perform a TCP sequence prediction attack, the hacker must sniff the traffic between two systems. Next, the hacker or the hacking tool must successfully guess the SN or locate an

ISN to calculate the next sequence number. This process can be more difficult than it sounds, because packets travel very fast.

When the hacker is unable to sniff the connection, it becomes much more difficult to guess the next SN. For this reason, most session-hijacking tools include features to permit sniffing the packets to determine the SNs.

Hackers generate packets using a spoofed IP address of the system that had a session with the target system. The hacking tools issue packets with the SNs that the target system is expecting. But the hacker's packets must arrive before the packets from the trusted system whose connection is being hijacked. This is accomplished by flooding the trusted system with packets or sending an RST packet to the trusted system so that it is unavailable to send packets to the target system.

Dangers Posed by Session Hijacking

TCP session hijacking is a dangerous attack: most systems are vulnerable to it, because they use TCP/IP as their primary communication protocol. Newer operating systems have attempted to secure themselves from session hijacking by using pseudo-random number generators to calculate the ISN, making the sequence number harder to guess. However, this security measure is ineffective if the attacker is able to sniff packets, which gives all the information required to perform this attack.

The following are reasons why it's important for a CEH to be aware of session hijacking:

- Most computers are vulnerable.
- Few countermeasures are available to adequately protect against it.
- Session hijacking attacks are simple to launch.
- Hijacking is dangerous because of the information that can be gathered during the attack.

Preventing Session Hijacking

To defend against session hijack attacks, a network should employ several defenses. The most effective protection is encryption, such as Internet Protocol Security (IPSec). This also defends against any other attack vectors that depend on sniffing. Attackers may be able to passively monitor your connection, but they won't be able to interpret the encrypted data. Other countermeasures include using encrypted applications such as Secure Shell (SSH, an encrypted telnet) and Secure Sockets Layer (SSL, for HTTPS traffic).

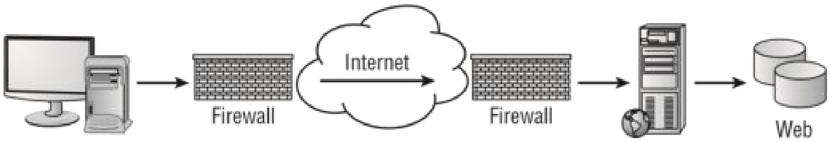
You can help prevent session hijacking by reducing the potential methods of gaining access to your network—for example, by eliminating remote access to internal systems. If the network has remote users who need to connect to carry out their duties, then use virtual private networks (VPNs) that have been secured with tunneling protocols and encryption (Layer 3 Tunneling Protocol [L3TP]/Point-to-Point Tunneling Protocol [PPTP] and IPSec).

The use of multiple safety nets is always the best countermeasure to any potential threat. Employing any one countermeasure may not be enough, but using them together to secure your enterprise will make the attack success rate minimal for anyone but the most professional and dedicated attacker. The following is a checklist of countermeasures that should be employed to prevent session hijacking:

- Use encryption.
- Use a secure protocol.
- Limit incoming connections.
- Minimize remote access.
- Have strong authentication.
- Educate your employees.
- Maintain different username and passwords for different accounts.
- Use Ethernet switches rather than hubs to prevent session hijacking attacks.

5.4. Hacking Web servers

Web servers use Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS) to allow web-based clients to connect to them and view and download files. HTTP is an Application-layer protocol in the TCP/IP stack. HTTP and HTTPS are the primary protocols used by web clients accessing web pages residing on web servers on the Internet. Hypertext Markup Language (HTML) is the language used to create web pages and allows those pages to be rendered in web browser software on web clients. The HTTP protocol operates as shown in Figure



Web Client

Web Application Serverand Database

Fig: HTTP protocol components

Types of Web Server Vulnerabilities

Web servers, like other systems, can be compromised by a hacker. The following vulnerabilities are most commonly exploited in web servers:

Misconfiguration of the Web Server Software :A common issue with using Microsoft's Internet Information Server (IIS) as a web server is the use of the default website. The permissions on the default website are open, meaning the default settings leave the site open to attack. For example, all users in the everyone group have full control to all the files in the default website directory. It is critical to edit and restrict permissions once IIS is installed on the server as the default system user, IUSR_COMPUTERNAME, is a member of the everyone group. Consequently, anyone accessing the default website will be able to access all files in the default website folder and will have dangerous permissions such as Execute and Full Control to the files.

Operating System or Application Bugs, or Flaws in Programming Code : All programs, including the OS and web server applications, should be patched or updated on a regular basis. For Windows systems, this includes security patches, hotfixes, and Windows Updates. All of these patches can be automated or manually applied to the systems once they have been tested.

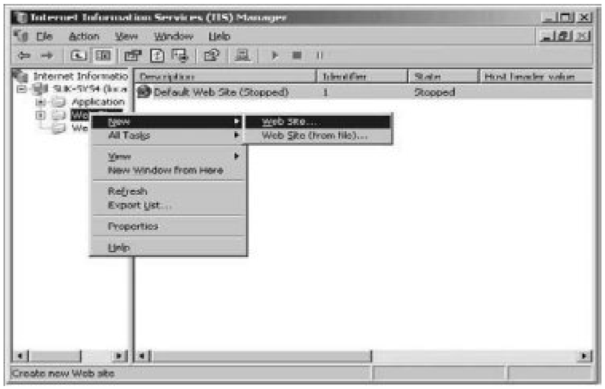
Vulnerable Default Installation : Operating system and web server software settings should not be left at their defaults when installed, and should be updated on a continuous basis.

Hackers exploit these vulnerabilities to gain access to the web server. Because web servers are usually located in a demilitarized zone (DMZ)—which is a publicly accessible area between two packet filtering devices and can be easily accessed by the organization's client systems—an exploit of a web server offers a hacker easier access to internal systems or databases.

Disabling the Default Website in Internet Information Server

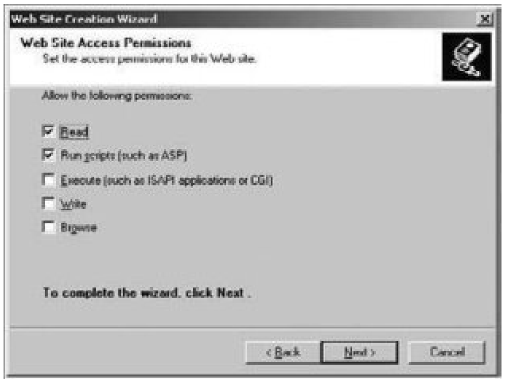
To disable the default website in IIS and add a new site, follow these steps:

1. Open IIS on your Windows Server or virtual machine (VM).
2. Select Web Sites in the left pane.
3. Right-click the default website in the right pane and select Stop from the context menu. The default website is now stopped.
4. To create a new site, right-click Web Sites in the left pane and select New Web Site.



Description	Identifier	State	Host header value
Default Web Site (Stopped)	1	Stopped	

5. The Web Site Creation Wizard launches. Within the wizard will be a screen to change permission on the website directory.



Web Site Creation Wizard

Web Site Access Permissions

Set the access permissions for this Web site.

Allow the following permissions:

☑ Read

☑ Run scripts (such as ASP)

☐ Execute (such as ISAPI applications or CGI)

☐ Write

☐ Browse

To complete the wizard, click Next .

One option is to right-click any web page and select View Source from the context menu. This command will open up a new window with the source code for the page. You can then save the text file as a document on the local machine. This approach works, but it isn't a practical way of copying all the files for a target website. An easy-to-use program called BlackWidow can make the process of copying website files much easier

Attacking a Web Server

Web servers typically listen on TCP port 80 (HTTP) and TCP port 443 (HTTPS). Because those ports must be open and available to web clients, any firewalls or packet filtering devices between the web client and web server must pass traffic destined for those ports. Web application software sits on top of the web server software and allows access to additional ports.

One of the initial information-gathering steps targeting web servers is ***banner grabbing***. Banner grabbing is an attempt to gather information about a web server such as the OS and web server software and version. Often the hacker displays their hacker name on the website's home page.

Common website attacks that enable a hacker to deface a website include the following:

- Capturing administrator credentials through man-in-the-middle attacks
- Revealing an administrator password through a brute-force attack
- Using a DNS attack to redirect users to a different web server
- Compromising an FTP or email server
- Exploiting web application bugs that result in a vulnerability

- Misconfiguring web shares
- Taking advantage of weak permissions
- Rerouting a client after a firewall or router attack
- Using SQL injection attacks (if the SQL server and web server are the same system)
- Using telnet or Secure Shell (SSH) intrusion
- Carrying out URL poisoning, which redirects the user to a different URL
- Using web server extension or remote service intrusion

Intercepting the communication between the client and the server and changing the cookie to make changing the cookie to make the server believe that there is a user with higher privileges (applies to cookie enabled security)

Hacking Internet Information Server

Windows IIS is one of the most popular web server software products. Because of the popularity and number of web servers running IIS, many attacks can be launched against IIS servers. The three most common attacks against IIS are as follows:

- ✓ Directory traversal
- ✓ Source disclosure
- ✓ Buffer overflow

A ***directory-traversal attack*** is based on the premise that web clients are limited to specific directories within the Windows files system. The initial directory access by web clients is known as the ***root directory*** on a web server. This root directory typically stores the home page usually known as Default or Index, as well as other HTML documents for the web server. Subdirectories of the root directory contain other types of files; for example, scripts may contain dynamic scripting files for the web server. The web server should allow users to access only these specific directories and subdirectories of root. However, a directory traversal attack permits access to other directories within the file system.

Windows 2000 systems running IIS are susceptible to a directory-traversal attack, also known as the Unicode exploit. The vulnerability in IIS that allows for the directory traversal/Unicode exploit occurs only in unpatched Windows 2000 systems and affects CGI scripts and Internet Server Application Programming Interface (ISAPI) extensions such as .asp. The vulnerability exists because the IIS parser was not properly interpreting Unicode, thus giving hackers system-level access.

Essentially, Unicode converts characters of any language to a universal hex code specification. However, the Unicode is interpreted twice, and the parser only scans the resulting request once (following the first interpretation). Hackers could therefore sneak file requests through IIS. For example, utilizing %c0% af instead of a slash in a relative pathname exploits the IIS vulnerability. In some cases, the request lets the hacker gain access to files that they otherwise shouldn't be able to see. The Unicode directory traversal vulnerability allows a hacker to add, change, or delete files, or upload and run code on the server. The ability to add or run files on the system enables a hacker to install a Trojan or backdoor on the system

Buffer overflow attacks are not unique to web servers and can also be launched against other types of systems. A buffer overflow involves sending more data, usually in the form of a text string, than the web server is capable of handling. The primary entry point for buffer overflows is a web form on the web server. Buffer overflows and countermeasures will be covered in detail in the next chapter.

Source disclosure attacks occur when the source code of a server application can be gathered. Source disclosure attacks can lead to a hacker identifying the application type, programming language, and other application-specific information. All this information can allow a potential hacker to identify security holes and potential exploits that can be delivered to the web server. Again, most of a hacker's time is spent gathering information about a target in order to identify the best point of entry for an exploit.

Patch-Management Techniques

Patch management plays a critical role in preventing and mitigating the risk of attack against web servers and web applications. ***Patch management*** is the process of updating appropriate patches and hotfixes required by a system vendor. Proper patch management involves choosing how patches are to be installed and verified, and testing those patches on a nonproduction network prior to installation.

You should maintain a log of all patches applied to each system. To make patch installation easier, you can use automated patch-management systems provided by PatchLink, St. Bernard Software, Microsoft, and other software vendors to assess your systems and decide which patches to deploy.

Web Server Hardening Methods

A web server administrator can do many things to *harden* a server (increase its security). The following are ways to increase the security of the web server:

- nRename the administrator account, and use a strong password. To rename the administrator account in Windows, open the User Manager, right-click the Administrator account, and select Rename.
- Disable default websites and FTP sites. The process to disable default websites was described earlier in this chapter: right-click the default website in IIS Manager and choose Stop. The same process works for the default FTP site.
- Remove unused applications from the server, such as WebDAV. Unnecessary applications can be removed on a server by using Add/Remove Programs in the Windows Control Panel.
- Disable directory browsing in the web server's configuration settings.
- Add a legal notice to the site to make potential attackers aware of the implications of hacking the site.
- Apply the most current patches, hotfixes, and service packs to the operating system and web server software.
- Perform bounds checking on input for web forms and query strings to prevent buffer overflow or malicious input attacks.
- Disable remote administration.
- Use a script to map unused file extensions to a 404 ("File not found") error message.
- Enable auditing and logging.

5.5. Hacking Web Applications

Web Application Vulnerabilities

Web applications are programs that reside on a web server to give the user functionality beyond just a website. Database queries, webmail, discussion groups, and blogs are all examples of web applications.

A web application uses a client/server architecture, with a web browser as the client and the web server acting as the application server. JavaScript is a popular way to implement web applications. Because web applications are widely implemented, any user with a web browser can interact with most site utilities.

The purpose of hacking a web application is to gain confidential data. Web applications are critical to the security of a system because they usually connect to a database that contains information such as identities with credit card numbers and passwords. Web application vulnerabilities increase the threat that hackers will exploit the operating system and web server or web application software. Web applications are essentially another door into a system and can be exploited to compromise the system.

Hacking web applications is similar to hacking other systems. Hackers follow a five-step process: they scan a network, gather information, test different attack scenarios, and finally plan and launch an attack. The steps are listed in Figure.

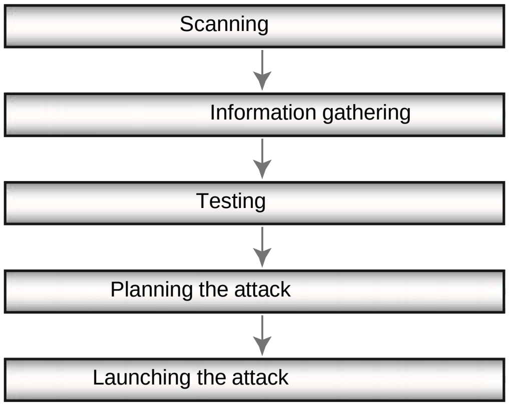


Fig: The stages of a web application attack

Web Application Threats and Countermeasures

Many web application threats exist on a web server. The following are the most common threats and their countermeasures:

Cross-Site Scripting: A parameter entered into a web form is processed by the web application. The correct combination of variables can result in arbitrary command execution. Countermeasure: Validate cookies, query strings, form fields, and hidden fields.

SQL Injection : Inserting SQL commands into the URL gets the database server to dump, alter, delete, or create information in the database.

Countermeasure:

Validate user variables.

Command Injection : The hacker inserts programming commands into a web form.

Countermeasure: Use language-specific libraries for the programming language.

Cookie Poisoning and Snooping : The hacker corrupts or steals cookies. Countermeasures: Don't store passwords in a cookie; implement cookie timeouts; and authenticate cookies.

Buffer Overflow : Huge amounts of data are sent to a web application through a web form to execute commands. Buffer overflows is covered in detail in Chapter 9. Countermeasures:

Validate user input length; perform bounds checking.

Authentication Hijacking : The hacker steals a session once a user has authenticated. Countermeasure: Use SSL to encrypt traffic.

Directory Traversal/Unicode : The hacker browses through the folders on a system via a web browser or Windows Explorer. Countermeasures: Define access rights to private folders on the web server; apply patches and hotfixes.

Google Hacking

Google hacking refers to using Google's powerful search engine to locate high-value targets or to search for valuable information such as passwords.

Many tools, such as http://johnny.ihackstuff.com and Acunetix Web Vulnerability Scanner, contain a list of Google hacking terms organized in a database, to make searching easier . For example, you can enter the term *password* or *medical records* in the Google search engine and see what information is available. Many times, Google can pull information directly out of private databases or documents.

Web-Based Password-Cracking Techniques

As a CEH(Certified Ethical Hacker), you need to be familiar with the techniques hackers use to crack web-based passwords. This includes being able to list the various authentication types, knowing what a password cracker is, identifying the classifications of password-cracking techniques, and knowing the available countermeasures.

Authentication Types

Web servers and web applications support multiple authentication types. The most common is HTTP authentication. There are two types of HTTP authentication: basic and digest. Basic HTTP authentication sends the username and password in cleartext, whereas digest authentication hashes the credentials and uses a challenge-response model for authentication.

In addition, web servers and web applications support the following types of authentication:

NTLM Authentication : This type uses Internet Explorer and IIS web servers, making NTLM more suitable for internal authentication on an intranet that uses Microsoft operating systems. Windows 2000 and 2003 servers utilize Kerberos authentication for a more secure option.

Certificate-Based Authentication : This type uses an x.509 certificate for public/private key technology.

Token-Based Authentication : A token, such as SecurID, is a hardware device that displays an authentication code for 60 seconds; a user uses this code to log into a network.

Biometric Authentication : This type uses a physical characteristic such as fingerprint, eye iris, or handprint to authenticate the user.

Password Attacks and Password Cracking

The three types of password attacks are as follows:

Dictionary : Uses passwords that can be found in a dictionary

Brute-Force : Guesses complex passwords that use letters, numbers, and special characters

Hybrid : Uses dictionary words with a number or special character as a substitute for a letter A *password cracker* is a program designed to decrypt passwords or disable password protection. Password crackers rely on dictionary searches (attacks) or brute-force methods to crack passwords.

The first step in a dictionary attack is to generate a list of potential passwords that can be found in a dictionary. The hacker usually creates this list with a dictionary generator program or dictionaries that can be downloaded from the Internet. Next, the list of dictionary words is hashed or encrypted. This hash list is compared against the hashed password the hacker is trying to crack. The hacker can get the hashed password by sniffing it from a wired or wireless network or directly from the Security Accounts Manager (SAM) or shadow password files on the hard drive of a system. Finally, the program displays the unencrypted version of the password. Dictionary password crackers can only discover passwords that are dictionary words.

If the user has implemented a strong password, then brute-force password cracking can be implemented. Brute-force password crackers try every possible combination of letters, numbers, and special characters, which takes much longer than a dictionary attack because of the number of permutations.

5.6. SQL Injection

SQL injection and buffer overflows are hacking techniques used to exploit weaknesses in applications. When programs are written, some parameters used in the creation of the application code can leave weaknesses in the program. SQL injection and buffer overflows are covered in the same chapter because they both are methods used to attack application and are generally caused by programming flaws. Generally, the purpose of SQL injection is to convince the application to run SQL code that was not intended.

SQL injection is a hacking method used to attack SQL databases, whereas buffer overflows can exist in many different types of applications. SQL injection and buffer overflows are similar exploits in that they're both usually delivered via a user input field. The input field is where a user may enter a username and password on a website, add data to a URL, or perform a search for a keyword in another application. The SQL injection vulnerability is caused primarily by unverified or unsanitized user input via these fields.

Both SQL Server injection and buffer overflow vulnerabilities are caused by the same issue: invalid parameters that are not verified by the application. If programmers don't take the time to validate the variables a user can enter into a variable field, the results can be serious and unpredictable. Sophisticated hackers can exploit this vulnerability, causing an execution fault and shutdown of the system or application, or a command shell to be executed for the hacker.

SQL injection and buffer overflow countermeasures are designed to utilize secure programming methods. By changing the variables used by the application code, weaknesses in applications can be greatly minimized. This chapter will detail how to perform a SQL injection and a buffer overflow attack and explore the best countermeasures to prevent the attack.

As a CEH(Certified Ethical Hacker), it's important for you to be able to define SQL injection and understand the steps a hacker takes to conduct a SQL injection attack. In addition, you should know SQL Server vulnerabilities, as well as countermeasures to SQL injection attacks.

SQL injection occurs when an application processes user-provided data to create a SQL statement without first validating the input. The user input is then submitted to a web application database server for execution. When successfully exploited, SQL injection can give an attacker access to database content or allow the hacker to remotely execute system commands. In the worst-case scenario, the hacker can take control of the server that is hosting the database. This exploit can give a hacker access to a remote shell into the server file system. The impact of a SQL injection attacks depends on where the vulnerability is in the code, how easy it is to exploit the vulnerability, and what level of access the application has to the database. Theoretically, SQL injection can occur in any type of application, but it is most commonly associated with web applications because they are most often attacked. During a web application SQL injection attack, malicious code is inserted into a web form field or the website's code to make a system execute a command shell or other arbitrary commands. Just as a legitimate user enters queries and additions to the SQL database via a web form, the hacker can insert commands to the SQL Server through the same web form field. For example, an arbitrary command from a hacker might open a command prompt or display a table from the database. A database table may contain personal information such as credit card numbers, social security numbers, or passwords. SQL Servers are very common database servers and used by many organizations to store confidential data. This makes a SQL Server a high-value target and therefore a system that is very attractive to hackers.

Finding a SQL Injection Vulnerability

Before launching a SQL injection attack, the hacker determines whether the configuration of the database and related tables and variables is vulnerable. The steps to determine the SQL Server's vulnerability are as follows:

1. Using your web browser, search for a website that uses a login page or other database input or query fields (such as an "I forgot my password" form). Look for web pages that display the POST or GET HTML commands by checking the site's source code.
2. Test the SQL Server using single quotes (''). Doing so indicates whether the user input variable is sanitized or interpreted literally by the server. If the server responds with an error message that says *use 'a'='a'* (or something similar), then it's most likely susceptible to a SQL injection attack.

3. Use the SELECT command to retrieve data from the database or the INSERT command to add information to the database.

Here are some examples of variable field text you can use on a web form to test for SQL vulnerabilities:

Blah' or 1=1- Login:blah' or 1=1-

Password::blah' or 1=1--

http://search/index.asp?id=blah' or 1=1--

These commands and similar variations may allow a user to bypass a login depending on the structure of the database. When entered in a form field, the commands may return many rows in a table or even an entire database table because the SQL Server is interpreting the terms literally. The double dashes near the end of the command tell SQL to ignore the rest of the command as a comment.

Here are some examples of how to use SQL commands to take control: To get a directory listing, type the following in a form field:

Blah';exec master..xp_cmdshell "dir c:\*.* /s >c:\directory.txt"-To create a file, type the following in a form field:

Blah';exec master..xp_cmdshell "echo hacker-was-here > c:\hacker.txt"-To ping an IP address, type the following in a form field:

Blah';exec master..xp_cmdshell "ping 192.168.1.1"--

The Purpose of SQL Injection

SQL injection attacks are used by hackers to achieve certain results. Some SQL exploits will produce valuable user data stored in the database, and some are just precursors to other attacks. The following are the most common purposes of a SQL injection attack:

Identifying SQL Injection Vulnerability The purpose is to probe a web application to discover which parameters and user input fields are vulnerable to SQL injection.

Performing Database Finger-Printing The purpose is to discover the type and version of database that a web application is using and "fingerprint" the database. Knowing the type and version of the database used by a web application allows an attacker to craft database specific attacks.

Determining Database Schema To correctly extract data from a database, the attacker often needs to know database schema information, such as table names, column names, and column data types. This information can be used in a follow-on attack.

Extracting Data These types of attacks employ techniques that will extract data values from the database. Depending on the type of web application, this information could be sensitive and highly desirable to the attacker.

Adding or Modifying Data The purpose is to add or change information in a database.

Performing Denial of Service These attacks are performed to shut down access to a web application, thus denying service to other users. Attacks involving locking or dropping database tables also fall under this category.

Evading Detection This category refers to certain attack techniques that are employed to avoid auditing and detection.

Bypassing Authentication The purpose is to allow the attacker to bypass database and application authentication mechanisms. Bypassing such mechanisms could allow the attacker to assume the rights and privileges associated with another application user.

Executing Remote Commands These types of attacks attempt to execute arbitrary commands on the database. These commands can be stored procedures or functions available to database users.

Performing Privilege Escalation These attacks take advantage of implementation errors or logical flaws in the database in order to escalate the privileges of the attacker.

SQL Injection Using Dynamic Strings

Most SQL applications do a specific, predictable job. Many functions of a SQL database receive static user input where the only variable is the user input fields. Such statements do not change from execution to execution. They are commonly called static SQL statements.

However, some programs must build and process a variety of SQL statements at runtime. In many cases the full text of the statement is unknown until application execution. Such statements can, and probably will, change from execution to execution. So, they are called dynamic SQL statements.

Dynamic SQL is an enhanced form of SQL that, unlike standard SQL, facilitates the automatic generation and execution of program statements. Dynamic SQL is a term used to mean SQL code that is generated by the web application before it is executed. Dynamic SQL is a flexible and powerful tool for creating SQL strings. It can be helpful when you find it necessary to write code that can adjust to varying databases, conditions, or servers. Dynamic SQL also makes it easier to automate tasks that are repeated many times in a web application.

A hacker can attack a web-based authentication form using SQL injection through the use of dynamic strings. For example, the underlying code for a web authentication form on a web server may look like the following:

```
SQLCommand = "SELECT Username FROM Users WHERE Username = '"
SQLCommand = SQLComand & strUsername
SQLCommand = SQLComand & "' AND Password = '"
SQLCommand = SQLComand & strPassword
SQLCommand = SQLComand & "'" strAuthCheck =
GetQueryResult(SQLQuery)
```

A hacker can exploit the SQL injection vulnerability by entering a login and password in the web form that uses the following variables:

```
Username: kimberly
Password: graves’ OR ‘’=’
```

The SQL application would build a command string from this input as follows:

```
SELECT Username FROM Users
WHERE Username = ‘kimberly’
```

AND Password = ‘graves’ OR ‘’=’’

This is an example of SQL injection: this query will return all rows from the user’s database, regardless of whether kimberly is a real username in the database or graves is a legitimate password. This is due to the OR statement appended to the WHERE clause. The comparison ‘’=’’ will always return a true result, making the overall WHERE clause evaluate to true for all rows in the table. This will enable the hacker to log in with any username and password.

SQL Injection Countermeasures

The cause of SQL injection vulnerabilities is relatively simple and well understood: insufficient validation of user input. To address this problem, defensive coding practices, such as encoding user input and validation, can be used when programming applications. It is a laborious and time-consuming process to check all applications for SQL injection vulnerabilities.

When implementing SQL injection countermeasures, review source code for the following programming weaknesses:

- Single quotes
- Lack of input validation

Buffer Overflows

The first countermeasures for preventing a SQL injection attack are minimizing the privileges of a user’s connection to the database and enforcing strong passwords for SA and Administrator accounts. You should also disable verbose or explanatory error messages so no more information than necessary is sent to the hacker; such information could help them determine whether the SQL Server is vulnerable. Remember that one of the purposes of SQL injection is to gain additional information as to which parameters are susceptible to attack.

Another countermeasure for preventing SQL injection is checking user data input and validating the data prior to sending the input to the application for processing.

Some countermeasures to SQL injection are

- Rejecting known bad input
- Sanitizing and validating the input field

As a CEH, you must be able to identify different types of buffer overflows. You should also know how to detect a buffer overflow vulnerability and understand the steps a hacker may use to perform a stack-based overflow attack. We'll look at these topics, as well as provide an overview of buffer-overflow mutation techniques, in the following sections.

Types of Buffer Overflows and Methods of Detection

Buffer overflows are exploits that hackers use against an operating system or application; like SQL injection attacks, they're usually targeted at user input fields. A buffer overflow exploit causes a system to fail by overloading memory or executing a command shell or arbitrary code on the target system. A buffer overflow vulnerability is caused by a lack of bounds checking or a lack of input-validation sanitization in a variable field (such as on a web form). If the application doesn't check or validate the size or format of a variable before sending it to be stored in memory, an overflow vulnerability exits.

The two types of buffer overflows are stack based and heap based.

The *stack* and the *heap* are storage locations for user-supplied variables within a running program. Variables are stored in the stack or heap until the program needs them. Stacks are static locations of memory address space, whereas heaps are dynamic memory address spaces that occur while a program is running. A heap-based buffer overflow occurs in the lower part of the memory and overwrites other dynamic variables

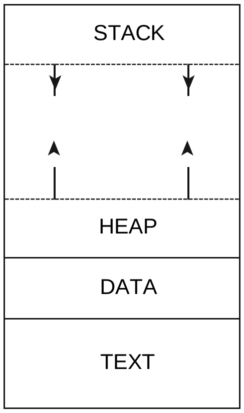


Fig: Stack versus Heap Memory

A call stack, or *stack*, is used to keep track of where in the programming code the execution pointer should return after each portion of the code is executed. A stack-based buffer overflow attack (occurs when the memory assigned to each execution routine is overflowed. As a consequence of both types of buffer overflows, a program can open a shell or command prompt or stop the execution of a program. The next section describes stackbased buffer overflow attacks.

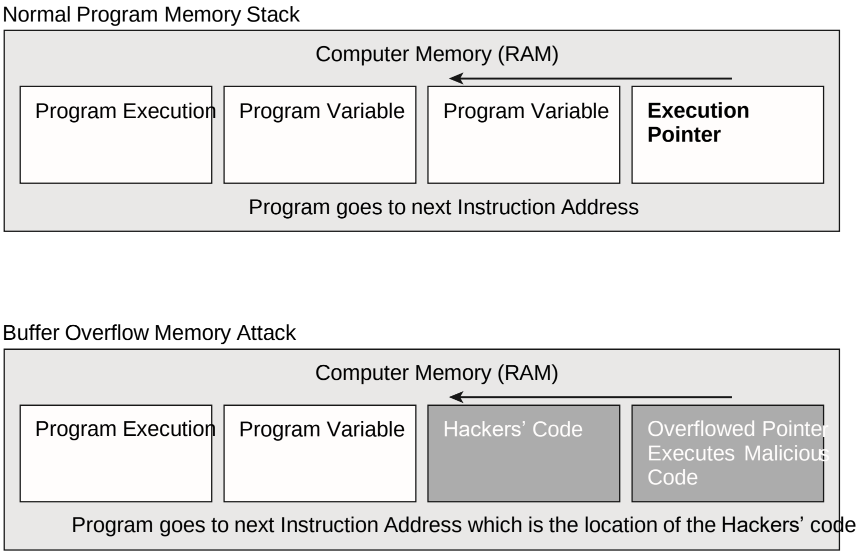

Normal Program Memory Stack
Computer Memory (RAM)
Program Execution
Program Variable
Program Variable
Execution Pointer
Program goes to next Instruction Address
Buffer Overflow Memory Attack
Computer Memory (RAM)
Program Execution
Program Variable
Hackers' Code
Overflowed Pointer Executes Malicious Code
Program goes to next Instruction Address which is the location of the Hackers' code

Fig: A stack-based buffer overflow attack

To detect program buffer overflow vulnerabilities that result from poorly written source code, a hacker sends large amounts of data to the application via a form field and sees what the program does as a result.

The following are the steps a hacker uses to execute a stack-based buffer overflow:

1. Enter a variable into the buffer to exhaust the amount of memory in the stack.
2. Enter more data than the buffer has allocated in memory for that variable, which causes the memory to overflow or run into the memory space for the next process. Then, add another variable, and overwrite the return pointer that tells the program where to return to after executing the variable.
3. A program executes this malicious code variable and then uses the return pointer to get back to the next line of executable code. If the hacker successfully overwrites the pointer, the program executes the hacker's code instead of the program code.

Most hackers don't need to be this familiar with the details of buffer overflows. Prewritten exploits can be found on the Internet and are exchanged between hacker groups.

5.7. Hacking Wireless Networks

Wireless networks add another entry point into a network for hackers. Much has been written about wireless security and hacking because wireless is a relatively new technology and rife with security vulnerabilities. From the increase of Wi-Fi hotspots to the rising number of cell phones, PDAs, and laptops equipped with Wi-Fi radios, wireless security is an ever increasing issue for many organizations.

Because of the broadcast nature of radio frequency (RF) wireless networks and the rapid adoption of wireless technologies for home and business networks, many hacking opportunities exist in wireless networking. Even for organizations with a “no wireless” policy—meaning they do not support any Wi-Fi connectivity—rogue wireless access points placed on the LAN are an increasing threat. The cost of Wi-Fi equipment is dropping and many organizations are pressing the IT staff to install wireless networks to complement or replace existing wired networks.

Wi-Fi and Ethernet

It is important to recognize that Wi-Fi networks are fundamentally different from Ethernet networks. Whereas in an Ethernet network the data is carried in frames on copper or fiber-optic cabling, in a Wi-Fi network the data travels across open air. Additionally, any encryption applied to wireless networks only encrypts the data itself, leaving the header potion of the wireless frame open to many types of attacks. The details of wireless attacks and countermeasures will be covered later in this chapter, but first you need to understand the fundamentals of the 802.11 standards and protocols.

802.11 Wireless LANs operate at layer 1 and 2 of the OSI Model. This means that the protocols in use on a WLAN are the same from Layer 3 (usually IP) on up to Layer 7 (the application layer).

Many people call 802.11 WLANs “wireless Ethernet,” which is a big misnomer. 802.11 has a completely different frame format at Layer 2 than does 802.3 (Ethernet). For example, Ethernet Layer 2 frames carry only two MAC addresses, while 802.11 frames have fields for four MAC addresses. Ethernet just defines source and destination addresses, while an 802.11 frame can define source, destination, transmitter and receiver. 802.11 frames also carry a frame control field in the MAC header used to indicate information about the frame, such as if the frame is encrypted.

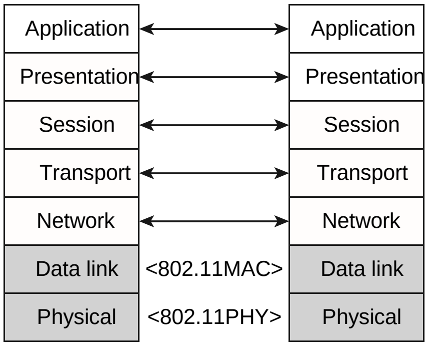
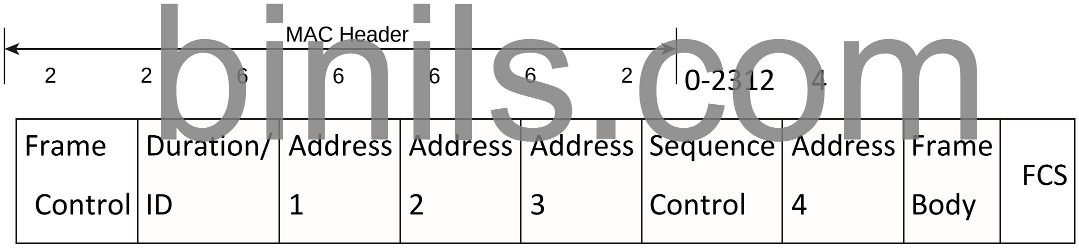


Fig: Wireless LANs in the OSI Model



2	2	6	6	6	6	2	0-2312	4
Frame Control	Duration/ ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Frame Body	FCS

Fig: 802.11 MAC Header

There are three types of 802.11 frames:

- Management—Used for notification, connection, disconnection, and information.
- Control—Used to control which station has access to the wireless network media.
- Data—Used to carry upper layer data.

Most wireless LANs (WLANs) are based on the IEEE 802.11 standards and amendments, such as 802.11a, 802.11b, 802.11g, and 802.11n. The lettered amendments have been rolled up into a final 802.11 standard and are now referred to by the clause or section number within the 802.11 standard. Table shows a comparison of the 802.11 standard amendments.

IEEE Standard	Frequency	Speed	Transmission Range	Spread Spectrum
802.11	2.4 GHz	Up to 2 Mbps	Depends on spread spectrum type	DSSS and FHSS
802.11a	5 GHz	Up to 54 Mbps	25 to 75 feet indoors; range can be affected by building materials	OFDM
802.11b	2.4 GHz	Up to 11 Mbps	Up to 150 feet indoors; range can be affected by building materials	DSSS
802.11g	2.4 GHz	Up to 54 Mbps	Up to 150 feet indoors; range can be affected by building materials	DSSS
802.11n	2.4 and 5 GHz	Up to 600 Mbps	At least as far as b, g, and a—and possibly much further	OFDM

Table 802.11 comparison

The initial 802.11 standard included only rudimentary security features and was fraught with vulnerabilities. The 802.11i a Binils.com – Free Anna University, Polytechnic, School Study Materialsmendment is the latest security solution that addresses the 802.11 weaknesses. The Wi-Fi Alliance created additional security certifications known as *Wi-Fi Protected Access* (WPA) and WPA2 to fill the gap between the original 802.11 standard and the latest 802.11i amendment. The security vulnerabilities and security solutions discussed in this chapter are all based on these IEEE and Wi-Fi Alliance standards.

Authentication and Cracking Techniques

Two methods exist in the 802.11 standard for authenticating wireless LAN clients to an access point: open system or shared-key authentication. Open system does not provide any security mechanisms but is simply a request to make a connection to the network. Sharedkey authentication has the wireless client hash a string of challenge text with the Wired Equivalent

Privacy (WEP) key to authenticate the client to the network. Table 10.2 compares the Wi-Fi security standards type of authentication and encryption.

WEP was the first security option for 802.11 WLANs. WEP is used to encrypt data on the WLAN and can optionally be paired with shared-key authentication to authenticate WLAN clients. WEP uses an RC4 64-bit or 128-bit encryption key to encrypt the Layer 2 data payload. This WEP key comprises a 40-bit or 104-bit user-defined key combined with a 24-bit Initialization Vector (IV), making the WEP key either 64 or 128 bit.

Wi-Fi Security	Authentication	Cipher	Encryption
WPA-Personal	Preshared Key	TKIP	RC4
WPA-Enterprise	802.1X/EAP	TKIP	RC4
WPA2-Personal	Preshared Key	CCMP (default), TKIP (optional)	AES (default), RC4 (optional)
WPA2-Enterprise	802.1X/EAP	CCMP (default), TKIP (optional)	AES (default), RC4 (optional)

Table : Wi-Fi security comparison

The process by which RC4 uses IVs is the real weakness of WEP: it gives a hacker the opportunity to crack the WEP key. The method, knows as the *Fluhrer, Mantin, and Shamir (FMS) attack*, uses encrypted output bytes to determine the most probable key bytes. The ability to exploit the WEP vulnerability was incorporated into products like AirSnort, WEPCrack, and Aircrack. Although a hacker can attempt to crack WEP by brute force, the most common technique is the FMS attack.

WPA employs the Temporal Key Integrity Protocol (TKIP)—which is a safer RC4 implementation—for data encryption and either WPA Personal or WPA Enterprise for authentication. WPA Personal uses an ASCII passphrase for authentication whereas WPA Enterprise uses a RADIUS server to authenticate users. WPA Enterprise is a more secure robust security option but relies on the creation and more complex setup of a RADIUS server. TKIP rotates the data encryption key to prevent the vulnerabilities of WEP and, consequently, cracking attacks.

WPA2 is similar to 802.11i and uses the Advanced Encryption Standard (AES) to encrypt the data payload. AES is considered an uncrackable encryption algorithm. WPA2 also allows for the use of TKIP during a transitional period called *mixed mode security*. This transitional mode means both TKIP and AES can be used to encrypt data. AES requires a faster processor, which means low-end devices like PDAs may only support TKIP.

WPA Personal and WPA2 Personal use a passphrase to authentication WLAN clients. WPA Enterprise and WPA2 Enterprise authenticate WLAN users via a RADIUS server using the 802.1X/Extensible Authentication Protocol (EAP) standards. Figure 10.3 shows the 802.1x/EAP process and the communication process used to authenticate a client using 802.1x/EAP.

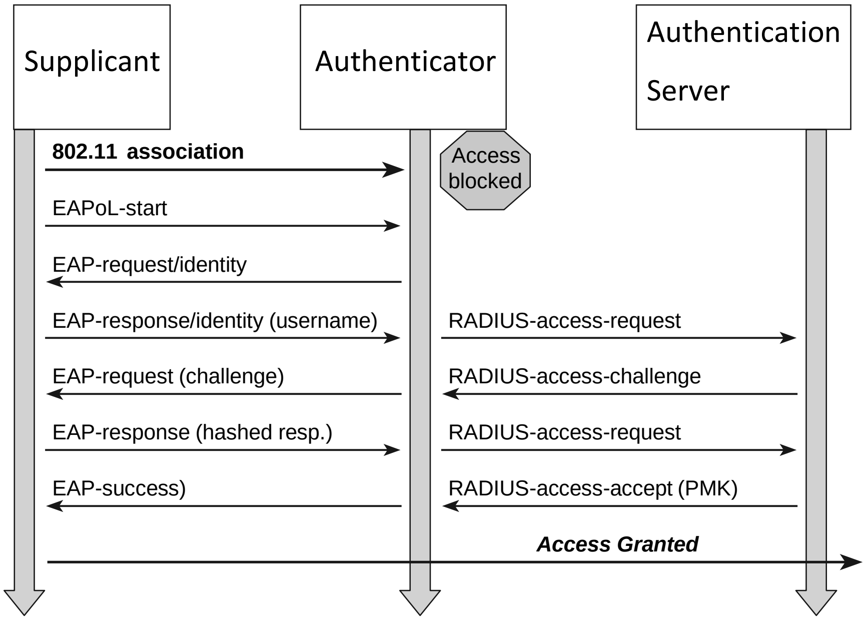


Fig:802.1X authentication process

Wireless Hacking Techniques

Most wireless hacking attacks can be categorized as follows:

Cracking Encryption and Authentication Mechanisms These mechanisms include cracking WEP, WPA preshared key authentication passphrases, and Cisco's Lightweight EAP authentication (LEAP). Hackers can use these mechanisms to connect to the WLAN using stolen credentials or can capture other users' data and decrypt or encrypt it. A protection against this attack is to implement a stronger type of encryption, such as AES.

Eavesdropping or Sniffing This type of attack involves capturing passwords or other confidential information from an unencrypted WLAN or hotspot. A protection against this attack is to use SSL application-layer encryption or a VPN to secure user data.

Denial of Service DoS can be performed at the physical layer by creating a louder RF signature than the AP with an RF transmitter, causing an approved AP to fail so users connect to a rogue AP. DoS can be performed at the Logical Link Control (LLC) layer by generating deauthentication frames (deauth attacks), by continuously generating bogus frames, or by having a wireless NIC send a constant stream of raw RF (Queensland attack). A countermeasure is to enforce a security perimeter around your WLAN and detect and remove sources of DoS attacks using an IDS.

AP Masquerading or Spoofing Rogue APs pretend to be legitimate APs by using the same configuration SSID settings or network name. A countermeasure to AP masquerading is to use a WIDS to detect and locate spoofed APs.

MAC Spoofing The hacker pretends to be a legitimate WLAN client and bypasses MAC filters by spoofing another user's MAC address. WIDSs can detect MAC spoofing, and not using MAC filtering is a way to avoid MAC spoofing attacks.

Planting Rogue Access Points The most dangerous attack is a rogue AP that has been planted to allow a hacker access to the target LAN. A countermeasure is to use a WIPS to detect and locate rogue APs.

Wireless networks give a hacker an easy way into the network if the AP isn't secured properly. There are many ways to hack or exploit the vulnerabilities of a WLAN. There are also effective countermeasures to many of these attacks. The next section will detail the best methods to secure wireless network.

Securing Wireless Networks

Because wireless networking is a relatively new technology compared to wired networking technologies, fewer security options are available. Security methods can be categorized by the applicable layer of the OSI model.

Layer 2, or MAC layer, security options are as follows:

- ✓ Static WEP (not recommended)
- ✓ WPA
- ✓ WPA2/802.11i

Layer 3, or Network layer, security options are as follows:

- ✓ IPSec
- ✓ SSL VPN

Layer 7, or Application layer, security options are as follows:

- ✓ Secure applications such as Secure Shell (SSH), HTTP over SSL (HTTPS), and FTP/SSL (FTPS)

Securing Home Wireless Networks

Many people setting up wireless home networks rush through the job to get their Internet connectivity working as quickly as possible. The small office, home office (SOHO) networking products on the market make setup quick and easy but not necessarily secure. Configuring additional security features can be time consuming and nonintuitive for some home users, and therefore they may not implement any security mechanism at all.

These days wireless networking products are so ubiquitous and inexpensive that just about anyone can set up a WLAN in a matter of minutes with less than $100 worth of equipment. This widespread use of wireless networks means that there may be dozens of potential network intruders within range of your home or office WLAN. Most WLAN hardware has gotten easy enough to set up that many users simply plug it in and start using the network without giving much thought to security. Nevertheless, taking a few extra minutes to configure the security features of your wireless router or access point is time well spent. The following recommendations will improve the security of your home wireless network:

Change default administrator passwords and usernames. When configuring your home access point, you usually use a web browser to access the configuration interface. Almost all routers and access points have an administrator password that's needed to log into the device and modify any configuration settings. To set up these pieces of equipment, manufacturers provide a default username and password. Many of the default logins are simple (such as username=admin and password=admin) and very well known to hackers on the Internet. Most devices use a weak default password like "password" or the manufacturer's name, and some don't have a default password at all. You should change the default password on your home AP as soon as possible. As soon as you set up a new WLAN router or access point, your first step should be to change the default administrative password to something else.

Use WEP/WPA encryption. Most Wi-Fi equipment supports some form of *encryption*. Encryption technology scrambles messages sent over wireless networks so that they cannot be easily read by hackers. You should configure the strongest form of encryption that works with your wireless clients. 802.11's WEP (Wired Equivalency Privacy) encryption has well known weaknesses that make it relatively easy for a determined user with the right equipment to crack the encryption and access the wireless network. A better way to protect your WLAN is with WPA (Wi-Fi Protected Access). WPA provides much better protection and is also easier to use, since your password characters aren't limited to 0–9 and A–F as they are with WEP. (Note: WEP can also use ASCII keys.)

Change the default SSID. Access points use a network name called an SSID to advertise the network to wireless users. Manufacturers normally ship their products with the same SSID set. For example, the SSID for Linksys devices is normally "Linksys." Just knowing the SSID does not by itself allow your neighbors to break into your network, but it is a start. More importantly, when someone finds a default SSID, it is usually an indication of a poorly configured network.

You should change the default SSID immediately when configuring wireless security on your network.

Do not auto-connect to open Wi-Fi networks. Connecting to an open Wi-Fi network such as a free wireless hotspot or an unknown WLAN exposes your computer to security risks. Most computers have a setting available allowing these connections to happen automatically without notifying you. Most versions of Windows will reconnect to a previously connected SSID. This setting should not be enabled except in temporary situations.

Enable firewall settings on your laptop and home access point. Most network routers contain built-in firewall capability, but the option also exists to disable them. Ensure that your router's firewall is turned on. You should always install and configure personal firewall software on each computer connected to the router.

Reduce your WLAN transmitter power. You won't find this feature on all wireless routers and access points, but some allow you to lower the power of your WLAN transmitter and thus reduce the range of the signal. (Normally this feature is only available with enterprise class access points.) Although it's usually impossible to fine-tune a signal so precisely that it won't leak outside your home or business, with some trial and error you can often limit how far outside your premises the signal reaches, minimizing the opportunity for outsiders to access your WLAN. This will also improve your throughput on your access point by limiting the wireless cell to just your premise.

Disable remote administration. Most WLAN routers have the ability to be remotely administered via the Internet. Ideally, you should use this feature only if it lets you define a specific IP address or limited range of addresses that will be able to access the router. Otherwise, almost anyone anywhere could potentially find and access your router. As a rule, unless you absolutely need this capability, it's best to keep remote administration turned off.

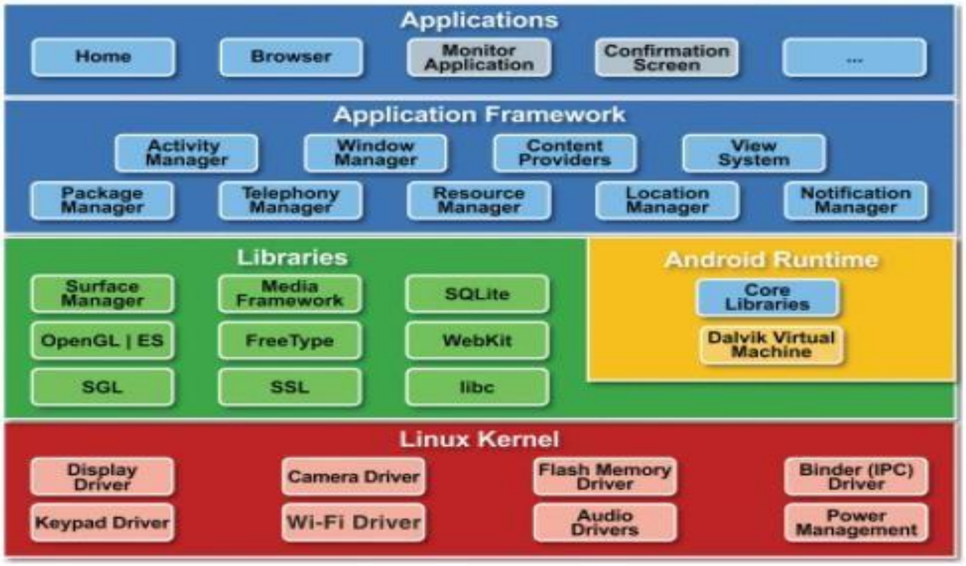
5.8. Hacking Mobile Platforms

The mobile device has become an inseparable part of life today. The attackers are easily able to compromise the mobile network because of various vulnerabilities, the majority of the attacks are because of the untrusted apps. SMS is another way the attackers are gaining access to the mobile devices by sending phishing messages/spam messages to users. The main operating systems used are:

- Android
- IOS
- Windows
- Blackberry

Android

Android occupies the major share of the world's mobile market because of its user friendliness. Android uses a Linux operating system, it uses Dalvik virtual machine which runs the java files by converting them to .dvk files for faster speed and performance. The native libraries and modules are used for various functions of android. The applications communicate with other applications through messages called intents.



Types of Android Attacks

Untrusted APK’s:

Attackers lure users to download applications from untrusted sources. These APK’s may contain malicious software inside them, giving the attacker remote access to the mobile device when the APK is installed by the user.

SMS:

The user may come across a suspicious SMS giving them big bounty’s. When the users click that particular link in the message, they may be redirected to a malicious website giving away their sensitive information or may lead to financial loss.

Email:

Phishing emails may redirect the users to malicious websites compromising the user’s details. SPAM emails may steal information from the users.

Spying:

Some applications may spy on the mobile users and report to the remote attackers.

App sandboxing issues:

Sandboxing is the process of testing an App in a limited resource environment against various threats and attacks. If sandboxing has issues, it means that malicious applications can bypass this mechanism.

Rooting:

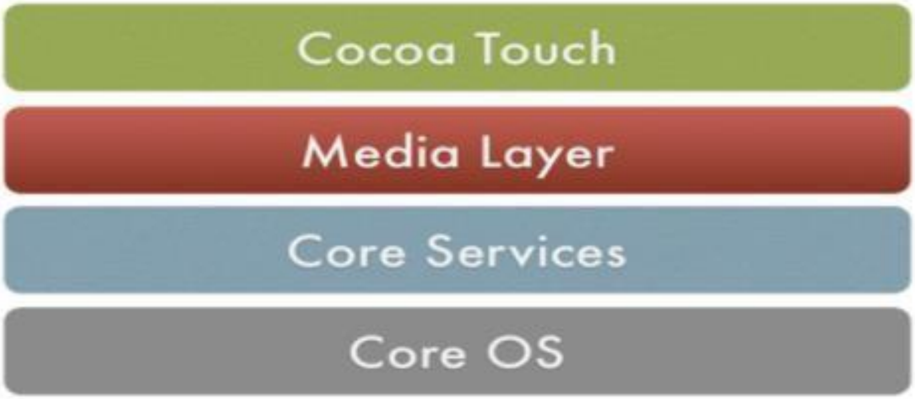
Rooting is done for increasing speed and performance of an android device. This is not a recommended solution by the android authorities. When a phone is rooted, it loses its warranty and may open the door for various malware and allows the attacker to take control of the device remotely.

Countermeasures:

- Do not root your phone.
- Do not download applications from untrusted third party sources.
- Do not click on suspicious emails.
- Do not open suspicious SMS.
- Use strong passwords/patterns.
- Use Device administration API to set up password policy, remote wipe, etc.
- Do not store passwords on phone.
- Update the operating system regularly.
- Use strong anti-virus.

IOS

IOS uses proprietary software. The attacks on these phones are limited since they are not open source systems.



Types of IOS Attacks:

Jailbreaking:

Jailbreaking may put the device at risk. It is done to gain administrative privileges and to download third-party application extensions, etc. Though, the device may lose its warranty, get infected with malware, drop in performance, etc.

There are three ways jailbreaking can be done-

Tethered:

After a device is jailbroken, it will no longer have a patched kernel; it might go to a partially functioning state and requires re-jailbreaking using the same computer.

Semi-tethered:

When the device is turned off and on, it will no longer be jailbroken. The device can be used for normal functions.

Untethered:

The device once jailbroken remains jailbroken, and the kernel will be patched completely after reboot.

Countermeasures:

- ✓ Do not jailbreak the device.
- ✓ Apply strong encryption.
- ✓ Always connect to safe networks.
- ✓ Follow common security guidelines.
- ✓ Do not open links/attachments from unknown sources.