Aim-: Analysis the Security Vulnerabilities of E-commerce services.

EXPERIMENT-5

5.0 Learning Objectives

After going through this session, you should be able to:

* Know about Security Vulnerabilities of E-commerce services.
* Identify the vulnerabilities input validations and database servers.
* Point out the vulnerabilities in TCP/IP Protocols used for communications.

5.1 Security Vulnerabilities of E-commerce services

Vulnerability is a weakness which allows an attacker to reduce a system's information assurance. Vulnerability is the intersection of three elements: a system susceptibility or flaw, attacker access to the flaw, and attacker capability to exploit the flaw. To exploit vulnerability, an attacker must have at least one applicable tool or technique that can connect to a system weakness

5.1.1 Software life cycle not secure.

Over the years, efforts to enhance software development life cycle (SDLC) practices have been shown to improve software quality, reliability, and fault-tolerance. Now-a-days strategies to improve the security of software in organizations such as Microsoft, Oracle, and Motorola have resulted in software products with less vulnerabilities and greater dependability, trustworthiness, and robustness.

As per the SANS Institute’s Top 20 list of security vulnerabilities, the MITRE Common Vulnerabilities and Exposures (CVE) site, the US-CERT Technical Cyber Security Alerts site, and the Microsoft Security Advisory site show that common software defects are the leading cause of security vulnerabilities (buffer overflows have been the most common software defect leading to security vulnerabilities).

Some of the things that can be incorporated in SDLC are: 1. Software should be installed using security defaults

2. A software patch management process should be there.

5.1.2 Vulnerabilities due to input validations

Buffer Overflow: A buffer overflow condition occurs when a program attempts to copy more data in a buffer than it can hold. Buffer overflow is probably the best known form of software security vulnerability. At the code level, buffer overflow vulnerabilities usually involve the violation of a programmer's assumptions. Hackers use buffer overflows to corrupt the execution stack of a web application. Buffer overflow flaws can be present in both the web server or application server products that serve the static and dynamic aspects of the site. Buffer overflows generally resulted in to crashes. Other type of attacks will create the situation like lack of availability are possible, including putting the program into an infinite loop.

5.1.3 Log Forging: Writing invalidated user input to log files can give access to attacker for forging log entries or injecting malicious content into the logs. Log forging vulnerabilities occur in following conditions:

* + 1. Data copied to an application from an unreliable source.
    2. The data is copied to an application or system log file.

Applications uses log file to store a history of events for later review and record, statistics gathering, or debugging. Analysis of the log files may be misdirected if an attacker can supply inappropriate data to the application. In the most common case, an attacker may be able to insert false entries into the log file by providing the application with input that includes appropriate characters. If the log file is processed automatically, the attacker can render the file unusable by corrupting the format of the file or injecting unexpected characters. A more dangerous attack might involve changing the log file statistics.

5.1.4 Missing XML Validation: Failure to implement validation when parsing XML gives an attacker the way to supply malicious input. By accepting an XML document without validating it against a DTD or XML schema, the programme gives chance to attackers to copy unexpected, unreasonable, or malicious input. It is not possible for an XML parser to validate all aspects of a document's content; a parser cannot understand the complete semantics of the data. However, a parser can do a complete and thorough job of checking the document's structure and therefore guarantee to the code that processes the document that the content is well-formed.

5.1.5 Validation checks in client: Performing validation check in client side code, mostly JavaScript, provides no protection for server-side code. An attacker can simply disable JavaScript, use telnet, or use a security testing proxy to bypass the client side validation. Client-side validation is widely used, but is not security relevant.

5.1.6 Vulnerabilities in database servers: There are various techniques to attack a database. External attacks may exploit configuration weaknesses that expose the database server. Also weak and insecure Web application can be used to exploit the database. An application with excess privilege in the database can put database at risk. The main threats to a database server are:

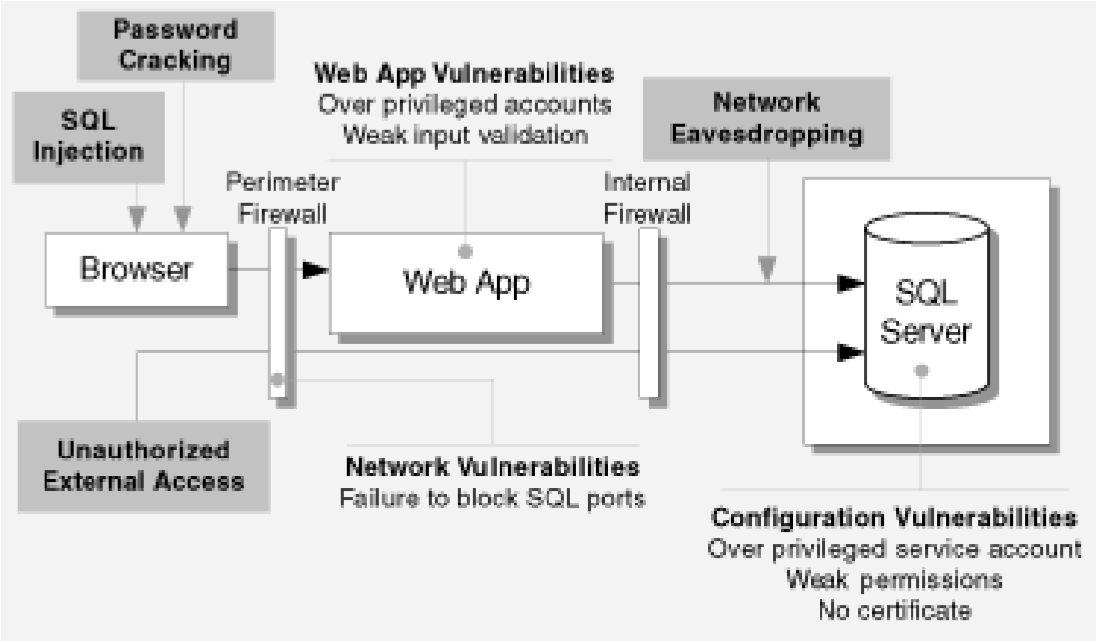


Fig 1: Main threats to a database server

* + - SQL injection: Technique used to attack database through website entry fields.
    - Network eavesdropping: It is a network level attack consisting of capturing packets from the networked computers.
    - Unauthorized server access: Attacked made unauthorised access through various loopholes in the system such as O/S, non availability of firewall etc.
    - Password cracking: Technique of recovering password from data stored in computer.

5.1.7 Vulnerabilities in TCP/IP Protocols used for communications TCP/IP is very popular and known to everyone, IP – (Internet Protocol) that handles routing packets of data from one computer to another or from one router to another. TCP, (Transmission Control Protocol) , deals with ensuring that the data packets are delivered in a reliable manner from one computer to another.

5.2 Major causes of vulnerabilities

* + - Dependency on IP source address for authentication
    - Minimal/no authentication in network control mechanisms, e.g. routing protocol, congestion control, flow control, ICMP messages, etc.

5.2.1 Vulnerabilities in firewall

Firewall vulnerability is defined as an error made during firewall design, implementation, or configuration that can be exploited to attack the trusted network that the firewall is supposed to protect. For example, common firewall vulnerabilities and improper configurations include:

* + 1. ICMP allowed, e.g., the firewall can be ping-ed;
    2. Provides the attacker with additional information, or improves the speed of the attacker’s port scan by doing Denial rather than drop of traffic to ports by the firewall suppose to block;
    3. Misconfiguration that allows a TCP ping of internal hosts with Internet-routable IP addresses (e.g., in-bound TCP 80 is not restricted to the web server);
    4. Trust or unrestricted access to certain IP addresses;
    5. Availability of extra/ non required services on the firewall;
    6. Unnecessarily open TCP and UDP ports;

5.2.2 Vulnerability in IPS: The main function of intrusion prevention systems is to identify malicious activity, log information about malicious activity, attempt to block/stop activity, and report activity. Some of the IPS Vulnerabilities are as follows:

* + 1. Under estimation of security capabilities, including information gathering, logging, detection, and prevention.
    2. Focus on Performance rather than security, including maximum capacity and performance features.
    3. Non-defined Management policies, including design and implementation (e.g., reliability, interoperability, scalability, product security), operation and maintenance (including software updates), and training, documentation, and technical support.

5.2.3 Vulnerability loopholes of the users

* + 1. Tolerating weak passwords: weak passwords are arguably the most nonsensical, yet simplest security flaws to fix.
    2. Connecting to unsecured Wi-Fi hotspots: Many people don’t think twice about logging onto a random (and unprotected) wireless network just to get some work done. That’s all it takes for someone with ill intent to capture a user’s login credentials and work his way onto your wireless network.
    3. Ignorance in encrypting hard drives and USB storage disks: Simply encrypting computer hard drives can eliminate a huge portion of information risks.
    4. Assuming that patches are under control: There are typically hundreds of missing patches on both workstations and servers. In many situations, admins are unaware of specific patches to be installed.
    5. Not balancing security with convenience: Unintended acts, security controls often get in the way of users, who then find ways around it. General habit of writing passwords on sticky notes is just the beginning.