LECTURE 6: WEB APPLICATION SECURITY

INTRODUCTION

Web application security is the practice of protecting websites, applications, and APIs from attacks. It is a broad discipline, but its ultimate aims are keeping web applications functioning smoothly from cyber vandalism, data theft, unethical competition, and other negative consequences.

Web application security is difficult because these applications are, by definition, exposed to the general public, including malicious users. Additionally, input to web applications comes from within HTTP requests. Correctly processing this input is difficult. The incorrect or missing input validation causes most vulnerabilities in web applications.

web security risks change over time as new vulnerabilities are discovered (or invented). And it’s not all doom and gloom; new defenses are developed every year too. New versions of application frameworks, web servers, operating systems, and web browsers all often add defensive technology to prevent vulnerabilities or limit the impact of a successful attack.

Network firewalls, network vulnerability scanners, and the use of Secure Socket Layer (SSL) are generally used but do not make a web site secure fully. It is estimated that over 70% of attacks against a company's web site or web application come at the application layer, not the network or system layer.

WEB APPLICATION: DEFINITION

The Web Application Security Consortium (WASC) defines a web application as “a software application, executed by a web server, which responds to dynamic web page requests over HTTP.” A web application is comprised of a collection of scripts, which reside on a web server and interact with databases or other sources of dynamic content.

Using the infrastructure of the Internet, web applications allow service providers and clients to share and manipulate information in a platform-independent manner.

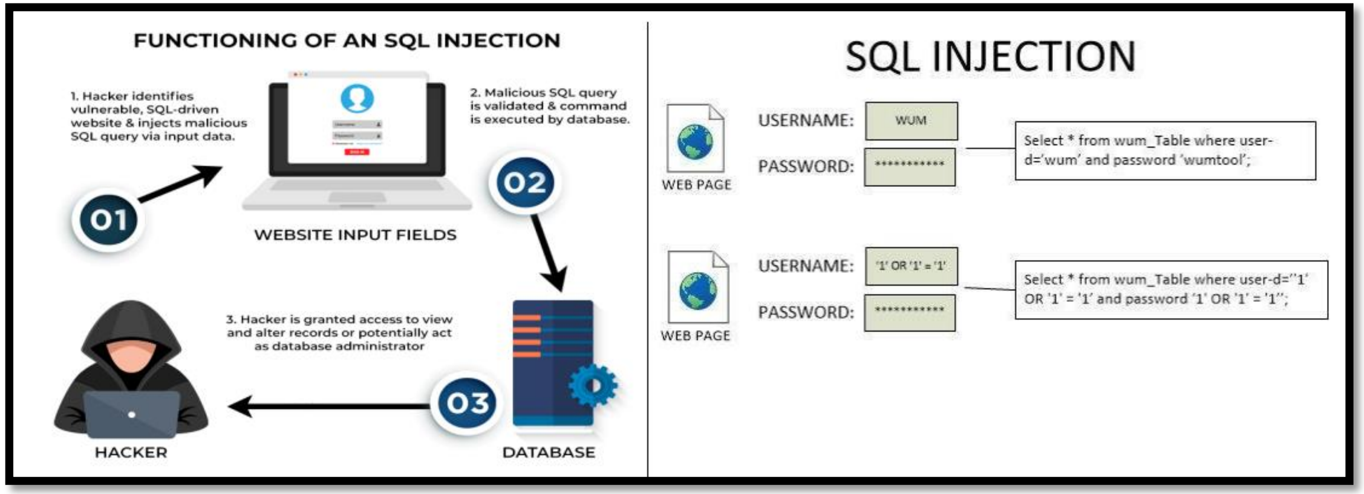
COMMON WEB VULNERABILITIES

1. SQL injection

SQL injection is a web security vulnerability that malicious users can inject SQL commands into an SQL statement, via web page input. Injected SQL commands can alter SQL statement and compromise the security of a web application.

SQL Injections happen when a developer accepts user input that is directly placed into a SQL Statement and doesn’t properly validate and filter out dangerous characters. This can allow an attacker to alter SQL statements passed to the database as parameters and enable her to not only steal data from your database, but also modify and delete it.

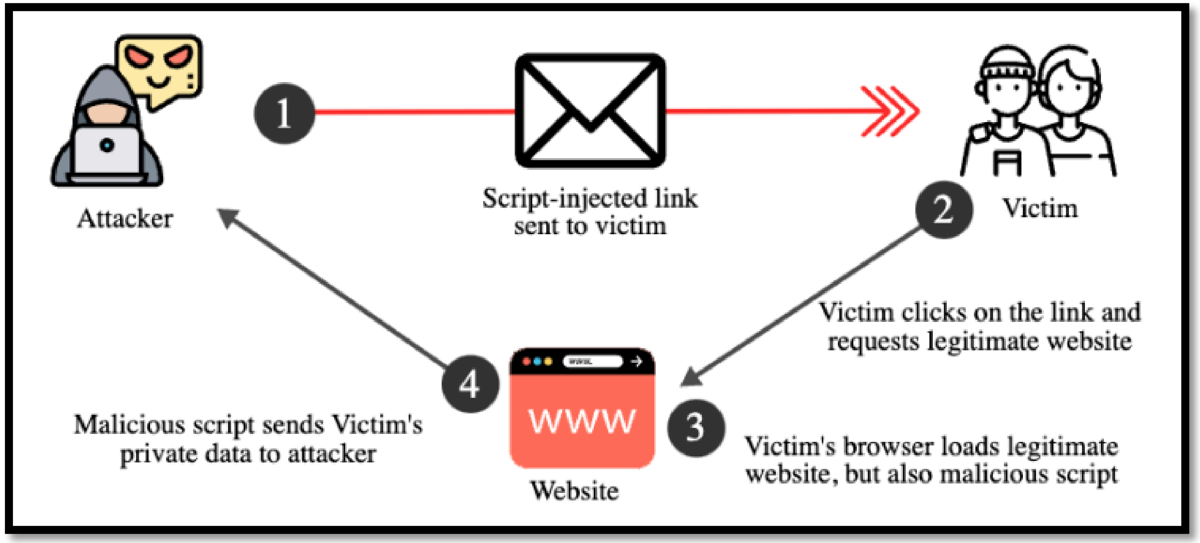
A database is vulnerable to SQL injections when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed. SQL injection attacks are also known as SQL insertion attacks.


FUNCTIONING OF AN SQL INJECTION
1. Hacker identifies vulnerable, SQL-driven website & injects malicious SQL query via input data.
2. Malicious SQL query is validated & command is executed by database.
3. Hacker is granted access to view and alter records or potentially act as database administrator
01
02
03
WEBSITE INPUT FIELDS
HACKER
DATABASE
SQL INJECTION
WEB PAGE
USERNAME:
WUM
PASSWORD:
****
Select * from wum_Table where user-d='wum' and password 'wumtool';
WEB PAGE
USERNAME:
'1' OR '1' = '1'
PASSWORD:
****
Select * from wum_Table where user-d=''1' OR '1' = '1' and password '1' OR '1' = '1'';

COMMON WEB VULNERABILITIES

2. Cross-Site Scripting (XSS)

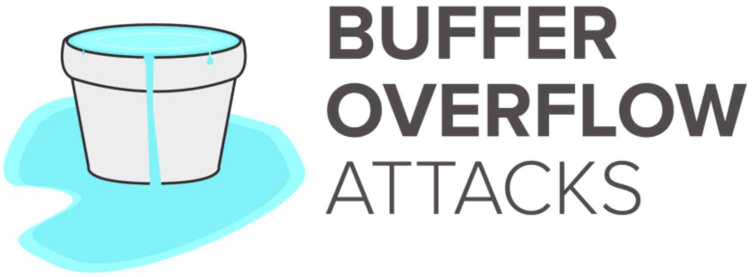
Cross-site scripting vulnerabilities are actually a specific type of injection vulnerability in which the attacker injects his own script code (such as JavaScript) or HTML into a vulnerable web page. At first glance, this may not seem like an incredibly critical vulnerability, but attackers have used cross-site scripting holes to steal victims’ login passwords, set up phishing sites, and even to create self-replicating worms that spread throughout the target web site.


Attacker
1
Script-injected link sent to victim
Victim
2
Victim clicks on the link and requests legitimate website
4
WWW
Website
3
Malicious script sends Victim's private data to attacker
Victim's browser loads legitimate website, but also malicious script

COMMON WEB VULNERABILITIES

3. buffer overflow

A buffer overflow vulnerability condition exists when an application attempts to put more data in a buffer than it can hold. Writing outside the space assigned to buffer allows an attacker to overwrite the content of adjacent memory blocks causing data corruption, crash the program, or the execution of an arbitrary malicious code

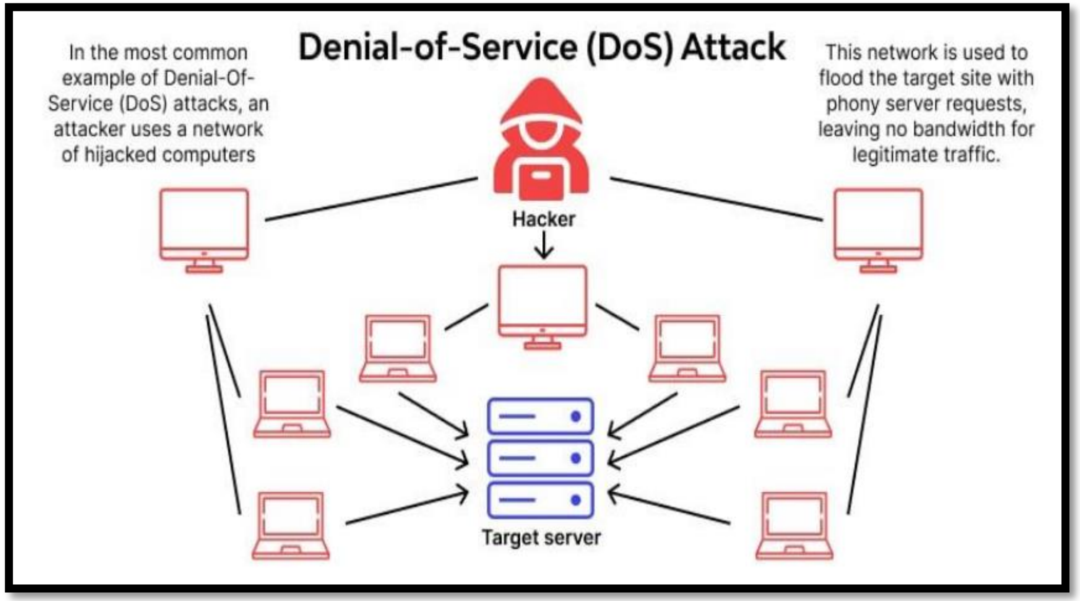


COMMON WEB VULNERABILITIES

4. Denial of Service (DoS)

A denial-of-service (DoS) attack occurs when legitimate users are unable to access information systems, devices, or other network resources due to the actions of a malicious cyber threat actor. Services affected may include email, websites, online accounts (e.g., banking), or other services that rely on the affected computer or network.

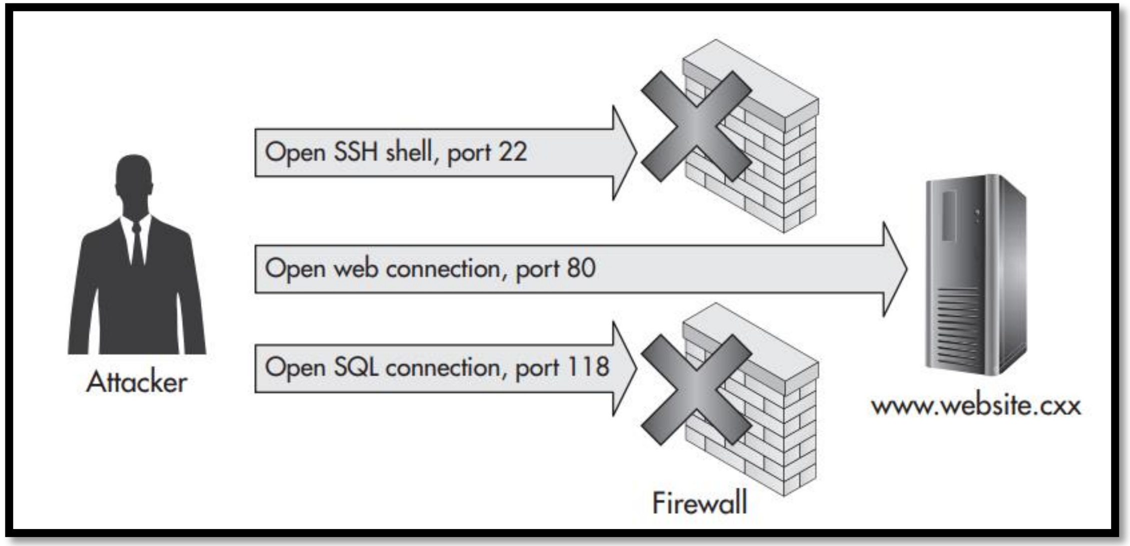
A denial-of-service condition is accomplished by flooding the targeted host or network with traffic until the target cannot respond or simply crashes, preventing access for legitimate users. DoS attacks can cost an organization both time and money while their resources and services are inaccessible.


Denial-of-Service (DoS) Attack
In the most common example of Denial-Of-Service (DoS) attacks, an attacker uses a network of hijacked computers
This network is used to flood the target site with phony server requests, leaving no bandwidth for legitimate traffic.
Hacker
Target server

WEB APPLICATION FIREWALLS (WAFS)

A web application firewall (WAF) protects web applications by monitoring and filtering internet traffic that flows between an application and the internet. In this way, a WAF works as a secure web gateway (SWG). It provides protection for web applications against attacks, including cross-site scripting, Structured Query Language (SQL) injection, and other threats.

In the Open Systems Interconnection (OSI) model, a WAF works within Layer 7. Even though it works against many internet threats, it is not intended to defend against all kinds of threats. A WAF often works within a suite of protective tools meant to defend a network, computer, or application.


Open SSH shell, port 22
Open web connection, port 80
Open SQL connection, port 118
Attacker
www.website.cxx
Firewall