P.S.R ENGINEERING COLLEGE

(An Autonomous Institution – Affliated to Anna University, Chennai)

SIVAKASI - 626 140

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBJECT NAME	:	**ETHICAL HACKING AND NETWORK DEFENSE**
SUBJECT CODE	:	191CS74
SEMESTER/YEAR/SEC	:	VII / IV / I
BRANCH	:	CSE
STAFF	:	Dr.R.Arun

PREPARED BY	**APPROVED BY**
(Dr.R.Arun)	**(HOD/CSE)**

191CS74	**ETHICAL HACKING AND NETWORK DEFENSE**	**L**	**T**	**P**	**C**
		3	**0**	**2**	4

Programme:	B.E. Computer Science and Engineering	**Sem:**	7	**Category:**	**PC**
Prerequisites:	NIL				

Aim: To understand and analyze Information security threats & countermeasures.

Course Outcomes: The Students will be able to

CO1:	Recall the basic concepts of vulnerabilities and hacking.
CO2:	Apply the various tools for port scanning and foot printing in the Windows/Linux OS.
CO3:	Examine the various hacking methodologies in system.
CO4:	Demonstrate the various web application vulnerabilities.
CO5:	Identify the types and tools for session hijacking.
CO6:	Interpret the different kinds of tools forhacking the wireless networks.

ETHICAL HACKING OVERVIEW AND VULNERABILITIES **9**

Understanding the importance of security, Concept of ethical hacking and essential Terminologies-Threat, Attack, Vulnerabilities, Target of Evaluation, Exploit. Phases involved in hacking.

FOOT PRINTING AND PORT SCANNING **9**

Foot printing - Introduction to foot printing, Understanding the information gathering methodology of the hackers, Tools used for the reconnaissance phase. Port Scanning - Introduction, using port scanning tools, ping sweeps, Scripting Enumeration-Introduction, Enumerating windows OS & Linux OS.

SYSTEM HACKING **9**

Aspect of remote password guessing, Role of eavesdropping ,Various methods of password cracking, Keystroke Loggers, Understanding Sniffers ,Comprehending Active and Passive Sniffing, ARP Spoofing and Redirection, DNS and IP Sniffing, HTTPS Sniffing.

HACKING WEB SERVICES AND SESSION HIJACKING **9**

Web application vulnerabilities, application coding errors, SQL injection into Back-end Databases, cross-site scripting, cross-site request forging, authentication bypass, web services and related flaws, protective http headers Understanding Session Hijacking, Phases involved in Session Hijacking, Types of Session Hijacking, Session Hijacking Tools.

HACKING WIRELESS NETWORKS **9**

Introduction to 802.11, Role of WEP, Cracking WEP Keys, Sniffing Traffic, Wireless, DOS attacks, WLAN Scanners, WLAN Sniffers, Hacking Tools, Securing Wireless Networks.

Total Periods: **45**

COMPONENT LAB – LIST OF EXPERIMENTS:

1. Implement boot sector virus and batch file execution
2. Implement any one password cracking algorithm
3. Develop DOS attack
4. Packet analyzer tool
5. Implement a program for cracking WEP password
6. Implement IP masking procedure.

Text Books:

1. Kimberly Graves, "Certified Ethical Hacker", Wiley India Pvt Ltd, 2010
2. Michael T. Simpson, "Hands-on Ethical Hacking & Network Defense", Course Technology, 2010

References:

1. RajatKhare, "Network Security and Ethical Hacking", Luniver Press, 2006
2. Ramachandran V, BackTrack 5 Wireless Penetration Testing Beginner’s Guide”, Packet, 3/e. blishing, 2011
3. Thomas Mathew, "Ethical Hacking", OSB publishers, 2003

Extracted from College Website: https://psr.edu.in/syllabus/

HOD/CSE

Ethical Hacking

* Ethical hacking is the authorized and legal process of finding out vulnerablities, threats and exploits in a system or computer network through penetration testing.

* White hat hacker is a ethical hacker So the hacker hacks into the target system with permission of the owner of the target system and compiles a report how he enters into the system all of the security ~~fl~~ laws and exploits and sends it to the owner or Employee of the system.

Purpose of Ethical Hacking

* Ethical hacker Can use the same Software tools and techniques as malicious hackers to find the security weakness ~~on~~. in Computer networks and systems.

* But It apply the necessary fix or Patch to prevent the malicious hacker from gaining access to the data.

* Ethical hackers are usually security professionals or network penetration testers who use their hacking skills and toolsets for defensive and protective purposes.

* Ethical hackers who are security professionals test their network and systems security for vulnerablities using the same tool that a hacker might use to compromise the network.

what do Ethical hackers Do?

⇒ Ethical hackers are motivated by different reasons, but their purpose is usually the same as that of crackers.

⇒ They are trying to determine what an intruder can see on a targeted network or system, and what the hacker can do with the information.

⇒ This process of testing the security of a system or network is known as a penertation test.

⇒ A penetration test plan can be built around the data that needs to be protected and potential risks.

⇒ Documenting the results of various tests is critical in producing the end product of the penetration test.

Goals of Attackers

All attacks like ethical hacker or malicious hacker are an attempt to breach Computer system Security.

Security consists of four basic elements

* Confidentiality

* Authenticity

* Integrity

* Availability

⇒ A hackers goal is to exploit vulnerablities in a system or network to find a weakness in one or more of the four elements of Security.

⇒ For example, in performing a denial-of-service (DoS) attack, a hacker attacks the availability elements of systems and networks.

⇒ DoS is denying Service to legitimate users of the system.

⇒ Information theft, such as Stealing passwords or other data as it travels in Cleartext across trusted networks, is a confidentiality attack, because it allows someone other than the intended recipient to gain access to the data.

⇒ This theft isn't limited to data on network servers. Laptops, disks and backups tapes are all at risk.

⇒ Bit-flipping attacks are considered integrity attacks because the data may have been tampered with in transit or at rest on computer system.

⇒ A bit flipping attack is an attack on a Cryptographic cipher: the attacker changes the cipher-text in such a way as to result in a predictable change of the plain text, although the attacker doesn't learn the plaintext itself.

⇒ This type of attack isn't directed against the cipher but against a message or series of messages.

⇒ MAC address spoofing is an authentication attack because it allows an unauthorized device to connect to the network when media access control (MAC) filtering is in place, such as on a wireless network.

⇒ By spoofing the MAC address of a legitimate wireless station, an intruder can take on that situation's identity and use the network.

Ethical hacking Terminologies

⇒ This terminology is how security professionals acting as ethical hackers communicate.

⇒ Terminologies that involved in Ethical hacking are

1. Threats
2. Exploit
3. Vulnerability
4. Target of Evaluation (TOE)
5. Attack
6. Remote
7. Local

1. Threats

→ An environment or situation that could lead to a potential breach of security.

→ Ethical hackers look for and prioritize threats when performing a security analysis.

→ Malicious hackers and their use of software and hacking techniques are themselves threats to an organization's information security.

Different types of threats include

* Physical Threat

* Internal Threat

* External Threat

* Human Threat

* Non- physical Threat

Physical Threat

⇒ Physical Threats may result in accidental or delibrate damage to the computer system hardware and infrastructure.

⇒ They can be occur intentionally, accidentally, or by any other means. like internal, external or even human errors.

Internal Threats

Internal factors like unstable power supply, hardware fault, internal humidity etc. may result in physical damage to the system.

External Threats

Lightning, floods, and earthquakes are some of the major and common external factors that may cause damage to the hardware and other physical parts of the computer system.

Human Threats

⇒ These may be intentional or accidental.

⇒ Theft, vandalism of infrastructure and/or hardware are some of the common damages caused by human errors or delibrate attempts.

Non-physical Threats

⇒ These include all potential reasons for contactless security breaches that results in data corruption information loss, operational distruption, and cybersecurity breaches etc.

2. Exploit

→ A piece of software or technology that takes advantage of a bug, glitch or vulnerability, leading to unauthorized access, privilage escalation, or denial of service on a computer system.

→ Malicious hackers are looking for exploits in computer systems to open the door to an initial attack.

→ Most exploits are small strings of computer code that, when executed on a system, expose vulnerability.

→ Experienced hackers create their own exploits, but it is not necessary to have ~~any~~ any programming skills to be an ethical hacker as many hacking software programs have ready-made exploits that can be launched against a computer system or network.

→ An exploit is a defined way to breach the security of an IT system through a vulnerablity.

3. Vulnerability

→ The existence of a software flaw, logic design, or implementation error that can lead to an unexpected and undesirable event executing bad or damaging instructions to the system.

→ Exploit code is written to target a vulnerability and cause a fault in the system in order to retrieve valuable data.

→ Vulnerablities mostly happened because of Hardware, Software, network and Procedural Vulnerabilities.

I. Hardware Vulnerability

* It is a weakness which can used to attack the system hardware through physically or remotely.

for example

1. Old Version of systems or devices
2. Unprotected Storage
3. Unencrypted devices etc.

II. Software Vulnerability

* A Software error happen In development or configuration such as the execution of it can Violate the security policy. For example

1. Lack of Input Validation
2. Unverified uploads
3. Cross-site Scripting
4. Unencrypted data, etc.

III. Network Vulnerability.

A weakness happen in network which can be hardware or software.

For example

1. Unprotected Communication
2. Malware or malicious software
3. Social engineering attacks
4. Misconfigured firewalls.

IV. Procedural Vulnerability

A weakness happen in an Organization operational methods.

1. Password procedure - password Should follow the standard policy.
2. Training procedure - Employees must know which actions should be taken and what to do to handle the security.

4. Target of Evaluation (TOE)

→ A system, program or network that is the subject of a security analysis or attack.

→ Ethical hackers are usually concerned with high-value TOE's systems that contain sensitive information such as account no, passwords, social security numbers, or other confidential data.

→ It is a goal of the ethical hacker to test hacking tools against the high value TOEs to determine the vulnerabilities and patch them to protect against exploits and exposure of sensitive data.

5. Attacks

→ An attack occurs when a system is compromised based on a vulnerability.

→ Many attacks are perpetuated via an exploit.

→ Ethical hackers use tools to find systems that may be vulnerable to an exploit because of the operating system, network configuration, or applications installed on the systems, and to prevent an attack.

There are different types of attacks, such as

i) Virus

A Virus is a harmful computer program that when executed, replicates itself and modifies the program of the host computer system by inserting its own code.

ii) Spyware

A collection of malicious programs, that is designed to extract information from computer systems, against its user's legitimate consent is known as spyware.

iii) Phishing

Phishing is referred to as the fradulent practice of sending emails pretending to be genuine in order to extract valuable information from the user.

iv) Worms

Computer worms are self-replicating malicious programs designed to spread across the computer network majorly in an organization.

v) Spam

⇒ Refers to irrelevant and unrecognized source messages sent via mail with the objective of advertising, malware Insertion, Phishing etc.

⇒ Spams can be distributed via Phone calls, text messages, or social media.

vi) DoS attacks

DoS stands for denial of Service. DoS attacks are designed to trigger crashes of the computer system resulting in a complete system shutdown making it inaccessible by its intended users.

There are two primary methods of delivering exploits to computer systems:

~~b) Remote~~

i) Remote

ii) Local

Remote

→ The exploit is sent over a network and exploits security vulnerabilities without any prior access to the vulnerable system.

→ Hacking attacks against corporate computer systems or networks initiated from the outside world are considered remote.

→ Most people think of this type of attack when they hear the term hacker, but in reality most attacks are in the next category called local.

Local

→ The exploit is delivered directly to the computer system or network, which requiries prior access to the vulnerable system to increase privileges.

→ Information security policies should be created in such a way that only those who need access to information should be allowed access and they should have the lowest level of access to perform their job function.

→ These concepts are commonly referred as "need to know" and "least privilege" and, when used properly, would prevent local exploits.

→ Most hacking attempts occur from within an organization and are perpetuated by employees, constractors, or others in a trusted position.

→ In order for an insider to launch an attack, they must have higher privileges than necessary based on the concept of "need to know".

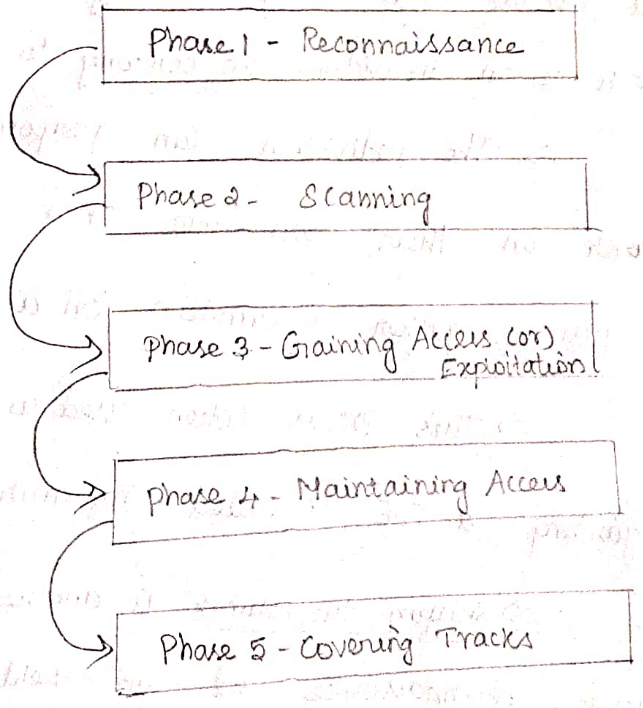
→ This can be accomplished by privilege escalation or weak security safeguards.

The Phases Of Ethical Hacking

⇒ The process of ethical hacking can be broken down into five distinct phases.

⇒ An ethical hacker follows processes similar to those of a malicious hacker.

⇒ The steps to gain and maintain entry into a computer system are similar no matter what the hackers intentions are. The following figure illustrates the five phases that hackers generally follow in hacking a computer system.


Phase 1 - Reconnaissance
Phase 2 - Scanning
Phase 3 - Gaining Access (or) Exploitation
Phase 4 - Maintaining Access
Phase 5 - Covering Tracks

Phase I : Passive and Active Reconnaissance

⇒ Passive reconnaissance involves gathering information about a potential target without the targeted individual's or company's knowledge.

⇒ Passive reconnaissance can be as simple as watching a building to Identify what time employees enter the building and when they leave.

⇒ However, most reconnaissance is done sitting in front of a Computer.

⇒ when hackers are looking for information on a potential target, they commonly run an internet search on an individual or company to gain information.

⇒ The individual can perform the same search on their own name or a potential employer, or just to gather information on a topic.

⇒ This process when used to gather information regarding a TOE is called Information gathering.

⇒ Sniffing the network is another means of passive reconnaissance and can yield useful information such as IP address ranges, naming conventions, hidden servers or networks, and other available services on the system or network.

⇒ Sniffing network traffic is a common hook for many ethical hackers. once they use some of the hacking tools and are able to see all the data that is transmitted in the clear over the commonunication networks, they are eager to learn and see more.

⇒ Sniffing tools are simple and easy to use and yield a great deal of valuable information.

⇒ Active reconnaissance involves probing the network to discover individual hosts, IP addresses and services on the network.

⇒ This process involves more risk of detection than passive reconnaissance and is sometimes called rattling the doorknobs.

⇒ Active reconnaissance can give a hacker an indication of security measures in place, but the process also increases the chance of being caught or atleast raising suspicions.

⇒ many software tools that perform active reconnaissance can be traced back to the computer that is running the tools, thus increasing the chance of detection for the hacker.

⇒ Both Passive and active reconnaissance can lead to the discovery of useful information to use in the attack.

Phase 2: Scanning

⇒ Scanning Involves taking the information discovered during reconnaissance and using it to examine the network.

⇒ Tools that a hacker may employ during the scanning phase include

* Dialers
* Port scanners
* Internet Control Message protocol (ICMP) Scanners
* Ping Sweeps
* Network mappers
* Simple Network management Protocol (SNMP) Sweepers
* Vulnerability Scanners

Hackers are seeking any information that can help them perpetrate an attack on a target, such as the following

* Computer names
* operating system (OS)
* Installed Software
* IP addresses
* User accounts.

Phase 3: Gaining Access or Exploitation

⇒ Vulnerablities exposed during the reconnaissance and scanning phase are now exploited to gain access to the target system.

⇒ The hacking attack can be delivered to the target program via a local Area network (LAN), either wired or wireless, local access to a PC, the internet, or offline.

⇒ Examples include stack based buffer overflows, denial of service, and session hijacking.

⇒ Gaining access is known in the hacker world as owning the system because once a system has been hacked, the hacker has controlled and use that system as they wish.

Phase 4: Maintaining Access

⇒ Once a hacker has gained access to a target system, they want to keep that access for future exploitation and attacks.

⇒ Sometimes, hackers harden the system from other hackers or security personnel by securing their exclusive access with backdoors, rootkits and Trojans.

⇒Once the hacker owns the system, they can use it as a base to launch additional attacks. In this case, the owned system is sometimes reporred to as a Zombie system.

Phase 5: Covering Tracks

⇒ Once hackers have been able to gain and maintain access, they cover their tracks to avoid detection by security personnel, to continue to use the owned system, to remove evidence of hacking, or to avoid legal action.

⇒ Hackers try to remove all traces of the attack, such as log files or intrusion detection system (IDS) alarms.

⇒ Examples of activities during this phase of the attack include

* Steganography
* Using a tunneling protocol
* Altering log files.

Steganography, using tunneling protocols, and altering log files for purposes of hacking will be discussed later.

UNIT - 2

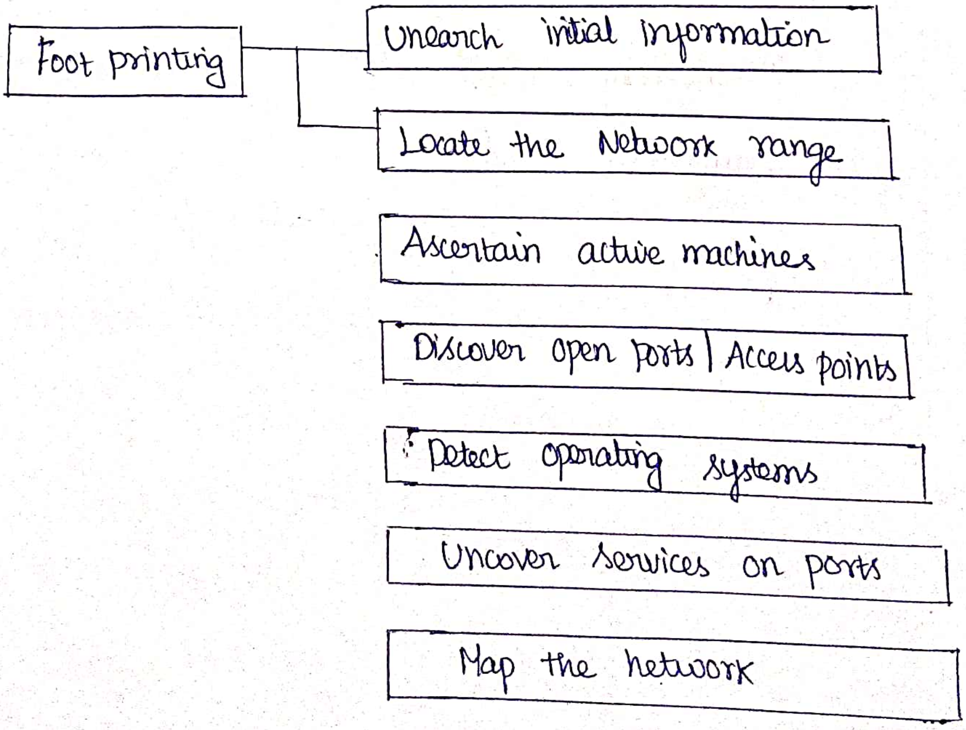
FOOT PRINTING AND PORT SCANNING

Information gathering methodology of hacker

Information gathering and getting to know the target systems is the first process in ethical hacking.

Reconnaissance is a set of processes and techniques (Footprinting, Scanning and enumeration) used to covertly discover and collect information about a target system.

During reconnaissance, an ethical hacker attempts to gather as much information about a target system as possible, following the seven steps listed below



Foot printing
Unearch initial information
Locate the Network range
Ascertain active machines
Discover open ports / Access points
Detect operating systems
Uncover services on ports
Map the network

There are two types of Reconnaissance

i) Active Reconnaissance

⇒ In this process you will directly interact with the computer system to gain information. The information can be relevant and accurate.

⇒ But there is a risk of getting detected if you are planning active reconnaissance without permission.

⇒ If you are detected, then system admin can take severe action against you and trail your subsequent activities.

ii) Passive Reconnaissance

⇒ In this process, you will not be directly connected to a computer system. This process is used to gather essential information without ever interacting with the target systems.

Foot Printing

Foot printing is the process of creating a blueprint or map of an organization's network ~~sys~~ and systems. Information gathering is also known as ~~footsri~~ footprinting an organization.

⇒ Foot printing begins by determining the target system, application or physical location of the target.

⇒ Once this information about the organization is gathered using nonintrusive method.

⇒ For example the organizations own web page may provide a personnel directory or a list of employee bios, which may prove using the hackers needs to use a social- Engineering attack to reach the objective.

⇒ The information the hacker is looking for during the footprinting phase is anything that gives clues as to the network architecture, server, and application types where valuable data is stored.

⇒ Here are some of the pieces of information to be gathered about a target during footprinting.

1. Domain name
2. Network blocks
3. Network Services and applications
4. System Architecture
5. Intrusion Detection System
6. Authentication mechanisms

7. Specific IP addresses

8. Access control mechanisms

9. Phone numbers

10. Contact addresses

There are two types of footprinting as following below

Active footprinting - Active footprinting means performing footprinting by getting in direct touch with the target machine.

Passive footprinting - Passive footprinting means collecting information about system located at a remote distance from the attacker.

Objectives of footprinting

1. Collect Network Information - Such as Domain name, Internal domain name, IP addresses, rogue website/ private websites within the domain, Access control mechanisms, protocols used, existing VPNs, authentication mechanisms, and system enumeration.

2, Collect System Information - such as users and group names, system banners, routing tables, and the routing protocols, SNMP information, System Architecture, operating system used, username and password.

3. Collect Organization's information : such as employee details, organizations website, company directory, local details, address & phone no, News, articles and press release.

Footprinting Tools

Footprinting can be done using hacking tools, either application or websites, which allow the hacker to locate Information passively.

By using these footprinting tools a hacker can gain some basic information on or "footprint" the target. Some of the common tools used for footprinting and Information gathering are as follows.

*Domain name lookup

* whois

* NSlookup

*Sam Spade

Footprinting a target

It is a part of the preparatory preattack Phase and involves accumulating data regarding a target's environment and architecture, usually for the purpose of finding ways to intrude into that environment.

Using google to gather information

⇒ A hacker may also do a google search or a yahoo! People search to locate information about employees or the organization itself.

⇒ The Use of google search engine to retrieve information has been termed google hacking.

⇒ Goto http://groups.google.com to search the Google newsgroups. The following commands can be used to have the Google search engine gather target information.

Site - Searches a specific website or domain. supply the ~~URL of the site after the colon~~. website you want to search after the colon.

filetype - Searches only within the text of a particular type of file.

link - Searches within hyperlinks for a search term and identifies linked pages.

Cache - Identifies the version of a webpage.

intitle - Searches for a term within the title of a document.

inurl - Searches only within the URL of a document.

For example, a hacker could use the following command to locate certain types of vulnerable web applications

INURL : ["Parameter="] with FILETYPE : [ext] and INURL : [Scriptname].

Scanning

After the reconnaissance and information gathering stages have been completed. Scanning is performed. During scanning, the hacker continues to gather information regarding network, and its individual host system. Information such as IP address, operating system, services and installed applications can help the hacker determine which type of exploit to use in hacking a system.

Types of scanning

Port scanning - Determines open ports and services

Network scanning - Identifies IP addresses on a given network or subnet.

Vulnerability scanning - Discovers presence of known weakness on target system.

1. Port scanning

Port scanning is the process of identifying open and available TCP/IP ports on a system. port scanning tools enable a hacker to learn about the services available in a given system.

Each service or application on a machine is associated with a well-known port number.

Port numbers are divided into three ranges.

⇒	well - known ports	: 0 - 1023
⇒	Registered ports	: 1024 - 49151
⇒	Dynamic ports	: 49152 - 65535

Common port numbers for the following applications

FTP, 21

Telnet, 23

Secure Shell, 22

HTTP, 80

HTTPS, 443

SMTP, 25

POP3, 110

additional port numbers

LDAP Server (TCP|UDP), 389

LDAP SSL (TCP|UDP), 636

NAT-T (UDP), 4500

RPC (TCP), 135

ASP.NET Session State (TCP), 42424

SMTP (TCP/UDP), 25

Rpc ~~(TCP~~

IMAP (TCP), 143

Network Scanning

It is a procedure for identifying active hosts on a network, either to attack them or as a network security assessment.

Hosts are identified by their individual IP addresses.

Network scanning tools attempt to identify all the live hosts on the network and their corresponding IP addresses.

Vulnerablity Scanning

It is the process of identifying the Vulnerablities of computer systems on a network.

Generally it identifies the operating system and version number, including service packs that may be installed.

Scanning can quickly identify which hosts are listening and active on a network, It is also a quick way to be Identified by an intrusion detection system (IDS).

Scanning tools probes can be recognized by most security intrusion detection tools.

Network and Vulnerablity scanning can usually be detected as well, because the scanner must interact with the target system over the network.

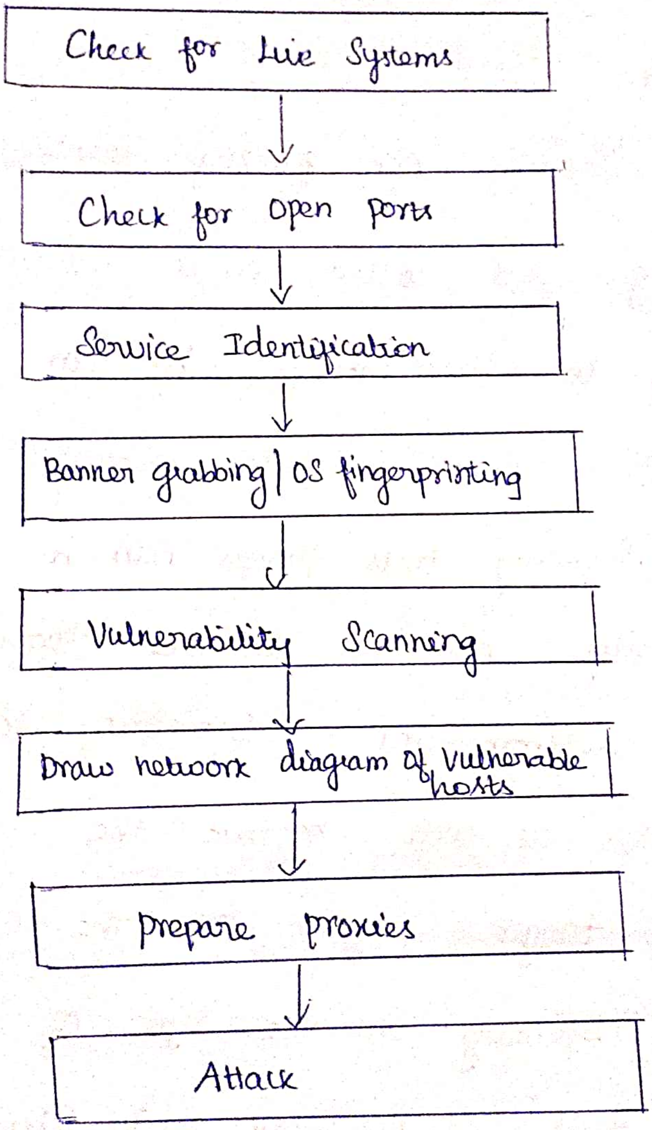
Depending on the type of scanning applications and the speed of the scan, an IDS will detect scanning and flag it as an IDS event.

As an ethical hacker their job is to gather as much information as possible and try and remain undetected.

The CEH Scanning Methodology

As a CEH, you're expected to familiar with the scanning methodology presented in the below figure.

This methodology is the process by which a hacker scans the network.



Ping Sweep Techniques

Ping sweep is also known as Internet Control Message protocol (ICMP) Scanning, as ICMP is a protocol used by the ping command.

The simplest, although not necessarily the most accurate, way to determine whether systems are live is to perform a ping sweep of the IP address range.

All system that respond with a ping reply are considered live on the network.

ICMP Scanning or a ping sweep is the process of sending an ICMP request or ping to all hosts on the network to determine which ones are up and responding to pings.

A benefit of ICMP Scanning is that it can be run in parallel, meaning all systems are scanned at the same time.

Ping sweep, which essentially performing an ICMP request to every host on the network. Systems that respond with the ping response are alive and listening on the network.

Hacking Tools

Pinger, Friendly pinger, and WS-Ping-Pro are all tools that perform ICMP queries.

To use windows ping

TO use the built in ping command in windows to test Connectivity to another system.

1. Open a command prompt in windows
2. Type a Ping www. microsoft.com

Detecting ping sweeps

Almost any IDS (or) Intrusion prevention system will detect and alert the security administrator to a ping sweep occurring on the network.

Most firewall and proxy servers block ping responses so a hacker Can't accurately determine whether systems are available using a ping sweep. alone.

More intense port scanning must be used if systems don't respond to a ping sweep.

Port - Scan Countermeasures

Countermeasures are processes or toolsets used by security administrators to detect and possibly thwart port scanning of hosts on their network.

⇒ Proper security architecture, such as implementation of IDS and firewalls, should be followed.

⇒ Ethical hackers use their toolset to test the Scanning countermeasures that have been implemented.

* Once a firewall is in place, a port-scanning tool should be run against hosts on the network to determine whether the firewall correctly detects and stops the port-scanning activity.

⇒ The firewall should be able to detect the probes sent by port-scanning tools. The firewall should carry out stateful inspections, which means it examines the data of the packet and not just the TCP header to determine whether the traffic is allowed to pass through the firewall.

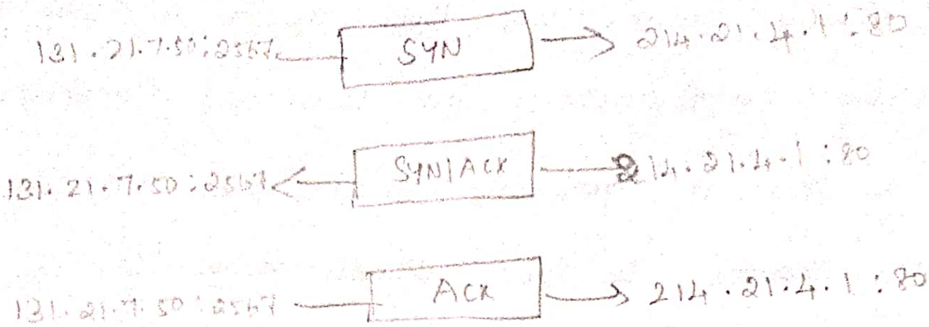
⇒ Network IDS should be kept open. The rest should be filtered or blocked.

⇒ The staff of the organization using the systems should be given appropriate training on security awareness.

TCP Communication flag Types

TCP scan types are built on the TCP three way handshake.

TCP connections require a three-way handshake before a connection can be made and data transferred between the sender and receiver.



To complete the threeway handshake and make a successful connection between two hosts, the sender must send a TCP packet with the synchronize (SYN) bit set.

Then the receiving system responds with a TCP packet with the synchronize (SYN) and acknowledge (ack) bit set to indicate the host is ready to receive data.

The source system sends a final Packet with the Ack bit set to indicate the connection is complete and data is ready to be sent.

TCP is a connection oriented protocol, a process for establishing a connection (three-way handshake), restarting a failed connection, and finishing a connection is the part of the Protocol. These protocol notifications are called flags.

TCP contains ACK, RST, SYN, URG, PSH and FIN Flags.

SYN - Synchronize. Initiates a connection between hosts.

ACK - Acknowledge. Established a connection between hosts.

PSH - Push. System is forwarding buffered data.

URG - Urgent. Data in packets must be processed quickly.

FIN - Finish. No more transmissions.

RST - Reset. Resets the connections.

TCP Scan types

	TCP Scan		flags sent by hacker
1.	XMAS Scan	–	All flags set (ACK, RST, SYN, URG, PSH, FIN)
2.	FIN Scan	–	FIN
3.	NULL Scan	–	No flags set
4	TCP connect / full-open scan	–	SYN, then ACK
5.	SYN Scan / half-open scan	–	SYN, then RST.

nmap Command switches

Nmap is a free, open source tool that quickly and efficiently performs ping sweeps, port scanning, service identification, IP address detection, and operating system detection.

NMAP has the benefit of scanning a large number of machines in a single session.

The state of the port as determined by an nmap scan can be open, filtered (or) unfiltered.

Open means that the target machine accepts incoming request on the port.

Filtered means a firewall or Network filter is screening the port and preventing nmap from discovering whether its open.

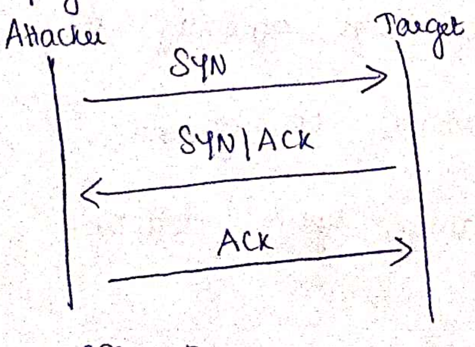
Unfiltered (or) Closed means the port is determined to be closed, and no firewall or filter is interfering with the nmap requests.

NMAP Scan type

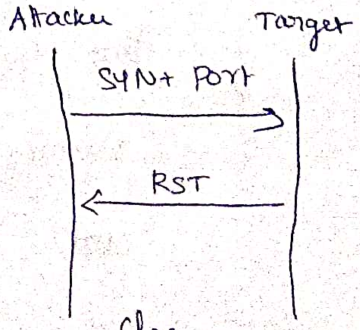
1. TCP Connect

The attacker makes a full TCP connection to the target system. The most reliable scan type but also the most detectable.

Open ports reply with a SYN|ACK while closed Ports reply with a RST|ACK.



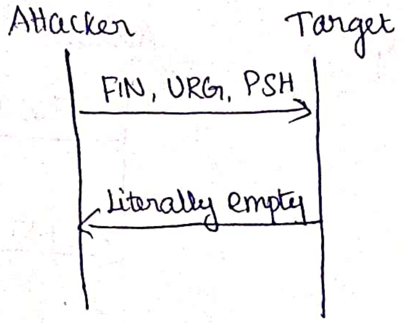
open port



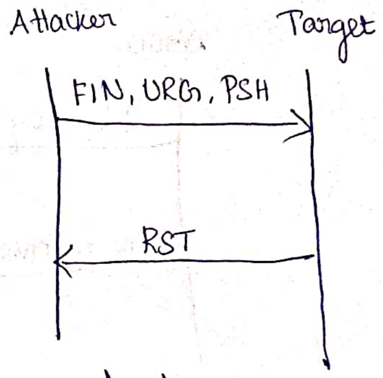
Close port

2. XMAS tree Scan

The attacker checks for TCP services by sending XMAS-tree packets, which are named as such because all the "lights" are on, meaning the FIN, URG, and PSH flags are set. Closed port reply with a RST flage.



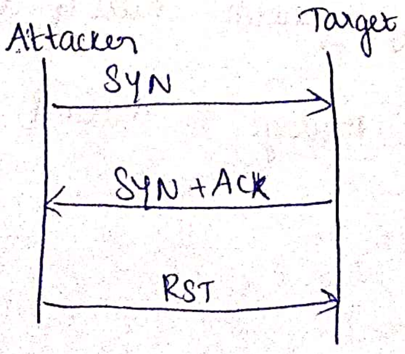
open port

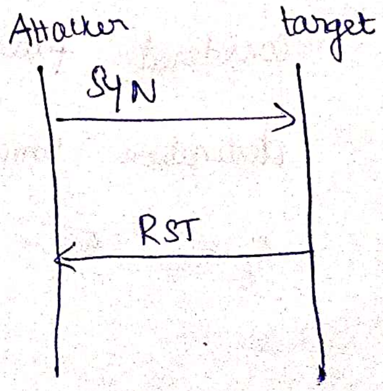


closed port

3. SYN Stealth Scan

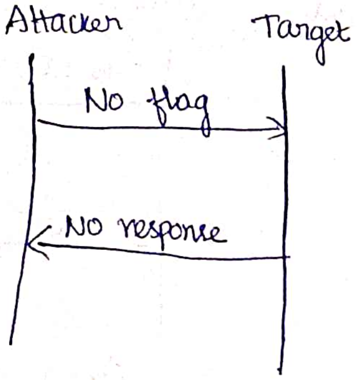
This is also known as half-open scanning. The hacker sends a SYN packet and receives a SYN-ACK back from the server. It's stealthy ~~SYN~~. because a full TCP connection isn't opened. Open ports reply with a SYN/ACK while closed ports reply with a RST/ACK.



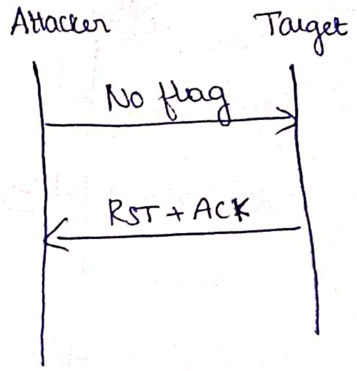


4. NULL Scan

This is an advanced scan that may be able to pass through firewalls undetected or modified. NULL Scan ~~was~~ has all flags off or not set. It only works on Unix systems. Closed ports will return a RST flag



Open port



Closed port

5. Windows Scan

This type of scan is similar to the ACK scan and also detect open ports.

6. ACK Scan

This type of scan is used to map out firewall rules. ACK scan only works on Unix. The port is considered filtered by firewall rules if an ICMP destination unreachable message is received as a result of the ACK scan.

nmap Command Switch

nmap Command switch	Scan performed
-ST	TCP Connect Scan
-sS	SYN Scan
-SF	FIN Scan
-SX	XMAS Tree Scan
-SN	Null Scan
-SP	Ping Scan
-SU	UDP Scan
-SO	Protocol scan
-SA	Ack Scan
-SW	window scan
-sL	List | DNS scan
-SR	RPC scan
-SI	IDLE Scan
-PO	Don't Ping
-PB	TCP and ICMP ping
-PT	TCP ping
-PI	ICMP Ping
-Ps	SYN Ping
-T Paranoid	Serial scan: 300 sec between Scan
-T Normal	Parallel scan
-T Aggressive	Parallel scan: 300 sec timeout and 1.25 sec | Probe
-T Insane	parallel scan, 75 Sec timeout and 3 sec | probe

Enumeration

Enumeration occurs after scanning and ~~its~~ is the process of gathering and compiling user names, machine names, network resources, shares, and services.

It also refers to actively querying or connecting to a target system to acquire this information.

Hackers need to be methodical in their approach to hacking.

Examples of hacker might perform hacking a target system.

1. Extract usernames and enumeration.
2. Gather information about the host using null sessions.
3. Perform windows enumeration using the Superscan tool
4. Acquire the user accounts using the tool GetAcct
5. perform SNMP port Scanning.

UNIT 3

SYSTEM HACKING

The Simplest Way to Get a Password

Many hacking attempts start with getting a password to a target system. Passwords are the key piece of information needed to access a system, and users often select passwords that are easy to guess. Many reuse passwords or choose one that's simple—such as a pet's name—to help them remember it. Because of this human factor, most password guessing is successful if some information is known about the target. Information gathering and reconnaissance can help give away information that will help a hacker guess a user's password. Once a password is guessed or cracked, it can be the launching point for escalating privileges, executing applications, hiding files, and covering tracks. If guessing a password fails, then passwords may be cracked manually or with automated tools such as a dictionary or brute-force method.

Types of Passwords

Several types of passwords are used to provide access to systems. The characters that form a password can fall into any of these categories:

- Only letters
- Only numbers
- Only special characters Types of Passwords
- Letters and numbers
- Only letters and special characters
- Only numbers and special characters
- Letters, numbers, and special characters

A strong password is less susceptible to attack by a hacker. The following rules, proposed by the EC-Council, should be applied when you're creating a password, to protect it against attacks:

- Must not contain any part of the user's account name
- Must have a minimum of eight characters
- Must contain characters from at least three of the following categories:
  - Nonalphanumeric symbols ($,:"%@!#)
  - Numbers
  - Uppercase letters
  - Lowercase letters

A hacker may use different types of attacks in order to identify a password and gain further access to a system.

Aspect of remote password guessing

Password guessing is the process of attempting to gain access to a system through the systematic guessing of passwords (and at times also usernames) in an attempt to gain a login to a target system. This is problematic in that it will generally create voluminous amounts of both network traffic when conducted remotely and system logs.

remote password guessinginclude the following task:

- Identify publicly-accessible services/applications that request username/password credentials and attempt bypassing them via manual guessing. Keep an eye out for account lock-out mechanisms.
- Query Google and examine your public website to identify possible usernames. (The Backtrack CD has some nice tools for that.)

- Compile a list of possible passwords the users might use, accounting for your organization's location, name, and industry-specific terminology. Add common names and words like "passsword" to the list. I find that having a short, but intelligently-crafted list is more effective than using a 100KB dictionary file (the long file often takes too long to cycle through remotely).
- After trying the manual route, make use of an automated password guessing tool to see whether it can guess logon credentials using the short password list you put together. Hydra is an excellent tool for this task. It's free, fast, and effective, even though it's poorly documented. (Anyone feels like writing a comprehensive guide to using Hydra, or pointing us to one that already exists?) Hydra is included on the above-mentioned Backtrack CD, and supports most of the protocols you're likely to encounter in the field.

The types of ***password attacks*** are as follows:

- **Passive Online**
  - ➢ Eavesdropping on network password exchanges.
  - ➢ Passive online attacks include sniffing, man-in-the-middle, and replay attacks.
- **Active Online**
  - ➢ Guessing the Administrator password.
  - ➢ Active online attacks include automated password guessing.
- **Offline**
  - ➢ Dictionary, hybrid, and brute-force attacks.
- **Nonelectronic**
  - ➢ Shoulder surfing, keyboard sniffing, and social engineering.
  - ➢ We'll look at each of these attacks in more detail in the following sections.

1.Passive Online Attacks

A passive online attack is also known as ***sniffing*** the password on a wired or wireless network. A passive attack is not detectable to the end user. The password is captured during the authentication process and can then be compared against a dictionary file or word list. User account passwords are commonly hashed or encrypted when sent on the network to prevent unauthorized access and use. If the password is protected by encryption or hashing, special tools in the hacker's toolkit can be used to break the algorithm.

Another passive online attack is known as ***man-in-the-middle (MITM).*** In a MITM attack, the hacker intercepts the authentication request and forwards it to the server. By inserting a sniffer between the client and the server, the hacker is able to sniff both connections and capture passwords in the process.

A***replay attack*** is also a passive online attack; it occurs when the hacker intercepts the password en route to the authentication server and then captures and resends the authentication packets for later authentication. In this manner, the hacker doesn't have to break the password or learn the password through MITM but rather captures the password and reuses the password-authentication packets later to authenticate as the client.

2. Active Online Attacks

The easiest way to gain administrator-level access to a system is to guess a simple password assuming the administrator used a simple password. Password guessing is an active online attack. It relies on the human factor involved in password creation and only works on weak passwords.

The Enumeration phase of system hacking, you learned the vulnerability of NetBIOS enumeration and null sessions. Assuming that the NetBIOS TCP 139 port is open, the most effective method of breaking into a Windows NT or Windows 2000 system is password guessing. This is done by attempting to connect to an enumerated share (IPC$ or C$) and trying a username and password combination. The most commonly used Administrator account and password combinations are words like Admin, Administrator, Sysadmin, or Password, or a null password.

A hacker may first try to connect to a default Admin$, C$, or C:\Windows share. To connect to the hidden C: drive share, for example, type the following command in the Run field (Start ➪ Run):

```
\\ip_address\c$
```

Automated programs can quickly generate dictionary files, word lists, or every possible combination of letters, numbers, and special characters and then attempt to log on using those credentials. Most systems prevent this type of attack by setting a maximum number of login attempts on a system before the account is locked.

In the following sections, we'll discuss how hackers can perform automated password guessing more closely, as well as countermeasures to such attacks.

Performing Automated Password Guessing

To speed up the guessing of a password, hackers use automated tools. An easy process for automating password guessing is to use the Windows shell commands based on the standard NET USE syntax. To create a simple automated password-guessing script, perform the following steps:

1. Create a simple username and password file using Windows Notepad. Automated tools such as the Dictionary Generator are available to create this word list. Save the file on the C: drive as credentials.txt. Types of Passwords 99

2. Pipe this file using the FOR command:

```
C:\> FOR /F “token=1, 2*” %i in (credentials.txt)
```

3. Type net use \\targetIP\IPC$ %i /u: %j to use the credentials.txt file to attempt to log on to the target system's hidden share.

Defending Against Password Guessing

Two options exist to defend against password guessing and password attacks. Both smart cards and biometrics add a layer of security to the insecurity that's inherent when users create their own passwords.

Both smart cards and biometrics use two-factor authentication, which requires two forms of identification (such as the actual smart card and a password) when validating a user. By requiring something the user physically has (a smart card, in this instance) and something the user knows (their password), security is increased, and the authentication process isn't susceptible to password attacks.

3. Offline Attacks

Offline attacks are performed from a location other than the actual computer where the passwords reside or were used. Offline attacks usually require physical access to the computer and copying the password file from the system onto removable media. The hacker then takes the file to another computer to perform the cracking. Several types of offline password attacks exist, as you can see in Table 4.1.

A ***dictionary attack***is the simplest and quickest type of attack. It's used to identify a password that is an actual word, which can be found in a dictionary. Most commonly, the attack uses a dictionary file of possible words, which is hashed using the same algorithm used by the authentication process. Then, the hashed dictionary words are compared with hashed passwords as the user logs on, or with passwords stored in a file on the server. The dictionary attack works only if the password is an actual dictionary word; therefore, this type of attack has some limitations. It can't be used against strong passwords containing numbers or other symbols.

A ***hybrid attack*** is the next level of attack a hacker attempts if the password can't be found using a dictionary attack. The hybrid attack starts with a dictionary file and substitutes numbers and symbols for characters in the password. For example, many users add the number 1 to the end of their password to meet strong password requirements. A hybrid attack is designed to find those types of anomalies in passwords.

The most time-consuming type of attack is a ***brute-force attack***, which tries every possible combination of uppercase and lowercase letters, numbers, and symbols. A brute-force attack is the slowest of the three types of attacks because of the many possible combinations of characters in the password. However, brute force is effective; given enough time and processing power, all passwords can eventually be identified.

A ***rainbow table*** is a list of dictionary words that have already been hashed. Rainbow tables can speed up the discovery and cracking of passwords by pre-computing the hashes for common strings of characters. For example, a rainbow table can include characters from a to z or A to Z. Essentially, rainbow table tools are hash crackers. A traditional brute-force cracker will try all possible plaintext passwords one by one in order. It is time consuming to break complex passwords in this way. The idea of rainbow tables is to do all cracking-time computation in advance.

4. Nonelectronic Attacks

Nonelectronic—or nontechnical attacks—are attacks that do not employ any technical knowledge. This kind of attack can include social engineering, shoulder surfing, keyboard sniffing, and dumpster diving.

Social engineering is the art of interacting with people either face to face or over the telephone and getting them to give out valuable information such as passwords. Social engineering relies on people's good nature and desire to help others. Many times, a help desk is the target of a social-engineering attack because their job is to help people—and recovering or resetting passwords is a common function of the help desk. The best defense against social-engineering attacks is security-awareness training for all employees and security procedures for resetting passwords.

Various methods of password cracking

Manual password cracking involves attempting to log on with different passwords. The
hacker follows these steps:

1. Find a valid user account (such as Administrator or Guest).
2. Create a list of possible passwords.
3. Rank the passwords from high to low probability.
4. Key in each password.
5. Try again until a successful password is found.

A hacker can also create a script file that tries each password in a list. This is still considered manual cracking, but it’s time consuming and not usually effective. A more efficient way of cracking a password is to gain access to the password file on asystem. Most systems hash (one-way encrypt) a password for storage on a system. During the logon process, the password entered by the user is hashed using the same algorithm and then compared to the hashed passwords stored in the file. A hacker can attempt to gain access to the hashing algorithm stored on the server instead of trying to guess or otherwise identify the password. If the hacker is successful, they can decrypt the passwords stored on the server

Hacking tools

1. Legion automates the password guessing in NetBIOS sessions. Legion scans multiple IP address ranges for Windows shares and also offers a manual dictionary attack tool.

2. NTInfoScan is a security scanner for NT 4.0. This vulnerability scanner produces an HTML-based report of security issues found on the target system and other information.

3. L0phtCrack is a password auditing and recovery package distributed by @stake software, which is now owned by Symantec. It performs Server Message Block (SMB) packet captures on the local network segment and captures individual login sessions. L0phtCrack contains dictionary, brute-force, and hybrid attack capabilities. Symantec has recently stopped development of the L0phtCrack tool, but it can still be found on the Internet.

4. LC5 is another good password cracking tool. LC5 is a suitable replacement for L0phtCrack.

5. John the Ripper is a command-line tool designed to crack both Unix and NT passwords. The cracked passwords are case insensitive and may not represent the real mixed-case password.

6. KerbCrack consists of two programs: kerbsniff and kerbcrack. The sniffer listens on the network and captures Windows 2000/XP Kerberos logins. The cracker can be used to find the passwords from the capture file using a brute-force attack or a dictionary attack.

Understanding the LAN Manager Hash

Windows 2000 uses NT LAN Manager (NTLM) hashing to secure passwords in transit on the network. Depending on the password, NTLM hashing can be weak and easy to break. For example, let’s say that the password is 123456abcdef. When this password is encrypted with the NTLM algorithm, it’s first converted to all uppercase: 123456ABCDEF. The password is padded with null (blank) characters to make it 14 characters long: 123456ABCDEF__. Before the password is encrypted, the 14-character string is split in half: 123456A and BCDEF__. Each string is individually encrypted, and the results are concatenated:

123456A = 6BF11E04AFAB197F

BCDEF__ = F1E9FFDCC75575B15

The hash is 6BF11E04AFAB197FF1E9FFDCC75575B15

Cracking Windows 2000 Passwords

The SAM file in Windows contains the usernames and hashed passwords. It's located in the Windows\system32\config directory. The file is locked when the operating system is running so that a hacker can't attempt to copy the file while the machine is booted to Windows. One option for copying the SAM file is to boot to an alternate operating system such as DOS or Linux with a boot CD. Alternately, the file can be copied from the repair directory. If a system administrator uses the RDISK feature of Windows to back up the system then a compressed copy of the SAM file called SAM._ is created in C:\windows\repair. To expand this file, use the following command at the command prompt:

```
C:\>expand sam._ sam
```

After the file is uncompressed, a dictionary, hybrid, or brute-force attack can be run against the SAM file using a tool like L0phtCrack. A similar tool to L0phtcrack is Ophcrack.

Redirecting the SMB Logon to the Attacker

Another way to discover passwords on a network is to redirect the Server Message Block (SMB) logon to an attacker's computer so that the passwords are sent to the hacker. In order to do this, the hacker must sniff the NTLM responses from the authentication server and trick the victim into attempting Windows authentication with the attacker's computer. A common technique is to send the victim an email message with an embedded link to a fraudulent SMB server. When the link is clicked, the user unwittingly sends their credentials over the network.

SMBRelay An SMB server that captures usernames and password hashes from incoming SMB traffic. SMBRelay can also perform man-in-the-middle (MITM) attacks.

SMBRelay2 Similar to SMBRelay but uses NetBIOS names instead of IP addresses to capture usernames and passwords.

pwdump2 A program that extracts the password hashes from a SAM file on a Windows system. The extracted password hashes can then be run through L0phtCrack to break the passwords.

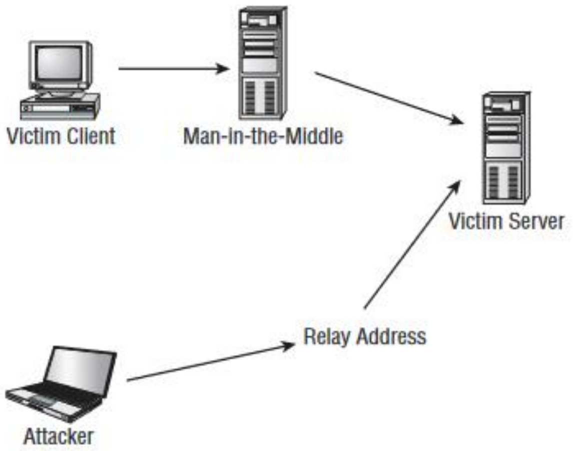
Samdump Another program that extracts NTLM hashed passwords from a SAM file.

C2MYAZZ A spyware program that makes Windows clients send their passwords as cleartext. It displays usernames and their passwords as users attach to server resources.

SMB Relay MITM Attacks and Countermeasures

An SMB relay MITM attack is when the attacker sets up a fraudulent server with a relay address. When a victim client connects to the fraudulent server, the MITM server intercepts the call, hashes the password, and passes the connection to the victim server. Figure 4.1 illustrates such an attack.

FIGURE 4.1 SMB relay MITM attack



NetBIOS DoS Attacks

A NetBIOS denial-of-service (DoS) attack sends a NetBIOS Name Release message to the NetBIOS Name Service on a target Windows systems and forces the system to place its name in conflict so that the name can no longer be used. This essentially blocks the client from participating in the NetBIOS network and creates a network DoS for that system.

Another way to create a more secure and memorable password is to follow a repeatable pattern, which will enable to password to be re-created when needed.

1. Start with a memorable phrase, such as
   Maryhadalittlelamb
2. Change every other character to uppercase, resulting in
   MaRyHaDaLiTtLeLaMb
3. Change a to @ and i to 1 to yield
   M@RyH@D@L1TtLeL@Mb
4. Drop every other pair to result in a secure repeatable password or
   M@H@L1LeMb

Now you have a password that meets all the requirements, yet can be “remade” if necessary.

Password-Cracking Countermeasures

- The strongest passwords possible should be implemented to protect against password cracking.
- Systems should enforce 8–12-character alphanumeric passwords.
- To protect against cracking of the hashing algorithm for passwords stored on the server, you must take care to physically isolate and protect the server.
- The system administrator can use the SYSKEY utility in Windows to further protect hashes stored on theserver’s hard disk.

A system administrator can implement the following security precautions to decrease the effectiveness of a ***brute-force password-cracking*** attempt:

- Never leave a default password.
- Never use a password that can be found in a dictionary.
- Never use a password related to the hostname, domain name, or anything else that can be found with Whois.
- Never use a password related to your hobbies, pets, relatives, or date of birth.
- As a last resort, use a word that has more than 21 characters from a dictionary as aÛN password.

Keystroke Loggers

Keyloggersare many hackers and script kiddie's favorite tools. Keylogging is a method that was first imagined back in the year 1983. Around then, the utilization of this product was uncommon and just the top examination organizations and spies could get their hands on it, yet today, it is a typical element offered by most government operative applications like TheOneSpy. Individuals use it as an opportunity to guarantee the assurance of their families, organizations, and the ones they care about.

Keylogger is a software that records each and every keystroke you enter, including mouse clicks. Hardware keyloggers are also available which will be inserted between keyboard and CPU. It provides the following ***features:***

1. It takes a minute to install this software/hardware in the victim's system, from the next second onwards attacker will get every activity going on in the victim computer.
2. Each and every activity happening in the victim's system with screenshots will be recorded. This activity will be saved in the victim's system or it can be mailed to the attacker email or can be uploaded to the FTP server. Wondered? Let's see how attackers do this along with protection techniques.
3. Keylogging highlight of spy applications is adept at recording each and every keystroke made by utilizing a console, regardless of whether it is an on-screen console.
4. It likewise takes a screen capture of the screen when the client is composing (Usually this screen capture is taken when a catch on the mouse is clicked).
5. It works watchfully, escaped the client's view, for example, the focused on the client could never discover that all his keystrokes are being recorded.
6. Keyloggers recorder can record writings, email, and any information you compose at whatever point using your support.
7. The log record made by the keyloggers would then have the option to be sent to a predefined gatherer.
8. Some keyloggers tasks will likewise record any email that tends to your use and Web website URLs you visit.

Some software keyloggers code can capture additional information without requiring any keyboard key presses as input. They include:

1. **Clipboard logging:** Anything duplicated to the clipboard is caught.
2. **Screen logging:** Randomly coordinated screen captures of your PC are logged.
3. **Control text capture:** The Windows API allows for programs to request the text value of some controls, it means a password can still be captured albeit it is behind a password mask.
4. **Activity tracking:** Recording of which programs, folders, and windows are opened and also the screenshots of every.
5. Recording of program queries, instant message conversations, FTP downloads alongside the other internet activities.

Types of Keylogger

There are basically two types of Keyloggers:

1. ***Hardware Keylogger:***This is a thumb-size device. It records all the keystrokes you enter from the keyboard then saves it in its memory. Later this data will be analyzed. The drawback of this device is, It can't record mouse clicks, can't take screenshots, and even can't email, more importantly, It requires physical access to the machine. Hardware Keylogger is advantageous because it's not hooked into any software nor can it's detected by any software.
2. ***Software Keylogger:***Software Keylogger can be installed in the victim's system even if they use updated Antivirus. There are lots of software available in market which make a Keylogger undetectable by latest antivirus, we are going to study about them too in upcoming chapters. There are many keyloggers available in market with various features. Some examples of Software Keyloggers are:
    1. RevealerKeylogger
    2. ArdamaxKeylogger
    3. WinSpy
    4. Invisible Keylogger
    5. RefogKeylogger
    6. Activity Keylogger
    7. Keystroke Keyloggers

How to Detect and Remove Keylogger?

1. choose the best **Antivirus**, to detect a Keylogger on your system. There is some specific sort of AV dedicated for such scans.
2. Press**Ctrl+Alt+Delete**to check the task list on your computer. Examine the tasks running, and if you're unacquainted any of them, look them abreast of an inquiry engine.
3. Scan your hard disc for the foremost recent files stored. Look at the contents of any files that **often update**, as they could be logs.
4. Use your system configuration utility to look at which programs are loaded at computer start-up. Access this list by typing "**msconfig**" into the run box.

Pros of Keylogger

1. Monitor Every Keystroke Made.
2. Protect Confidential Information.
3. Safety Concerns.

Cons of Keylogger

1. Zero Privacy.
2. Release of Sensitive Information.
3. Gives Keylogging Service Providers Free Reign.

Understanding Sniffers

A sniffer is a packet-capturing or frame-capturing tool. It basically captures and displays the data as it is being transmittedfrom host to host on the network. Generally a sniffer intercepts traffic on the network and displays it in either a command-line or GUI format for ahacker to view. Most sniffers display both the Layer 2 (frame) or Layer 3 (packet) headers and the data payload. Some sophisticated sniffers interpret the packets and can reassemble the packet stream into the original data, such as an email or a document. Sniffers are used to capture traffic sent between two systems, but they can also provide alot of other information. Depending on how the sniffer is used and the security measures inplace, a hacker can use a sniffer to discover usernames, passwords, and other confidentialinformation transmitted on the network. Several hacking attacks and various hacking toolsrequire the use of a sniffer to obtain important information sent from the target system.

How a Sniffer Works

Sniffer software works by capturing packets not destined for the sniffer system's MAC address but rather for a target's destination MAC address. This is known as promiscuous mode. Normally, a system on the network reads and responds only to traffic sent directly to its MAC address. However, many hacking tools change the system's NIC to promiscuous mode. In promiscuous mode, a NIC reads all traffic and sends it to the sniffer for processing. Promiscuous mode is enabled on a network card with the installation of special driver software. Many of the hacking tools for sniffing include a promiscuous-mode driver to facilitate this process. Not all Windows drivers support promiscuous mode, so when using hacking tools ensure that the driver will support the necessary mode.

Any protocols that don't encrypt data are susceptible to sniffing. Protocols such asHTTP, POP3, Simple Network Management Protocol (SNMP), and FTP are most commonly captured using a sniffer and viewed by a hacker to gather valuable information suchas usernames and passwords.

Comprehending Active and Passive Sniffing

What is Sniffing?

Sniffing is a process of monitoring and capturing all data packets passing through given network. Sniffers are used by network/system administrator to monitor and troubleshoot network traffic. Attackers use sniffers to capture data packets containing sensitive information such as password, account information etc. Sniffers can be hardware or software installed in the system. By placing a packet sniffer on a network in promiscuous mode, a malicious intruder can capture and analyze all of the network traffic.

There are two types:

- Passive Sniffing
- Active Sniffing

Active Sniffing:

Sniffing in the switch is active sniffing. A switch is a point to point network device. The switch regulates the flow of data between its ports by actively monitoring the MAC address on each port, which helps it pass data only to its intended target. In order to capture the traffic between target sniffers has to actively inject traffic into the LAN to enable sniffing of the traffic. This can be done in various ways.

Passive Sniffing:

This is the process of sniffing through the hub. Any traffic that is passing through the non-switched or unbridged network segment can be seen by all machines on that segment. Sniffers operate at the data link layer of the network. Any data sent across the LAN is actually sent to each and every machine connected to the LAN. This is called passive since sniffers placed by the attackers passively wait for the data to be sent and capture them.

Passive sniffing involves listening and capturing traffic, and is useful in a network connected by hubs; active sniffing involves launching an Address Resolution Protocol (ARP) spoofing or traffic-floodingattack against a switch in order to capture traffic. As the names indicate, active sniffing isdetectable but passive sniffing is not detectable.

In networks that use hubs or wireless media to connect systems, all hosts on the network can see all traffic; therefore, a passive packet sniffer can capture traffic going to and from all hosts connected via the hub. A switched network operates differently. The switch looks at the data sent to it and tries to forward packets to their intended recipients based on MAC address. The switch maintains a MAC table of all the systems and the port numbers to which they're connected. This enables the switch to segment the network traffic and send traffic only to the correct destination MAC addresses. A switch network has greatly improved throughput and is more secure than a shared network connected via hubs. Another way to sniff data through a switch is to use a span port or port mirroring to enable all data sent to a physical switch port to be duplicated to another port. In many cases, span ports are used by network administrators to monitor traffic for legitimate purposes.

SniffingCountermeasures

The best security defense against a sniffer on the network is encryption. Although encryption won't prevent sniffing, it renders any data captured during the sniffing attack useless because hackers can't interpret the information. Encryption such as AES and RC4 or RC5 can be utilized in VPN technologies and is commonly used to prevent sniffing on a network.

ARP Spoofing and Redirection

Address Resolution Protocol (ARP) is a stateless protocol used for resolving IP addresses to machine MAC addresses. All network devices that need to communicate on the network broadcast ARP queries in the system to find out other machines' MAC addresses. ARP Poisoning is also known as **ARP Spoofing**.

Here is how ARP works −

- When one machine needs to communicate with another, it looks up its ARP table.
- If the MAC address is not found in the table, the **ARP_request** is broadcasted over the network.
- All machines on the network will compare this IP address to MAC address.
- If one of the machines in the network identifies this address, then it will respond to the **ARP_request** with its IP and MAC address.
- The requesting computer will store the address pair in its ARP table and communication will take place.

What is ARP Spoofing?

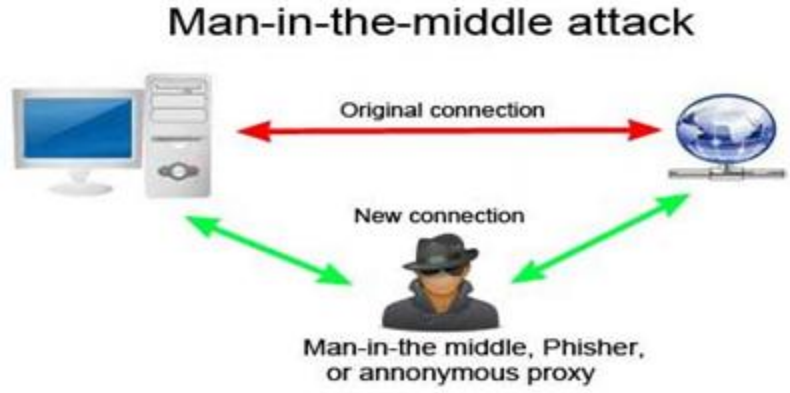
ARP packets can be forged to send data to the attacker's machine.

- ARP spoofing constructs a large number of forged ARP request and reply packets to overload the switch.
- The switch is set in **forwarding mode** and after the **ARP table** is flooded with spoofed ARP responses, the attackers can sniff all network packets.

Attackers flood a target computer ARP cache with forged entries, which is also known as **poisoning**. ARP poisoning uses Man-in-the-Middle access to poison the network.

What is MITM?

The Man-in-the-Middle attack (abbreviated MITM, MitM, MIM, MiM, MITMA) implies an active attack where the adversary impersonates the user by creating a connection between the victims and sends messages between them. In this case, the victims think that they are communicating with each other, but in reality, the malicious actor controls the communication.



A third person exists to control and monitor the traffic of communication between two parties. Some protocols such as **SSL** serve to prevent this type of attack.

ARP Spoofing and Poisoning Countermeasures

To prevent ARP spoofing, permanently add the MAC address of the gateway to the ARP cache on a system. You can do this on a Windows system by using the ARP -s command at the command line and appending the gateway's IP and MAC addresses. Doing so prevents a hacker from overwriting the ARP cache to perform ARP spoofing on the system but can be difficult to manage in a large environment because of the number of systems. In an enterprise environment, port-based security can be enabled on a switch to allow only one MAC address per switch port.

DNS and IP Sniffing

This is a technique that tricks a DNS server into believing it has received authentic information when in reality it hasn't. Once the DNS server has been poisoned, the information is generally cached for a while, spreading the effect of the attack to the users of the server. When a user requests a certain website URL, the address is looked up on a DNS server to find the corresponding IP address. If the DNS server has been compromised, the user is redirected to a website other than the one that was requested, such as a fake website.

To perform a DNS attack, the attacker exploits a flaw in the DNS server software that can make it accept incorrect information. If the server doesn't correctly validate DNS responses to ensure that they come from an authoritative source, the server ends up caching the incorrect entries locally and serving them to users that make subsequent requests.

This technique can be used to replace arbitrary content for a set of victims with content of an attacker's choosing. For example, an attacker poisons the IP address's DNS entriesfor a target website on a given DNS server, replacing them with the IP address of a server the hacker controls. The hacker then creates fake entries for files on this server with names matching those on the target server. These files may contain malicious content, such as a worm or a virus. A user whose computer has referenced the poisoned DNS server is tricked into thinking the content comes from the target server and unknowingly downloads malicious content.

The **types of DNS spoofing** techniques are as follows:

1. ***Intranet Spoofing*** Acting as a device on the same internal network

2.Internet Spoofing Acting as a device on the Internet

3. Proxy Server DNS Poisoning Modifying the DNS entries on a proxy server so the user is redirected to a different host system

4. DNS Cache Poisoning Modifying the DNS entries on any system so the user is redirected to a different host

HTTPS Sniffing

HTTP sniffer is an application that monitors traffic data to and from a computer network link. It can be an independent software application or hardware device equipped with the relevant firmware and software.

Sniffers exist in a variety of platforms including both commercial and open source versions. Some sniffers can only intercept data from TCP/IP protocols but the more complex ones even capture and decode data packets for the more secure SSL /HTTPS protocol that use asymmetric cryptography.

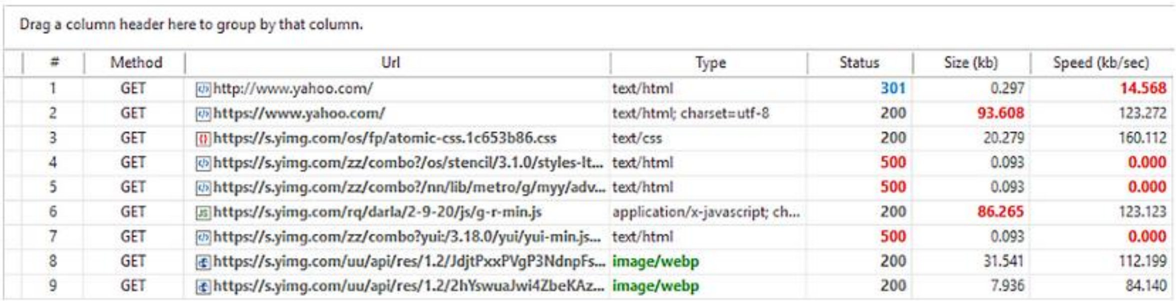
Types of Sniffers

Sniffers come in a variety of forms and the major ones include online, proxy, and application sniffers.

Online HTTP sniffers are limited to basic analyzing of a particular webpage. You provide a link to check and they respond with the HTTP header and HTTP content of the requested page. Online sniffers can be used to quickly check the basic server settings and see the source code of the requested page.

Proxy sniffers monitor all traffic between internet applications, including your web browser and the web on certain protocols like HTTP or HTTPS. These HTTP sniffers can only operate with applications configured to use proxy servers and unlike other sniffers, these may have an effect on the traffic. Proxy HTTP sniffer may decode SSL / HTTPS traffic, but developers need to install a special self-signed root certificate issued by the proxy vendor. This certificate is needed to implement the Man-in-the-middle technique for decrypting SSL.

Application sniffers are the most powerful ones. They are standalone applications that work on a developer’s computer and have the ability to capture network traffic for all protocols. A packet sniffer can even capture data packets from other computers in the same network. Application HTTP sniffer is not limited to Man-in-the-middle technique when decrypting the SSL / HTTPS traffic as a proxy sniffer. It may use API hooks for decoding SSL and don’t require root certificate to operate. But API hooks have limited usage, for example, API hooks don't work with recent versions of Google Chrome or Opera. HTTP Debugger is an example of application HTTP sniffer. Versions prior to v5.0 were using API Hooks for decoding SSL while new versions use Man-in-the-middle technique.



Drag a column header here to group by that column.

#	Method	Url	Type	Status	Size (kb)	Speed (kb/sec)
1	GET	http://www.yahoo.com/	text/html	301	0.297	14.568
2	GET	https://www.yahoo.com/	text/html; charset=utf-8	200	93.608	123.272
3	GET	https://s.yimg.com/os/fp/atomic-css.1c653b86.css	text/css	200	20.279	160.112
4	GET	https://s.yimg.com/zz/combo?/os/stencil/3.1.0/styles-lt...	text/html	500	0.093	0.000
5	GET	https://s.yimg.com/zz/combo?/nn/lib/metro/g/myy/adv...	text/html	500	0.093	0.000
6	GET	https://s.yimg.com/rq/darla/2-9-20/js/g-r-min.js	application/x-javascript; ch...	200	86.265	123.123
7	GET	https://s.yimg.com/zz/combo?yui:/3.18.0/yui/yui-min.js...	text/html	500	0.093	0.000
8	GET	https://s.yimg.com/uu/api/res/1.2/JdjtPxxPVgP3NdnpFs...	image/webp	200	31.541	112.199
9	GET	https://s.yimg.com/uu/api/res/1.2/2hYswuaJwi4ZbeKAz...	image/webp	200	7.936	84.140

HTTP Sniffers for Developers

Modern browsers provide some basic information about website traffic usage (for example for Google Chrome you can see this information on Network tab of Web Inspector). This information, though, is limited and not easy to analyze.

HTTP sniffers, however, provide in-depth information on every aspect of loading a webpage and/or any web resource including but not limited to built-in HTML, CSS, JS, JSON, XML syntax highlighters, JSON and XML tree structure viewers, built-in image viewers, automatic server error and performance bottlenecks detecting and even website structure tree viewer. With some sniffers, you can modify and reply back modified HTTP requests to the web server to test it with various conditions in order to reproduce and fix website errors.

Some advanced HTTP sniffers can visualize your traffic in the form of charts or diagrams and generate HTTP traffic reports.

HTTP Sniffers for Security Analyzing

Network and System Administrators use network sniffer software to monitor and troubleshoot the network traffic. For example, administrators may use our http analyzer to see the HTTP data packets sent by malware programs and identify the security risks or to detect undesirable activities and maintain effective network data flow.

UNIT 4

HACKING WEB SERVICES AND SESSION HIJACKING

WEB APPLICATION VULNERABILITIES

Web application vulnerabilities involve a system flaw or weakness in a web-based application. They have been around for years, largely due to not validating or sanitizing form inputs, misconfigured web servers, and application design flaws, and they can be exploited to compromise the application's security. These vulnerabilities are not the same as other common types of vulnerabilities, such as network or asset. They arise because web applications need to interact with multiple users across multiple networks, and that level of accessibility is easily taken advantage of by hackers.

There are web application security solutions designed specifically for applications, and as such it's important to look beyond traditional vulnerability scanners when it comes to identifying gaps in an organization's application security. To really understand your risks, learn more about some types of web application and cybersecurity attacks, and how web scanners can help increase the safety of your applications.

1. SQL Injection Attacks

Structured Query Language (SQL) is now so commonly used to manage and direct information on applications that hackers have come up with ways to slip their own SQL commands into the database. These commands may change, steal or delete data, and they may also allow the hacker access to the root system. SQL (officially pronounced *ess-cue-el*, but commonly pronounced "sequel") stands for structured query language; it's a programming language used to communicate with databases. Many of the servers that store critical data for websites and services use SQL to manage the data in their databases.

An SQL injection attack specifically targets this kind of server, using malicious code to get the server to divulge information it normally wouldn't. This is especially problematic if the server stores private customer information from the website or web application, such as credit card numbers, usernames and passwords (credentials), or other personally identifiable information, which are tempting and lucrative targets for an attacker.

Successful SQL injection attacks typically occur because a vulnerable application doesn't properly sanitize inputs provided by the user, by not stripping out anything that appears to be SQL code. For example, if an application is vulnerable to an injection attack, it may be possible for an attacker to go to a website's search box and type in code that would instruct the site's SQL server to dump all of its stored usernames and passwords for the site.

2. Cross-Site Scripting (XSS)

In an SQL injection attack, an attacker goes after a vulnerable website to target its stored data, such as user credentials or sensitive financial data. But if the attacker would rather directly target a website's users, they may opt for a cross-site scripting attack. Similar to an SQL injection attack, this attack also involves injecting malicious code into a website or web-based app. However, in this case the malicious code the attacker has injected only runs in the user's browser when they visit the attacked website, and it goes after the visitor directly.

One of the most common ways an attacker can deploy a cross-site scripting attack is by injecting malicious code into an input field that would be automatically run when other visitors view the infected page. For example, they could embed a link to a malicious JavaScript in a comment on a blog.

Cross-site scripting attacks can significantly damage a web company's reputation by placing the users' information at risk without any indication that anything malicious even occurred. Any sensitive information a user sends to the site or the application such as their credentials, credit card information, or other private data can be hijacked via cross-site scripting without the owners realizing there was even a problem in the first place.

3. <u>Cross-Site Request Forgery (CSRF)</u>

A Cross-Site Request Forgery (CSRF) attack is when a victim is forced to perform an unintended action on a web application they are logged into. The web application will have already deemed the victim and their browser trustworthy, and so executes an action intended by the hacker when the victim is tricked into submitting a malicious request to the application. This has been used for everything from harmless pranks on users to illicit money transfers.

One way website owners can help cut down on their chance of attack is to have advanced validation techniques in place for anyone who may visit pages on their site or app, especially when it comes to social media or community sites. This will enable them to identify the user's browser and session to verify their authenticity.

While there are a variety of ways a hacker may infiltrate an application due to web application vulnerabilities, there are also a variety of ways to defend against it. There are web application security testing tools specially designed to monitor even the most public of applications. Using these scanners reduce your chances of being the victim of a hack by showing you exactly where to make the changes needed for more secure applications.

4. <u>Carriage Return and Line Feed (CRLF) Injection</u>

Carriage return is a command that indicates the start of a line of code, normally denoted as \r. Line feed is a command that indicates the end of a line of code, normally denoted as \n. Like many other software, each operating system uses a different combination of carriage return and line feed. When malicious actors engage in CRLF injections, the inserted code changes the way that the web application responds to commands. This can be used to either disclosure sensitive information or execute code.

5. Broken access control

Access controls define how users interact with data and resources including what they can read or edit. A broken access control vulnerability exists when a user has the ability to interact with data in a way that they don't need. For example, if a user should only be able to read payment details but can actually edit them, this is a broken access control. Malicious actors use this vulnerability to gain unauthorized access to systems, networks, and software. They can then escalate the privileges, give the user ID additional access within the ecosystem, to negatively impact data confidentiality, integrity, or availability.

6. Broken authentication

Broken authentication vulnerabilities also focus on user access. However, in this case, malicious actors compromise the information that confirms a user's identity, such as by stealing passwords, keys, or session tokens. The malicious actor gains unauthorized access to the systems, networks, and software because the company failed to adequately set appropriate identity and access management controls.

7.Insecure direct object references (IDOR)

Web application URLs can expose the format/pattern used for directing users to backend storage locations. For example, a URL might indicate the format/pattern for a record identifier in a storage system such as a database or file system.

Alone, the IDOR may be a low-risk issue. However, an IDOR in combination with a failed access control check gives attackers a way to successfully launch an enumeration attack.

8. Insufficient logging and monitoring

Insufficient logging and monitoring vulnerabilities occur when your data event logs fail to capture the necessary information that can prevent an attack. Every user, device, and resource generates an event log that tells your security team what is happening in your systems, networks, and applications.

Since successful attacks often use vulnerability probing during the reconnaissance stage, collecting the right event log data is a way to mitigate risk. Common logging and monitoring weaknesses include:

- Failure to collect logs for auditable events like logins, failed logins, and high-value transactions
- Failure to generate an adequate and clear warning and error logs
- Failure to monitor application and API logs for abnormal activity
- Storing logs locally
- Failure to effectively set alerting thresholds and response escalation processes

- Lack of alert triggers during penetration tests and dynamic application security testing (DAST) scans
- Lack of real-time or near real-time application detection, escalation, and alerting functions.

9. Insufficient session expiration

Session timeout is when an application automatically logs a user out after being idle for a specified amount of time. When an application is idle and open, attackers look to steal the credentials associated with the account.

Some examples of insufficient session expiration weaknesses include:

- Lack of session timeout
- Session timeouts that are longer than necessary
- Inability to trace session creation/destruction to analyze trends

10. Lightweight Directory Access Protocol (LDAP) injection

LDAP is a protocol that lets applications talk with directory services servers that store user IDs, passwords, and computer accounts. When applications accept user input and execute it, attackers can exploit the LDAP server by sending malicious requests.

Some examples of LDAP coding issues include:

- Excess access privileged assigned to LDAP accounts
- Lack of output regulation
- Inability to perform dynamic checks
- Lack of static source code analysis.

APPLICATION CODING ERROR

1. Invalid inputs

By not validating what content and inputs get uploaded, the website is left vulnerable to injection attacks like **cross-site scripting** (XSS), **SQL injection**, **command injection**, and other such security attacks. Input uploads must be validated from both the server and browser ends. Often, organizations validate inputs only from the browser end because it is easy and fail to validate server end inputs which leads to malicious/malformed data/scripts to run on the website and its databases.

2. Irregular or no website security scans

The importance of regular **website security scanning** cannot be stressed enough. It is only through regular scanning that we can find vulnerabilities and gaps that exist, and accordingly, fix them. Organizations often make the cardinal error of not scanning their websites every day and after major changes to the business policies, systems, etc.

3. Authentication and permissions

- Weak root passwords from the admin or server end like admin, 1234, or other commonly used words. These can be easily cracked using password-cracking programs and if the password is cracked, the website will be compromised.
- Not enforcing a strong password policy and multi-factor authentication for the website users. When the website allows its users to continue with default passwords, allows weak passwords without password expiry, and relies uni-dimensionally on passwords for security, the organization is making itself vulnerable to breaches and attacks.
- Giving administrator permissions and privileges mindlessly to end-users and external entities make the website vulnerable.
- Changing folder and file permission structures based on poor advice from the internet to fix permission errors but opening the website up for anyone to change its structure, modify codes, and run malicious programs.

4. Unconsolidated security measures

It often happens that organizations and web developers are not thinking of website security in a holistic manner and therefore, adopting unconsolidated security measures. For instance, they may employ a web security scanner but not a **Web Application Firewall** (WAF). So, the vulnerabilities and gaps are effectively identified by the scanner, but the website is left in the vulnerable condition till the vulnerabilities are fixed (which takes over 100 days even for critical vulnerabilities) or the developers are focusing on patching the website instead of fixing the vulnerabilities.

5. Homegrown security methods and algorithms

Based on the flawed assumption that homegrown/self-developed algorithms and methods are better and that they are safer as attackers are unfamiliar, developers employ these homegrown and ‘authentic’ security measures. This just increases the probability of vulnerabilities and gaps that can be easily detected by attackers and the bots they employ. It is always better to use well-tested methods and algorithms.

6. Outdated software, Components with known vulnerabilities & unnecessary/unwanted components

Updates contain critical patches and by not updating the software regularly, we are just sending out invitations to attackers (who continuously snoop around for loopholes and security lapses) to orchestrate breaches. Old and wanted files, applications, databases, etc. not being cleaned out from the website create portals for attackers.

Developers using components that are known to have vulnerabilities such as unpatched third-party software, outdated plug-ins, open-source components, uninspected and copy-pasted codes, etc. too make the website insecure, weak and susceptible to attacks.

7. Not tested on a regular basis

While website scanning needs to be done every day and after major changes, it is not sufficient. It is essential to test every bit of code, software, updates, and a component that goes on the website. Also, quarterly penetration testing and security audits by certified security experts is a must. This will ensure that your website is secure and that your users are well-protected.

8. Unencrypted sensitive data

One of the most dangerous mistakes committed by organizations is not encrypting sensitive data such as personal information, credit card, and baking details, passwords, etc. at all times (transit, rest and storage) By not encrypting all the sensitive data and having it plain text format, we are simply increasing the risk of exposure.

9. Missing function level access control

When sensitive request handlers have insufficient or non-existent authentication check, the vulnerability that results is known as a missing function level access control. Example- an unauthorized entity can access a URL that contains sensitive information or hidden functionality, etc. because there is no authentication check put in place. The impact of this vulnerability varies from access to unimportant information to complete website takeover by attackers.

10. Lax attitude towards website security

This is the most dangerous of all website security mistakes. The top management must have a proactive attitude towards website security, investing wisely for the right purposes, developing a sound cybersecurity strategy, and honing a culture of proactivity and preparedness within the organization as well. Silos must be broken, and critical information must be seamlessly shared across departments.

Employing an intelligent, comprehensive, and managed website security solution like **AppTrana** is a definite way forward. AppTrana takes a 360-degree view of web application security and provides round-the-clock, end-to-end website security with zero assured false positives through everyday scanning of the website, blocking malicious/bad requests by patching the application-layer vulnerabilities until fixed, continuously monitoring for **DDoS attacks**, analyzing attack patterns and so on. It combines the power of technology and automation with the irreplaceable human expertise of certified security professionals to secure your website while you concentrate on your core business activities.

SQL INJECTION

In computing, **SQL injection** is a code injection technique used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker).[1][2] SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.

This form of injection relies on the fact that SQL statements consist of both data used by the SQL statement and commands that control how the SQL statement is executed. For example, in the SQL statement

select * **from** person **where** name = 'susan' **and** age = 2 the string 'susan' is data and the fragment **and** age = 2 is an example of a command (the value 2 is also data in this example).

SQL injection occurs when specially crafted user input is processed by the receiving program in a way that allows the input to exit a data context and enter a command context. This allows the attacker to alter the structure of the SQL statement which is executed.

As a simple example, imagine that the data 'susan' in the above statement was provided by user input. The user entered the string 'susan' (without the apostrophes) in a web form text entry field, and the program used string concatenation statements to form the above SQL statement from the three fragments **select** * **from** person **where** name=', the user input of 'susan', and ' **and** age = 2.

Now imagine that instead of entering 'susan' the attacker entered ' **or** 1=1; **--**.

The program will use the same string concatenation approach with the 3 fragments of

select * **from** person **where** name=', the user input of ' **or** 1=1; **--**, and ' **and** age = 2

and construct the statement

select * **from** person **where** name='' **or** 1=1; **--** *and age = 2*.

Many databases will ignore the text after the '**--**' string as this denotes a comment. The structure of the SQL command is now

select * **from** person **where** name='' **or** 1=1;

and this will select all person rows rather than just those named 'susan' whose age is 2. The attacker has managed to craft a data string which exits the data context and entered a command context.

A more complex example is now presented.

Imagine a program creates a SQL statement using the following string assignment command :

var statement = "SELECT * FROM users WHERE name = '" + userName + "'";

This SQL code is designed to pull up the records of the specified username from its table of users. However, if the "userName" variable is crafted in a specific way by a malicious user, the SQL statement may do more than the code author intended. For example, setting the "userName" variable as:

```
' OR '1'='1
```

or using comments to even block the rest of the query (there are three types of SQL comments[14]). All three lines have a space at the end:

```
' OR '1'='1' --
' OR '1'='1' {
' OR '1'='1' /*
```

renders one of the following SQL statements by the parent language:

```
SELECT * FROM users WHERE name = '' OR '1'='1';
SELECT * FROM users WHERE name = '' OR '1'='1' -- ';
```

If this code were to be used in authentication procedure then this example could be used to force the selection of every data field (*) from *all* users rather than from one specific user name as the coder intended, because the evaluation of '1'='1' is always true.

The following value of "userName" in the statement below would cause the deletion of the "users" table as well as the selection of all data from the "userinfo" table (in essence revealing the information of every user), using an API that allows multiple statements:

```
a';DROP TABLE users; SELECT * FROM userinfo WHERE 't' = 't
```

This input renders the final SQL statement as follows and specified:

```
SELECT * FROM users WHERE name = 'a';DROP TABLE users; SELECT * FROM userinfo
WHERE 't' = 't';
```

While most SQL server implementations allow multiple statements to be executed with one call in this way, some SQL APIs such as PHP's `mysql_query()` function do not allow this for security reasons. This prevents attackers from injecting entirely separate queries, but doesn't stop them from modifying queries.

Types of SQL Injection (SQLi)

SQL Injection can be used in a range of ways to cause serious problems. By levering SQL Injection, an attacker could bypass authentication, access, modify and delete data within a database. In some cases, SQL Injection can even be used to execute commands on the operating system, potentially allowing an attacker to escalate to more damaging attacks inside of a network that sits behind a firewall.

SQL Injection can be classified into three major categories – *In-band SQLi*, *Inferential SQLi* and *Out-of-band SQLi*.

In-band SQLi (Classic SQLi)

In-band SQL Injection is the most common and easy-to-exploit of SQL Injection attacks. In-band SQL Injection occurs when an attacker is able to use the same communication channel to both launch the attack and gather results.

The two most common types of in-band SQL Injection are *Error-based SQLi* and *Union-based SQLi*.

Error-based SQLi

Error-based SQLi is an in-band SQL Injection technique that relies on error messages thrown by the database server to obtain information about the structure of the database. In some cases, error-based SQL injection alone is enough for an attacker to enumerate an entire database. While errors are very useful during the development phase of a web application, they should be disabled on a live site, or logged to a file with restricted access instead.

Union-based SQLi

Union-based SQLi is an in-band SQL injection technique that leverages the UNION SQL operator to combine the results of two or more SELECT statements into a single result which is then returned as part of the HTTP response.

Inferential SQLi (Blind SQLi)

Inferential SQL Injection, unlike in-band SQLi, may take longer for an attacker to exploit, however, it is just as dangerous as any other form of SQL Injection. In an inferential SQLi attack, no data is actually transferred via the web application and the attacker would not be able to see the result of an attack in-band (which is why such attacks are commonly referred to as "blind SQL Injection attacks"). Instead, an attacker is able to reconstruct the database structure by sending payloads, observing the web application's response and the resulting behavior of the database server.

The two types of inferential SQL Injection are *Blind-boolean-based SQLi* and *Blind-time-based SQLi*.

Boolean-based (content-based) Blind SQLi

Boolean-based SQL Injection is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the application to return a different result depending on whether the query returns a TRUE or FALSE result.

Depending on the result, the content within the HTTP response will change, or remain the same. This allows an attacker to infer if the payload used returned true or false, even though no data from the database is returned. This attack is typically slow (especially on large databases) since an attacker would need to enumerate a database, character by character.

Time-based Blind SQLi

Time-based SQL Injection is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the database to wait for a specified amount of time (in seconds) before responding. The response time will indicate to the attacker whether the result of the query is TRUE or FALSE.

Depending on the result, an HTTP response will be returned with a delay, or returned immediately. This allows an attacker to infer if the payload used returned true or false, even though no data from the database is returned. This attack is typically slow (especially on large databases) since an attacker would need to enumerate a database character by character.

Out-of-band SQLi

Out-of-band SQL Injection is not very common, mostly because it depends on features being enabled on the database server being used by the web application. Out-of-band SQL Injection occurs when an attacker is unable to use the same channel to launch the attack and gather results.

Out-of-band techniques, offer an attacker an alternative to inferential time-based techniques, especially if the server responses are not very stable (making an inferential time-based attack unreliable).

Out-of-band SQLi techniques would rely on the database server's ability to make DNS or HTTP requests to deliver data to an attacker. Such is the case with Microsoft SQL Server's xp_dirtree command, which can be used to make DNS requests to a server an attacker controls; as well as Oracle Database's UTL_HTTP package, which can be used to send HTTP requests from SQL and PL/SQL to a server an attacker controls.

CROSS-SITE SCRIPTING

Cross-site scripting (also known as XSS) is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application. It allows an attacker to circumvent the same origin policy, which is designed to segregate different websites from each other. Cross-site scripting vulnerabilities normally allow an attacker to masquerade as a victim user, to carry out any actions that the user is able to perform, and to access any of the user's data. If the victim user has privileged access within the application, then the attacker might be able to gain full control over all of the application's functionality and data.

How does XSS work?

Cross-site scripting works by manipulating a vulnerable web site so that it returns malicious JavaScript to users. When the malicious code executes inside a victim's browser, the attacker can fully compromise their interaction with the application.

XSS proof of concept

You can confirm most kinds of XSS vulnerability by injecting a payload that causes your own browser to execute some arbitrary JavaScript. It's long been common practice to use the `alert()` function for this purpose because it's short, harmless, and pretty hard to miss when it's successfully called. In fact, you solve the majority of our XSS labs by invoking `alert()` in a simulated victim's browser.

Unfortunately, there's a slight hitch if you use Chrome. From version 92 onward (July 20th, 2021), cross-origin iframes are prevented from calling `alert()`. As these are used to construct some of the more advanced XSS attacks, you'll sometimes need to use an alternative PoC payload. In this scenario, we recommend the `print()` function. As the simulated victim in our labs uses Chrome, we've amended the affected labs so that they can also be solved using `print()`. We've indicated this in the instructions wherever relevant.

What are the types of XSS attacks?

There are three main types of XSS attacks. These are:

- Reflected XSS, where the malicious script comes from the current HTTP request.
- Stored XSS, where the malicious script comes from the website's database.
- DOM-based XSS, where the vulnerability exists in client-side code rather than server-side code.

Reflected cross-site scripting

Reflected XSS is the simplest variety of cross-site scripting. It arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.

Here is a simple example of a reflected XSS vulnerability:

```
https://insecure-website.com/status?message=All+is+well.  <p>Status:  All  is
well.</p>
```

The application doesn't perform any other processing of the data, so an attacker can easily construct an attack like this:

```
https://insecure-
website.com/status?message=<script>/*+Bad+stuff+here...+*/</script>
<p>Status: <script>/* Bad stuff here... */</script></p>
```

If the user visits the URL constructed by the attacker, then the attacker's script executes in the user's browser, in the context of that user's session with the application. At that point, the script can carry out any action, and retrieve any data, to which the user has access.

Stored cross-site scripting

Stored XSS (also known as persistent or second-order XSS) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.

The data in question might be submitted to the application via HTTP requests; for example, comments on a blog post, user nicknames in a chat room, or contact details on a customer order. In other cases, the data might arrive from other untrusted sources; for example, a webmail application displaying messages received over SMTP, a marketing application displaying social media posts, or a network monitoring application displaying packet data from network traffic.

Here is a simple example of a stored XSS vulnerability. A message board application lets users submit messages, which are displayed to other users:

```
<p>Hello, this is my message!</p>
```

The application doesn't perform any other processing of the data, so an attacker can easily send a message that attacks other users:

```
<p><script>/* Bad stuff here... */</script></p>
```

DOM-based cross-site scripting

DOM-based XSS (also known as DOM XSS) arises when an application contains some client-side JavaScript that processes data from an untrusted source in an unsafe way, usually by writing the data back to the DOM.

In the following example, an application uses some JavaScript to read the value from an input field and write that value to an element within the HTML:

```
var search = document.getElementById('search').value; var results =
document.getElementById('results'); results.innerHTML = 'You searched for: '
+ search;
```

If the attacker can control the value of the input field, they can easily construct a malicious value that causes their own script to execute:

```
You searched for: <img src=1 onerror='/* Bad stuff here... */'>
```

In a typical case, the input field would be populated from part of the HTTP request, such as a URL query string parameter, allowing the attacker to deliver an attack using a malicious URL, in the same manner as reflected XSS.

Impact of XSS vulnerabilities

The actual impact of an XSS attack generally depends on the nature of the application, its functionality and data, and the status of the compromised user. For example:

- In a brochureware application, where all users are anonymous and all information is public, the impact will often be minimal.
- In an application holding sensitive data, such as banking transactions, emails, or healthcare records, the impact will usually be serious.
- If the compromised user has elevated privileges within the application, then the impact will generally be critical, allowing the attacker to take full control of the vulnerable application and compromise all users and their data.

How to prevent XSS attacks

Preventing cross-site scripting is trivial in some cases but can be much harder depending on the complexity of the application and the ways it handles user-controllable data.

In general, effectively preventing XSS vulnerabilities is likely to involve a combination of the following measures:

- **Filter input on arrival.** At the point where user input is received, filter as strictly as possible based on what is expected or valid input.
- **Encode data on output.** At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.
- **Use appropriate response headers.** To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the `Content-Type` and `X-Content-Type-Options` headers to ensure that browsers interpret the responses in the way you intend.

- **Content Security Policy.** As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.

CROSS-SITE REQUEST FORGING

What is cross-site request forgery?

Cross-site request forgery (*CSRF*) is a web vulnerability that lets a malicious hacker trick the victim into submitting a request that allows the attacker to perform state-changing actions on behalf of the victim. Cross-site request forgery is also called *XSRF*, *sea surf*, *session riding*, or *one-click attack*.

Severity:	■■□□□	severe in rare circumstances
Prevalence:	■■■■□	discovered often
Scope:	■■■□□	web applications with authentication
Technical impact:		attacker triggers unauthorized actions
Worst-case consequences:		depend on application capabilities
Quick fix:		use anti-CSRF tokens

How does cross-site request forgery work?

Most web applications require authentication and some authenticated users are able to perform very sensitive actions. Authentication in web applications is often performed based on user sessions. After you authenticate, your browser stores a session cookie on your computer and sends it with every request you make to that web application. Less commonly, applications can also use NTLM or Basic Auth for authentication instead of session cookies, or even recognize users based on their IP address.

When you are using an application, many HTTP requests sent from your browser to the application are the result of your explicit actions, for example, when you type an URL in the address bar or click a link. However, other HTTP requests are sent by your browser implicitly as it processes code included on the current web page. For example, if the page includes an image, the image will be fetched by a separate HTTP request.

Such implicit requests can also be directed to domains that have nothing to do with the location of the page you're viewing. For example, an image displayed on *testinvicti.com* may in reality come from *example.com*. The crucial thing in such cases is that requests to both locations come from the same browser, so your current authentication method (whether it's a session cookie or another method) applies to both locations. So if your browser opens *testinvicti.com* and fetches an image from *example.com*, thus creating a user session at *example.com*, the *example.com* web application will treat you as an authenticated user (even though you originally opened *testinvicti.com*, not *example.com*).

Combined, these two behaviors can be exploited to perform cross-site request forgery attacks in the following way:

1. The victim is authenticated in the target web application (such as *example.com*).
2. The attacker uses social engineering to trick the victim into visiting a malicious website (for example, *testinvicti.com*).
3. The malicious web page includes code (such as an image tag) that causes the victim's browser to send an implicit request to the target (such as *example.com*).
4. The malicious request causes the target to perform actions that were not intended by the user. The consequences will vary depending on the application.

Note that CSRF used to have its own separate category in the OWASP Top 10 (for example, A5:2013). However, with the development of more efficient AppSec and therefore the reduced impact of such vulnerabilities, since 2017 OWASP decided to merge CSRF into another, more generic category.

Types of cross-site request forgery vulnerabilities

CSRF vulnerabilities can be based on GET or POST requests.

In the case of CSRF based on GET requests, the attacker can simply use an image tag (or any other tag that allows for cross-site requests) on a malicious page:

```
<img
src="http://example.com/bank.php/?action=transfer&target=attacker_account">
```

When the user visits the page with the above image tag (for example, after clicking a malicious link), the user's browser tries to open the image but instead makes a GET request to the targeted site, thus performing a malicious action while logged into the user's account. Assuming that the user is authenticated on *example.com*, the web application will be unable to differentiate between a legitimate user request and a malicious request, since they are both sent from the same browser.

In the case of CSRF based on POST requests, the attacker needs to work a bit harder. The simplest way to perform such an attack is to force the user's browser to automatically submit a form by using JavaScript:

```
<body onload="document.csrf.submit()">
<form action="http://example.com/bank.php" method="POST" name="csrf">
    <input type="hidden" name="action" value="transfer">
    <input type="hidden" name="target" value="attacker_account">
</form>
```

The `onload` argument of the `<body>` tag will cause the browser to submit the form as soon as the user opens the malicious page.

Example of a cross-site request forgery attack

The developer of a financial business application creates a function that allows users to set the email address they want to use for daily financial reports from the application. To set or change the email address, an authenticated user must fill out an HTML form on the *http://example.com/set_email.php* page:

```
<form action="/set_email.php" method="post" id="set_email">
    <label for="email">Enter the email address to receive reports:</label>
    <input type="email" id="email" name="email">
    <button type="submit" form="submit" value="submit">Set email</button>
</form>
```

The attacker creates a malicious page *http://example.attacker/exploit.html* with the following content:

```
<body onload=document.email.submit()>
    <form       action="http://example.com/set_email.php"       method="post"
name="email">
        <input type="hidden" id="email" value="attacker@example.attacker">
    </form>
</body>
```

Then, the attacker creates a phishing email and sends it to a user of the financial application, tricking the user into visiting *http://example.attacker/set_email.html*. Assuming that the user is already logged in to the application at *example.com*, the application will receive the forged request and change the reporting email to *attacker@example.attacker*. As a result, the attacker will receive daily sensitive reports about the company's financial operations.

Potential consequences of a cross-site request forgery attack

Cross-site request forgery vulnerabilities are considered medium severity for several reasons:

- In this type of attack, the attacker never receives the HTTP response and therefore cannot use this technique to directly read/access sensitive information. They don't even have access to the session cookie value that is sent with the malicious request.
- The attack is limited by the functionality of the web application, or, more precisely, what the application allows the current user to do using a state-changing request. For example, if you have a ticketing system and the current user can only create and resolve issues, the most that an attacker can achieve with CSRF is clear the ticket queue. They won't, for example, be able to get the administrator's credentials.
- This type of attack is most effective when aimed at a specific person or a small group of people with high privileges. Unlike with cross-site scripting (XSS), it often makes no sense to send a malicious CSRF payload to a large number of random victims. CSRF is

usually carefully prepared to take advantage of a specific user in the business, such as the CEO, the administrator, or a financial department employee.

Examples of known cross-site request forgery vulnerabilities

Due to the nature of CSRF, there are no known major breaches caused by successful CSRF attacks. However, in the past, several popular web applications were found to be vulnerable to cross-site request forgery and could have been used in targeted attacks:

- Netflix: In 2006, when Netflix was still a DVD-rental service, it was found to have a CSRF vulnerability that could let an attacker change the credentials and completely overtake an account.
- ING: In 2008, researchers discovered CSRF vulnerabilities in *ingdirect.com* that could allow an attacker to open bank accounts on behalf of the victim and transfer funds from the victim's account.
- WordPress: In 2020, researchers found that 25 popular WordPress plugins had CSRF vulnerabilities.

These are just a few examples out of many and while we are not aware of any dire consequences of these vulnerabilities, it is possible that they were used for individually targeted attacks that simply never made it to the media.

How to detect cross-site request forgery vulnerabilities?

The best way to detect CSRF vulnerabilities varies depending on whether they are already known or unknown.

- If you only use commercial or open-source web applications and do not develop web applications of your own, it may be enough to identify the exact version of the application you are using. If the identified version is susceptible to CSRF, you can assume that your website is vulnerable. You can identify the version manually or use a suitable security tool, such as a software composition analysis (SCA) solution.
- If you develop your own web applications or want the ability to potentially find previously unknown CSRF vulnerabilities (zero-days) in known applications, you must be able to successfully exploit the CSRF vulnerability to be certain that it exists. This requires either performing manual penetration testing with the help of security researchers or using a security testing tool (scanner) that can use automation to exploit web vulnerabilities. Examples of such tools are Invicti and Acunetix by Invicti. We recommend using this method even for known vulnerabilities.

How to prevent cross-site request forgery vulnerabilities in web applications?

The primary method of protecting against CSRF attacks is to create a way for the web application to differentiate between legitimate requests (made on behalf of that application) and potentially malicious ones (sent by the application under outside influence). The following two techniques are the most effective and widely used.

Anti-CSRF tokens

This protection technique is based on sending a special token with each legitimate request and always validating that token when receiving requests. This *anti-CSRF token*, sometimes called a *synchronizer token*, is generated on the server side and attackers have no way of knowing its correct value – it is known only to the web application and the browser. Requests sent as CSRF attack attempts won't have a valid token, which allows the application to ignore them as invalid, log them as attack attempts, or even raise an alarm.

Once you have generated an anti-CSRF token, you can include it in a hidden form field or automatically add it in a special header for every request. Note that anti-CSRF tokens should be used not just for every form in the authenticated zone of the web application but also for unauthenticated login forms, APIs, and AJAX requests (`XMLHttpRequest`).

There are many libraries available to generate and verify anti-CSRF tokens safely using cryptographic techniques, for example, Paragonie anti-CSRF for PHP and GDS anti-CSRF for Java. We recommend using such libraries instead of trying to create your own code, which would be more prone to errors and harder to maintain.

Also note that while many modern development frameworks already have synchronizer tokens built in, their CSRF protection is often limited to HTTP methods that are designed for state-changing requests. This means that GET requests are typically not covered. Therefore, if a developer creates state-changing functions that take their input from GET requests, which is very bad programming practice, these requests will not be covered by built-in CSRF protection.

SameSite cookies

Another very effective way to differentiate legitimate requests from potentially harmful ones is by looking at the origin of the request. You can safely assume that if the request comes from the same domain/site, it's most likely legitimate. If it comes from an external domain, it could be harmful. To take advantage of this method, you can use a specific cookie security flag.

Modern browsers support the `SameSite` cookie attribute, which you can use when setting your session cookies. This can have one of three settings:

- Lax: The browser does not send cookies for cross-site subrequests, for example, to load images or frames into a third-party site, but does send cookies when a user follows a link.
- Strict: The browser sends cookies only in a first-party context and does not send them at all with requests initiated by third-party websites.
- None: The browser sends cookies in all contexts, but you must also set the Secure attribute or the browser will block the cookie.

While most modern browsers set the `SameSite` attribute to `Lax` by default for all cookies, we recommend that you manually set it in your web application anyway (to `Lax` or `Strict`, depending on whether you need cross-site subrequests or not). This is in case you get users with older browser versions that set `SameSite` to `None` by default.

Unfortunately, if this is the only method you use to protect your users against CSRF, a small number of users with legacy browsers such as Internet Explorer that don't support SameSite cookies at all will remain vulnerable to CSRF attacks.

Other protection techniques

While synchronizer tokens and SameSite cookies are considered the best methods of CSRF protection, there are also other ways to differentiate between legitimate and potentially malicious requests. Some developers will, for example, use the *referer* header to spot this difference. Others attempt to implement protection based on mechanisms such as the same-origin policy, which is ineffective against CSRF.

While methods such as referrer detection can be effective, they are not as foolproof as anti-CSRF tokens, so we do not recommend using any methods other than synchronizer tokens and SameSite cookies, preferably together.

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. The details of sequence-number prediction will be discussed later in this chapter in the sequence prediction section. 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.

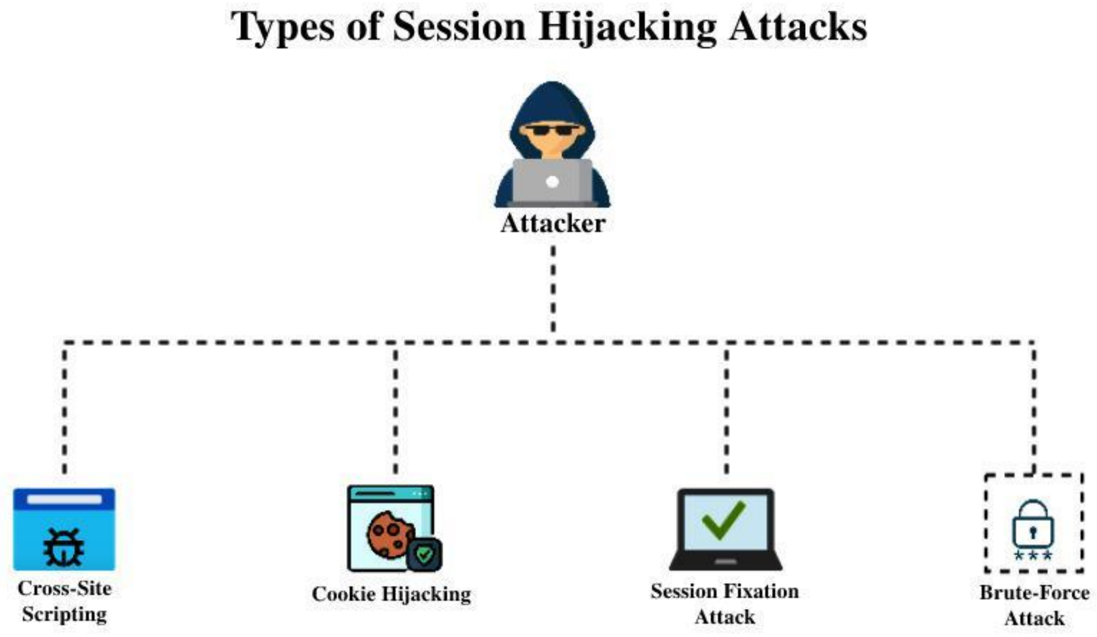
Desynchronizing the Connection: The hacker sends the valid user's system a TCP reset

(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.

Session Hijacking is a Hacking Technique. In this, the hackers (the one who perform hacking) gain the access of a target's computer or online account and exploit the whole web session control mechanism. This is done by taking over an active TCP/IP communication session by performing illegal actions on a protected network. Normally, the web sessions are managed by the session token. The Session Hijacker has access over everything which the actual user has. **For Example,** shopping in an online store or paying your electricity bills, the session hijackers attack over web browsers or web application sessions.



Types of Session Hijacking:

Session Hijacking is of Three types:

1. **Active Session Hijacking** : An Active Session Hijacking occurs when the attacker takes control over the active session. The actual user of the network becomes in offline mode, and the attacker acts as the authorized user. They can also take control over the communication between the client and the server. To cause an interrupt in the communication between client and server, the attackers send massive traffic to attack a valid session and cause a denial of service attack(DoS).
2. **Passive Session Hijacking** : In Passive Session Hijacking, instead of controlling the overall session of a network of targeted user, the attacker monitors the communication between a user and a server. The main motive of the hacker is to listen to all the data and record it for the future use. Basically, it steals the exchanged information and use for irrelevant activity. This is also a kind of man-in-middle attack (as the attacker is in between the client and the server exchanging information.
3. **Hybrid Hijacking** : The combination of Active Session Hijacking and Passive Session Hijacking is referred to as Hybrid Hijacking. In this the attackers monitors the communication channel (the network traffic), whenever they find the issue, they take over the control on the web session and fulfill their malicious tasks.

Methods of Session Hijacking

To perform these all kinds of Session Hijacking attacks, the attackers use various methods. They have the choice to use a single method or more than one method simultaneously to perform Session Hijacking. Those methods are:

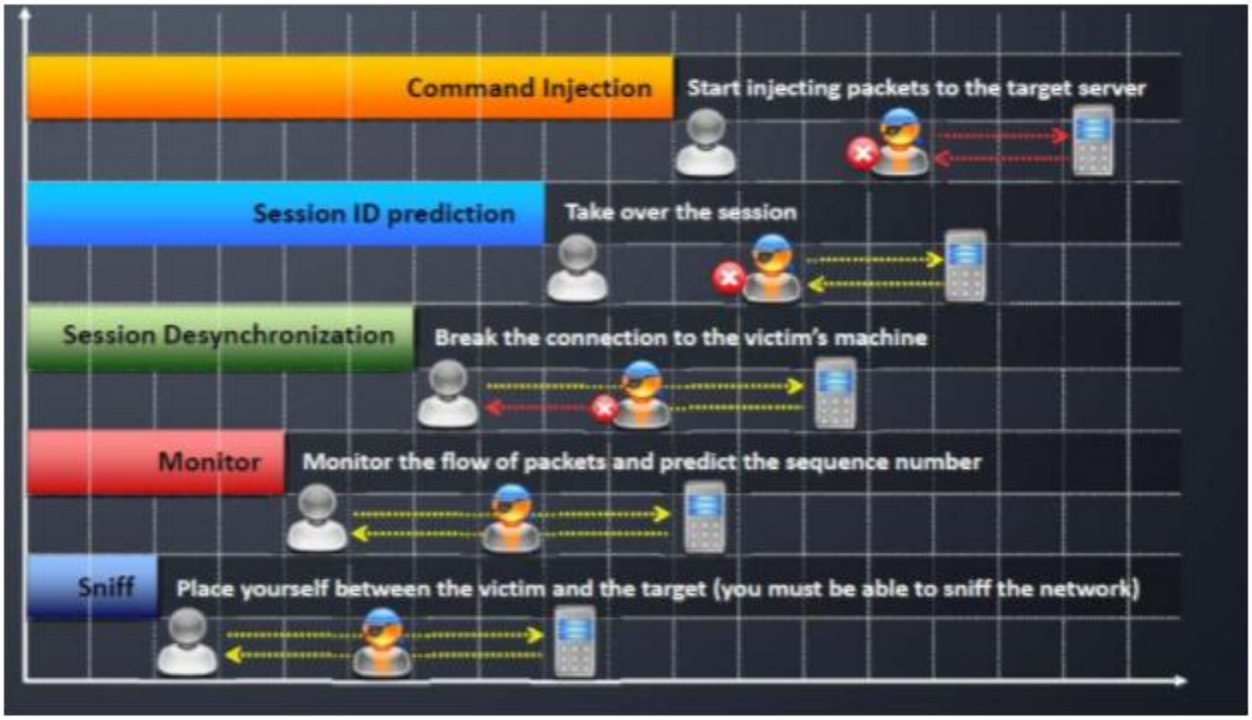
1. Brute-forcing the Session ID
2. Cross-Site Scripting (XSS) or Misdirected Trust
3. Man-in-the-browser
4. Malware infections
5. Session Fixation
6. Session side-jacking

These all Session Hijacking methods can be elaborated as:

1. **Brute-forcing the Session ID** : As the name suggests, the attack user uses guessing and trial method to find Session ID depending on its length. This is due to lack of security and shorter length. The introduction of a strong and long session key made this method increase in a slow rate.
2. **Cross-Site Scripting (XSS) or Misdirected Trust** : In Cross-Site-Scripting, the attacker tries to find out the flaws and the weak point in the web server and injects its code into that. This activity of the attacker will help the attacker to find out the Session ID.
3. **Man-in-the-browser** : Man-in-the-browser uses a Trojan Horse (program that uses malicious code) to perform its required action. The attacker puts themselves in the communication channel of a server and a client. The main purpose of performing this attacks by the attacker is to cause financial fraud.
4. **Malware infections** : In Malware Infections, attacker can deceive the user to open a link that is a malware or Trojans program which will install the malicious software in the device. These are programmed to steal the browser cookies without the user's knowledge.
5. **Session Fixation** : Attackers create a duplicate or another disguised session in Session Fixation. It simply motivates or trick the user into authenticating the vulnerable server. This can be done by sending an email to the user, which on clicking directs to the attacker session.
6. **Session side-jacking** : In Session side-jacking, the attackers tries to get access over a session using the network traffic. This becomes easy when the user is using an insecure Wi-Fi. The reading of

network traffic and stealing of session cookie is done by packet sniffing. Packet Sniffing is a technique by which the data flowing across a network is observed.

SESSION HIJACKING PHASES


Command Injection
Start injecting packets to the target server
Session ID prediction
Take over the session
Session Desynchronization
Break the connection to the victim's machine
Monitor
Monitor the flow of packets and predict the sequence number
Sniff
Place yourself between the victim and the target (you must be able to sniff the network)

The first step in the session hijack attack is locating a target user. Attackers look for two things prior to their attack- first, they look for networks that have a high level of utilization; high volume networks help attackers to remain anonymous and they also provide a healthy supply of users to choose from, which also helps the attack. Secondly, users who use insecure network protocols such as Telnet, rlogin (remote login), and FTP (file transfer protocol) are easy targets due to their inherently insecure design. Packet sniffing software can be used to sniff network traffic for the purpose of locating vulnerable protocols like FTP, Telnet, and rlogin. Port scanning software can also be used to identify servers that have FTP, Telnet, or rlogin ports open.

1. Sniffing into Active Session:

The attacker then finds an active session between the target and another machine and places himself between them. Using a sniffer like Wireshark, he captures the traffic and tries to gather information about the session.

2. Monitor:

He then monitors the traffic for vulnerable protocols like HTTP, telnet, rlogin, etc., and tries to find any valid authentication packets passing through.

3. Session Id Retrieval:

The attacker tries to predict the session id using available information. Now that a target has been chosen, the next step in the session hijacking process is sequence number prediction. Sequence number prediction is a critical step because failing to predict the correct sequence number will result in the server sending reset packets and terminating the connection attempt. If the attacker guesses the sequence numbers wrong repeatedly, the likelihood of detecting the attack increases.

4. Stealing:

In application-level hijacking, active attacks are pursued to steal the session Id. Man in the middle attack, cross-site scripting, sniffing are used to steal the session id.

Brute Forcing: This is a time-consuming process.

While sequencing number guessing can be done manually by skilled attackers, software tools are available to automate the process.

5. Take One of the Parties Offline:

Once a session is chosen and sequence numbers predicted, one of the targets has to be silenced. This is generally done with a denial of service attack. The attacker must ensure that the client computer remains offline for the duration of the attack, or the client computer will begin transmitting data on the network causing the workstation and the server to repeatedly attempt to synchronize their connections; resulting in a condition known as an ACK storm.

6. Take over the Session and Maintain the Connection:

The final phase of the session hijack attack entails taking over the communication session between the workstation and server. The attacker will spoof their client IP address, to avoid detection, and include a sequence number that was predicted earlier. If the server accepts this information, the attacker has successfully attacked the communication session.

Session Hijacking Levels

Session Hijacking can be done at two levels:

1. Network Level
2. Application Level

Network Level hijacking includes TCP and UDP sessions.

Application Level hijacking occurs with HTTP Sessions.

Application Level Hijacking:

Here the valid session token is stolen or predicted to take over the session. Various attacks involved here are-

Man in the middle attack:

By using automated tools/spoofing methods the attacker splits the connection between the targets into two. One connection between the client and attacker and another one between attacker and server. Since the attacker becomes the man in the middle, all the traffic goes through him, hence he can capture the session Id.

Cross-site scripting:

Client-side vulnerabilities like XSS attacks allow an attacker to craft a malicious script to get the session Id from the application.

Using Proxy:

By setting up a proxy and causing the traffic to flow through the proxy, one can capture the session Id details.

Man-in the–Browser:

By installing a Trojan in the victim’s browser will notify the attacker the session Id.

Session Replay:

Capturing the authentication packets by sniffing the traffic; replaying those packets after a time interval may cause the attacker to successfully login to the session of the authorized user.

Session Hijacking Tools

Juggernaut is a network sniffer that can be used to hijack TCP sessions. It runs on Linux operating systems and can be used to watch for all network traffic, or it can be given a keyword such as a password to look for. The program shows all active network connections, and the attacker can then choose a session to hijack.

Hunt is a program that can be used to sniff and hijack active sessions on a network. Hunt performs connection management, Address Resolution Protocol (ARP) spoofing, resetting of connections, monitoring of connections, Media Access Control (MAC) address discovery, and sniffing of TCP traffic.

TTY Watcher is a session-hijacking utility that allows the hijacker to return the stolen session to the valid user as though it was never hijacked. TTY Watcher is only for Sun Solaris systems.

IP Watcher is a session-hijacking tool that lets an attacker monitor connections and take over a session. This program can monitor all connections on a network, allowing the attacker to watch an exact copy of a session in real time.

T-Sight is a session-monitoring and -hijacking tool for Windows that can assist when an attempt at a network break-in or compromise occurs. With T-Sight, a system administrator can monitor all network connections in real time and observe any suspicious activity that takes place. T-Sight can also hijack any TCP session on the network. For security reasons, En Garde Systems licenses this software only to predetermined IP addresses.

The **Remote TCP Session** Reset Utility displays current TCP session and connection information such as IP addresses and port numbers. The utility is primarily used to reset TCP sessions.

UNIT V HACKING WIRELESS NETWORKS

INTRODUCTION TO 802.11

The architecture of the IEEE 802.11 WLAN is designed to support a network where most decision making is distributed to mobile stations. This type of architecture has several advantages. It is tolerant of faults in all of the WLAN equipment and eliminates possible bottlenecks a centralized architecture would introduce. The architecture is flexible and can easily support both small, transient networks and large, semipermanent or permanent networks. In addition, the architecture and protocols offer significant power saving and prolong the battery life of mobile equipment without losing network connectivity

Two network architectures are defined in the IEEE 802.11 standard:

- **Infrastructure network:** An infrastructure network is the network architecture for providing communication between wireless clients and wired network resources. The transition of data from the wireless to wired medium occurs via an AP. An AP and its associated wireless clients define the coverage area. Together all the devices form a basic service set (refer figure 1).
- **Point-to-point (ad-hoc) network:** An ad-hoc network is the architecture that is used to support mutual communication between wireless clients. Typically, an ad-hoc network is created spontaneously and does not support access to wired networks. An ad-hoc network does not require an AP.

The components of an IEEE 802.11 architecture are as follows −

- **Stations (STA)** − Stations comprises of all devices and equipment that are connected to the wireless LAN. A station can be of two types−
  - Wireless Access Point (WAP) − WAPs or simply access points (AP) are generally wireless routers that form the base stations or access.
  - Client. Clients are workstations, computers, laptops, printers, smartphones, etc.
- Each station has a wireless network interface controller.
- **Basic Service Set (BSS)** − A basic service set is a group of stations communicating at the physical layer level. BSS can be of two categories depending upon the mode of operation−
  - Infrastructure BSS − Here, the devices communicate with other devices through access points.
  - Independent BSS − Here, the devices communicate in a peer-to-peer basis in an ad hoc manner.
- **Extended Service Set (ESS)** − It is a set of all connected BSS.
- **Distribution System (DS)** − It connects access points in ESS.

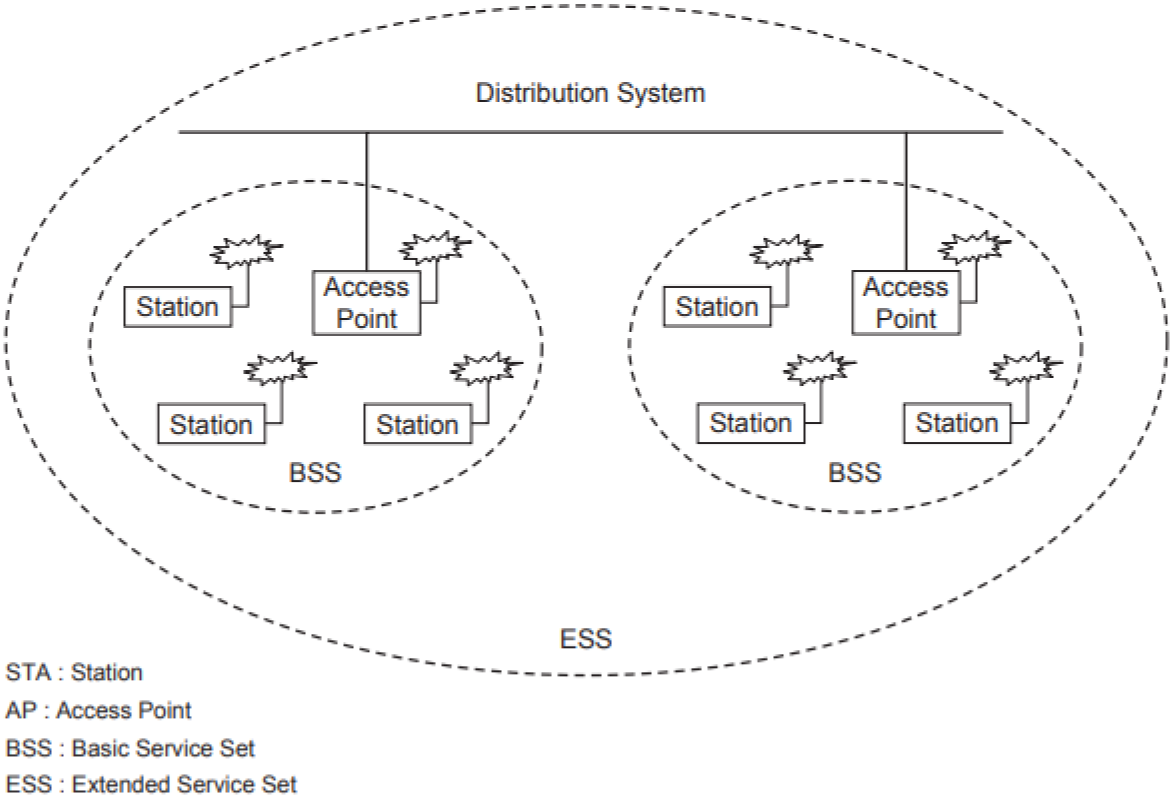


Fig1: BSS and ESS configuration of IEEE 802.11 WLAN

Basic service set: The basic service set configuration relies on an AP that acts as the logical server for a single WLAN cell or channel. Communications between station 1 and station 4 actually flow from station 1 to AP1 and then from AP1 to AP2 and then from AP2 to AP4 and finally AP4 to station 4 (refer to Figure 2). An AP performs a bridging function and connects multiple WLAN cells or channels, and connects WLAN cells to a wired enterprise LAN.

Extended service set: The ESS configuration consists of multiple basic service set cells that can be linked by either wired or wireless backbones called a distributed system. IEEE 802.11 supports ESS configurations in which multiple cells use the same channel to boost aggregate through put to network. The equipment outside of the ESS, the ESS and all of its mobile stations appear to be a single MAC layer network where all stations are physically stationary. Thus, the ESS hides the mobility of the mobile stations from everything outside the ESS

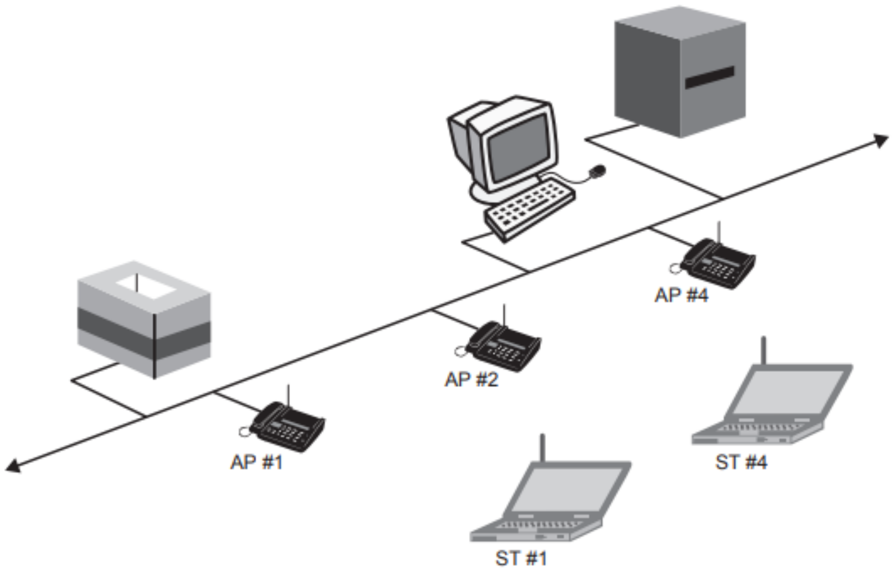
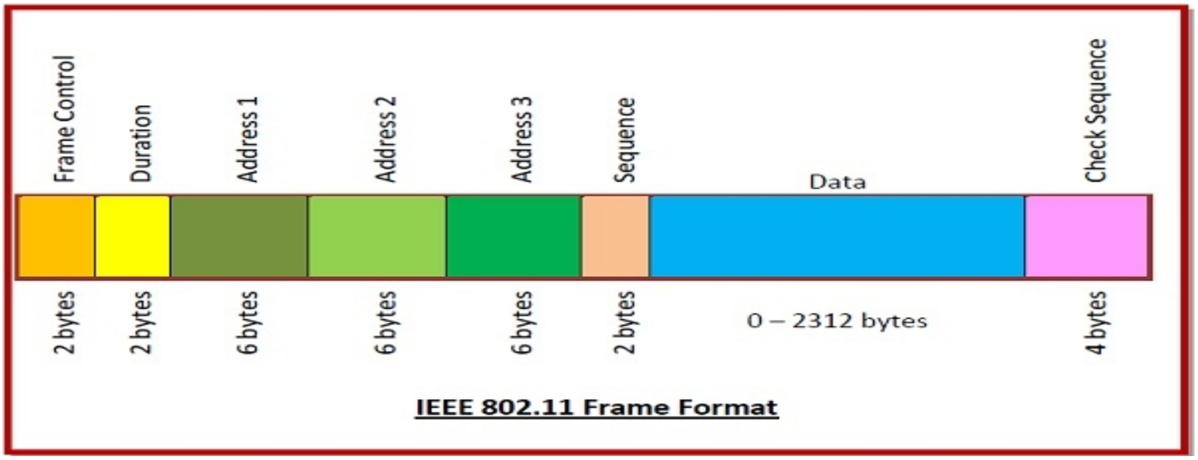


Fig.2 Access point-based topology

Frame Format of IEEE 802.11

The main fields of a frame of wireless LANs as laid down by IEEE 802.11 are −

- **Frame Control** − It is a 2 bytes starting field composed of 11 subfields. It contains control information of the frame.
- **Duration** − It is a 2-byte field that specifies the time period for which the frame and its acknowledgment occupy the channel.
- **Address fields** − There are three 6-byte address fields containing addresses of source, immediate destination, and final endpoint respectively.
- **Sequence** − It a 2 bytes field that stores the frame numbers.
- **Data** − This is a variable-sized field that carries the data from the upper layers. The maximum size of the data field is 2312 bytes.
- **Check Sequence** − It is a 4-byte field containing error detection information.



Frame Control	Duration	Address 1	Address 2	Address 3	Sequence	Data	Check Sequence
2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	0 – 2312 bytes	4 bytes

IEEE 802.11 Frame Format

Wired Equivalent Privacy (WEP)

Since wireless networks transmit data through radio waves, data can be easily intercepted unless security measures are in place. Introduced in 1997, Wired Equivalent Privacy (WEP) was the first attempt at wireless protection. The aim was to add security to wireless networks by encrypting data. If wireless data were intercepted, it would be unrecognizable to the interceptors since it had been encrypted. However, systems that are authorized on the network would be able to recognize and decrypt the data. This is because devices on the network make use of the same encryption algorithm.

WEP encrypts traffic using a 64- or 128-bit key in hexadecimal. This is a static key, which means all traffic, regardless of device, is encrypted using a single key. A WEP key allows computers on a network to exchange encoded messages while hiding the messages' contents from intruders. This key is what is used to connect to a wireless-security-enabled network.

One of WEP's main goals was to prevent Man-in-the-Middle attacks, which it did for a time. However, despite revisions to the protocol and increased key size, various security flaws were discovered in the WEP standard over time. As computing power increased, it became easier to exploit for criminals to exploit those flaws. Because of its vulnerabilities, the Wi-Fi Alliance officially retired WEP in 2004. Today, WEP security is considered obsolete, although it is still sometimes in use – either because network administrators haven't changed the default security on their wireless routers or because devices are too old to support newer encryption methods like WPA.

ROLE OF WEP

The Wired Equivalent Privacy protocol adds security similar to a wired network's physical security by encrypting data transmitted over the WLAN. Data encryption protects the vulnerable wireless link between clients and access points.

After WEP secures wireless data transmissions, other LAN security mechanisms can ensure privacy and data confidentiality. These include password protection, end-to-end encryption, virtual private networks and authentication.

The basic network security services the protocol provides for wireless networks include the following:

- **Privacy.** WEP initially used a 64-bit key with the RC4 stream encryption algorithm to encrypt data transmitted wirelessly. Later versions of the protocol added support for 128-bit keys and 256-bit keys for improved security. WEP uses a 24-bit initialization vector, which resulted in effective key lengths of 40, 104 and 232 bits.
- **Data integrity.** WEP uses the CRC-32 checksum algorithm to check that transmitted data is unchanged at its destination. The sender uses the CRC-32 cyclic redundancy check to generate a 32-bit hash value from a sequence of data. The recipient uses the same check on receipt. If the two values differ, the recipient can request a retransmission.
- **Authentication.** WEP authenticates clients when they first connect to the wireless network access point. It enables authentication of wireless clients with these two mechanisms:

1. **Open System Authentication.** With OSA, Wi-Fi-connected systems can access any WEP network access point, as long as the connected system uses a service set identifier that matches the access point SSID.
2. **Shared Key Authentication.** With SKA, Wi-Fi-connected systems use a four-step challenge-response algorithm to authenticate.

Drawbacks to Wired Equivalent Privacy

WEP is widely implemented and deployed, but it suffers from serious security weaknesses. These include:

- **Stream cipher.** Encryption algorithms applied to data streams, called stream ciphers, can be vulnerable to attack when a key is reused. The protocol's relatively small key space makes it impossible to avoid reusing keys.
- **RC4 weaknesses.** The RC4 algorithm itself has come under scrutiny for cryptographic weakness and is no longer considered safe to use.
- **Optional.** As designed, the protocol use is optional. Because it's optional, users often failed to activate it when installing WEP-enabled devices.
- **Shared key.** The default configuration for these systems uses a single shared key for all users. You can't authenticate individual users when all users share the same key.

These weaknesses doomed WEP. Most standards bodies deprecated the protocol soon after the Wi-Fi Protected Access (WPA) protocol became available in 2003.

WEP vs. WPA

The IEEE introduced Wired Equivalent Privacy in the 802.11 wireless networking standard in 1997 and then released WPA as a proposed replacement five years later. Efforts to fix WEP during its short lifetime failed to produce a secure solution to wireless network access. WPA2 formally replaced it in 2004.

WEP variants and improved versions of WPA include the following protocols:

- **WEP2.** After security issues emerged, changes to the IEEE specifications increased the WEP key length to 128 bits and required the use of Kerberos authentication. However, these changes proved insufficient to make WEP more secure and were dropped from the standard.
- **WEPplus or WEP+.** Agere Systems, an integrated circuit component company, developed this proprietary variant. WEP+ eliminated weak keys from the key space. However, fundamental issues remained, and only Agere Systems Wi-Fi products used WEP+.
- **WPA.** The first version of WPA increased key length to 128 bits, and replaced the CRC-32 integrity check with the Temporal Key Integrity Protocol. However, WPA still uses the RC4 encryption algorithm, and retained other weaknesses from WEP.
- **WPA2.** This WPA update added stronger encryption and integrity protection. It uses the Counter Mode Cipher Block Chaining Message Authentication Code Protocol, which incorporates the Advanced Encryption Standard algorithm for encryption and integrity verification of wireless transmissions. WPA2 comes in the following two modes:
    1. **WPA2-Enterprise** requires a Remote Authentication Dial-In User Service authentication server to authenticate users.