Web App Sec_TA-2 _Assignment

Q1. Write about HTTP vs HTTPS.

Ans:- HTTP :- Full form of HTTP is Hypertext Transfer Protocol. HTTP offers set of rules and standards which govern how any information can be transmitted on the World Wide Web. HTTP provides standard rules for web browsers & servers to communicate.

HTTPS:- HTTPS stands for Hyper Text Transfer Protocol Secure. It is highly advanced and secure version of HTTP. It uses the port no. 443 for Data Communication. It allows the secure transactions by encrypting the entire communication with SSL. It is a combination of SSL/TLS protocol and HTTP. It provides encrypted and secure identification of a network server.

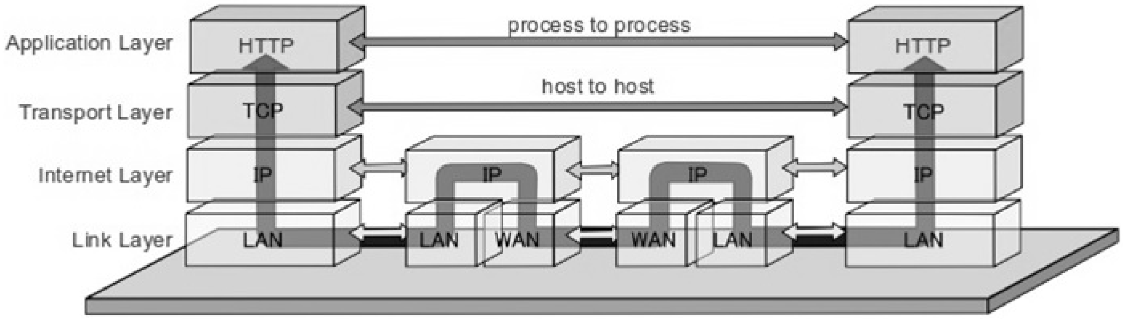
Parameter	**HTTP**	**HTTPS**
Protocol	It is hypertext transfer protocol.	It is hypertext transfer protocol with secure.
Security	It is less secure as the data can be vulnerable to hackers.	It is designed to prevent hackers from accessing critical information. It is secure against such attacks.
Port	It uses port 80 by default	It was use port 443 by default.

Q2. Explain about TCP Architecture and TCP Protocol.

Ans:- TCP protocol stands for the Transmission Control Protocol.This protocol tells us how data should be addressed, stored, transferred, coded and decoded by the devices for end-to-end communication between devices. It is the main protocol of the Internet Protocol Suite. TCP provides a reliable, safe and error-free transfer of data between applications running on hosts through an Internet media.

TCP Architecture

It is a four-layered protocol stack. It helps in the interconnection of network devices over the internet. Each layer contains certain protocols that help in the functioning of the layer. The four layers of TCP/IP protocol are Application Layer, Transport Layer, Networking/Internet Layer and the Data Link/physical layer.



Q3. Explain Encoding and types of Encoding.

Ans:- The process of conversion of data from one form to another form is known as Encoding. It is used to transform the data so that data can be supported and used by different systems. Encoding works similarly to converting temperature from centigrade to Fahrenheit.

Encoding is used in mainly two fields:

1. **Encoding in Electronics:** In electronics, encoding refers to converting analog signals to digital signals.
2. **Encoding in Computing:** In computing, encoding is a process of converting data to an equivalent cipher by applying specific code, letters, and numbers to the data.

Type of Encoding Technique

1. Character Encoding
2. Image & Audio and Video Encoding

1. Character Encoding:- Character encoding encodes characters into bytes.

There are different types of Character Encoding techniques, which are given below:

1. HTML Encoding
2. URL Encoding
3. Unicode Encoding
4. Base64 Encoding
5. Hex Encoding
6. ASCII Encoding

2. Image and Audio & Video Encoding

Image and audio & video encoding are performed to save storage space

Q4. Elaborate fingerprinting in web server?

Ans:- One of the first tasks when conducting a web application penetration test is to try to identify the version of the web server and the web application.The reason for that is that it allows us to discover all the well-known vulnerabilities that are affecting the web server and the application.This process is called web application fingerprinting

The web application fingerprinting can be done with the use of a variety of tools or manually.

1. **Manual Fingerprinting :-** This can be done with the use of different utilities such as the telnet or the netcat.
2. **Automated Fingerprinting:-** Web application fingerprinting can be done as well with the use of automated tools that have been designed for that purpose.

Q5. Explain Virtual Host and Explain how to find the virtual host.

Ans:- Virtual Host:- What are virtual hosts?

A web server can host more than one site. Each of this site is a virtual host. The name (identifier) of the virtual host can be:

- Domain name

- Subdomain (of any level, for example, kali.tools, en.kali.tools, test.en.kali.tools – these are all virtual hosts that can have different settings and even different IP addresses)
- Arbitrary names (any string that is not a valid domain name or subdomain)

This tool uses multiple discovery techniques, such as:

- Searching in public search engines
- DNS resolutions
- Analyzing web redirects
- Searching in SSL certificates

Q6. Explain Google Hacking and steps of google hacking.

Ans:- Google hacking, sometimes, referred to as Google dorking, is an information gathering technique used by an attacker leveraging advanced Google searching techniques. Google hacking search queries can be used to identify security vulnerabilities in web applications, gather information for arbitrary or individual targets, discover error messages disclosing sensitive information, discover files containing credentials and other sensitive data.

For example:- intitle:"index of" filetype:sql"

Steps of google hacking:-

1. **Intitle**: This will ask google to show pages that have the term in their html title.
2. **Inurl**: Searches for specified term in the URL. For example: inurl:register.php
3. **Filetype**: Searched for certain file type. Example: filetype:pdf will search for all the pdf files in the websites.
4. **Ext**: It works similar to filetype. Example: ext:pdf finds pdf extension files.
5. **Intext**: This will search content of the page. This works somewhat like plain google search

Q7. What is subdomain enumeration?

Ans:- Sub-domain enumeration is the process of finding sub-domains for one or more domains. It helps to broader the attack surface, find hidden applications, and forgotten subdomains.

1. **Passive Enumeration**
    - Certificate Transparency
    - Google Dorking
    - DNS Aggregators
    - ASN Enumeration
    - Subject Alternate Name (SAN)
    - Rapid7 Forward DNS dataset
2. **Active Enumeration**
    - Brute Force Enumeration
    - Zone Transfer
    - DNS Records
    - Content Security Policy (CSP) Header

Q8. Write up about enumerating resources?
Ans:- In certain situations, you might want to discover the resource contents of an unknown portable executable (PE) module. The Windows SDK provides resource enumeration functions that enable your application to obtain lists of resource types, names, and languages in a specified module.

- The EnumResourceTypeW and EnumResourceTypesExW functions provide a list of resource types found in the module.
- The EnumResourceNamesW and EnumResourceNamesExW functions provide the name of each resource within a given type.
- The EnumResourceLanguagesW and EnumResourceLanguagesExW functions provide the language of each resource of a given name and type.

These functions and their associated callback functions enable your application to create a list of all resources in a module. This process is described in Creating a resource list.

Q9. Define following terminology.
CVE:- CVE, short for Common Vulnerabilities and Exposures, is a list of publicly disclosed computer security flaws. When someone refers to a CVE, they mean a security flaw that's been assigned a CVE ID number.

CWE:- Common Weakness Enumeration (CWE™) is a community-developed list of common software and hardware weakness types that have security ramifications. “Weaknesses” are flaws, faults, bugs, or other errors in software or hardware implementation, code, design, or architecture that if left unaddressed could result in systems, networks, or hardware being vulnerable to attack.

CVSS:- The Common Vulnerability Scoring System (aka CVSS Scores) provides a numerical (0-10) representation of the severity of an information security vulnerability. CVSS scores are commonly used by infosec teams as part of a vulnerability management program to provide a point of comparison between vulnerabilities, and to prioritize remediation of vulnerabilities.

OWAPS:- The Open Web Application Security Project® (OWASP) is a nonprofit foundation that works to improve the security of software. Through community-led open-source software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web.

- Tools and Resources
- Community and Networking
- Education & Training

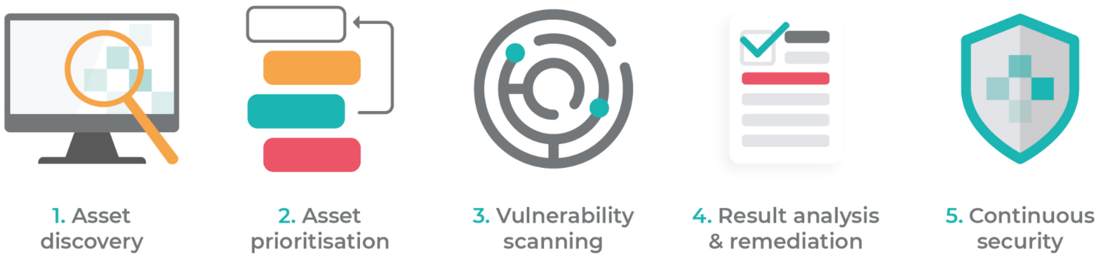
STRIDE:- STRIDE methodology aims to ensure that an application meets the security requirements of Confidentiality, Integrity, and Availability (CIA), besides Authorisation, Authentication, and Non-Repudiation.
STRIDE is an acronym. It stands for

- Spoofing
- Tampering
- Repudiation
- Information disclosure
- Denial of service (DoS)
- Elevation of privilege

DREAD:- DREAD is part of a system for risk-assessing computer security threats that was formerly used at Microsoft.[1] It provides a mnemonic for risk rating security threats using five categories.
The categories are:

- Damage – how bad would an attack be?
- Reproducibility – how easy is it to reproduce the attack?
- Exploitability – how much work is it to launch the attack?
- Affected users – how many people will be impacted?
- Discoverability – how easy is it to discover the threat?

Q10. Explain Vulnerability Assessment and Life Cycle of VA.
Ans:- A vulnerability assessment is a systematic review of security weaknesses in an information system. It evaluates if the system is susceptible to any known vulnerabilities, assigns severity levels to those vulnerabilities, and recommends remediation or mitigation, if and whenever needed.
Life Cycle of VA:-



1. **Asset discovery** :- First, you need to decide what you want to scan.
2. **Prioritization**:- Once you know what you've got, the next question is whether you can afford to run a vulnerability assessment on all of it.
3. **Vulnerability scanning**:- Vulnerability scanners are designed to identify known security weaknesses and provide guidance on how to fix them.

4. **Result analysis & remediation**:- After the vulnerability scan is complete, the scanner provides an assessment report. When reading and developing remediation plans based on this report
5. **Continuous cyber security**:- A vulnerability scan provides a point in time snapshot of the vulnerabilities present in an organization's digital infrastructure. However, new deployments, configuration changes, newly discovered vulnerabilities, and other factors can quickly make the organization vulnerable again. For this reason, you must make vulnerability management a continuous process rather than a one-time exercise.

Q11. Short note on Unknown Vulnerability with Examples?

Ans:- A vulnerability is a weakness that can be exploited by cybercriminals to gain unauthorized access to a computer system. After exploiting a vulnerability, a cyberattack can run malicious code, install malware, and even steal sensitive data.Vulnerabilities can be exploited by a variety of methods, including SQL injection, buffer overflows, cross-site scripting (XSS), and open-source exploit kits that look for known vulnerabilities and security weaknesses in web applications.Many vulnerabilities impact popular software, placing the many customers using the software at a heightened risk of a data breach, or supply chain attack. Such zero-day exploits are registered by MITRE as a Common Vulnerability Exposure (CVE).

1. **Hardware -** Any susceptibility to humidity, dust, soiling, natural disaster, poor encryption, or firmware vulnerability.
2. **Software -** Insufficient testing, lack of audit trail, design flaws, memory safety violations (buffer overflows, over-reads, dangling pointers), input validation errors (code injection, cross-site scripting (XSS), directory traversal, email injection, format string attacks, HTTP header injection, HTTP response splitting, SQL injection), privilege-confusion bugs (clickjacking, cross-site request forgery, FTP bounce attack), race conditions (symlink races, time-of-check-to-time-of-use bugs), side channel attacks, timing attacks and user interface failures (blaming the victim, race conditions, warning fatigue).
3. **Network -** Unprotected communication lines, man-in-the-middle attacks, insecure network architecture, lack of authentication, default authentication, or other poor network security.
4. **Personnel -** Poor recruiting policy, lack of security awareness and training, poor adherence to security training, poor password management, or downloading malware via email attachments.
5. **Physical site -** Area subject to natural disaster, unreliable power source, or no keycard access.

Q12. What is false positive in web application security?

Ans:- False positives occur when a scanning tool, web application firewall (WAF), or intrusion prevention system (IPS) incorrectly flag a security vulnerability during software testing. False positives describe the situation where a test case fails, but in actuality there is no bug and functionality is working correctly. Because false positives need to be checked out and this can be a time-consuming process, they typically eat up valuable IT bandwidth that should be applied to more important tasks.

Q13.Elaborate secure source code review process.
Ans:- A Secure Code Review is a manual or automated technique that examines an application's code base to discover existing flaws and vulnerabilities. The process also checks for logical errors and inspects spec implementation and style guidelines.

Code review can be of two types – Manual and Automated Code Review. Automated Code Review involves a tool that reviews the application source code using a predefined set of rules. On the other hand, manual review involves a human element inspecting the source code line by line to detect susceptibilities.

the combination of both automated and manual approaches. The human element that the manual review involves is vital, and if you combine it with the SAST tool functionalities, it can enhance the overall security of the code. Additionally, it helps minimize the number of flaws or susceptibilities flowing into the production cycle.

Q14. What is vulnerability Scanner?
Ans:- A vulnerability scanner is an automated vulnerability testing tool that monitors for misconfigurations or coding flaws that pose cybersecurity threats. Vulnerability scanners either rely on a database of known vulnerabilities or probe for common flaw types to discover unknown vulnerabilities. The scanner logs detect vulnerabilities and sometimes assign a risk score.

Q15. What is web authentication Mechanism?
Ans:- An authentication mechanism defines rules about security information, such as whether a credential is forwardable to another Java™ process, and the format of how security information is stored in both credentials and tokens. You can select and configure an authentication mechanism by using the administrative console.

<u>Authenticate only when the URI is protected:-</u> Specifies that the web client can retrieve an authenticated identity only when it accesses a protected Uniform Resource Identifier (URI). WebSphere Application Server challenges the web client to provide authentication data when the web client accesses a URI that is protected by a J2EE role. This default option is also available in previous versions of WebSphere Application Server.

<u>Use available authentication data when an unprotected URI is accessed:-</u> Specifies that the web client is authorized to call the getRemoteUser, isUserInRole, and getUserPrincipal methods; retrieves an authenticated identity from either a protected or an unprotected URI. Although the authentication data is not used when you access an unprotected URI, the authentication data is retained for future use. This option is available when you select the Authentication only when the URI is protected check box.

Q16. Explain Authentication and authorization bypass in detail?

Ans:- Authentication bypass vulnerability allows unauthenticated users to escalate privileges to a device or application. The attacker could further create a legit admin session with the 'username=admin' cookie in the HTTP request. After gaining access, the criminal can download malicious firmware and modify the system's settings. A successful attack enables attackers to view, edit, delete, copy, and overwrite data on the connected devices.

Authorization bypass:-This is simply any place within an application that the user can access something the user shouldn't be able to. Any situation where a user can view or manipulate content in an unintended manner counts as an instance where the user is bypassing my intended level of authorization. This includes lateral movement (accessing data of another account at the same privilege level) and vertical movement or privilege escalation (accessing content the user shouldn't have access to in my current role). So this is clearly a very generic description of a vulnerability, and as you might be able to guess at this point, there are many different ways that this can be exploited in practice.

Q17. What is burp sequencer?
Ans:- Burp Sequencer is a tool for analyzing the quality of randomness in a sample of data items. The data items can either be application's session ID's, CSRF tokens, password reset or forget password tokens or any specific unpredictable ID generated by the application.

The Burp Sequencer is one of the most amazing tools that try to capture the randomness or the variances in the session ID's by employing some standard statistical tests which are based on the principle of testing a hypothesis against a sample of evidence, and calculating the probability of the observed data occurring.

Q18:- How to analyzing website content?
Ans:- Website analysis is the practice of testing and analyzing a website's performance in relation to SEO, speed, competition, and traffic.

Any site can benefit from some form of website analysis if the results are then used to improve it—for example, by reducing page size to increase overall speed or optimizing a landing page with lots of traffic for more conversions.

Traditional website analysis roughly falls into 4 categories:

1. Search Engine Optimization (SEO)
2. Speed
3. Competition
4. Traffic

Q19. Explain the term fuzzing with burp intruder?
Ans:- fuzzing:- Is a way to automate your process of finding bugs/vulnerabilities by sending a lot of requests to an application with different data, expecting that the application trigger an

action. It doesn't specific to Web Applications and can be used to a lot of services and attacks like buffer overflow. You can fuzz a web application for find several vulnerabilities, like XSS, SQL Injection, LFI, SSRF and etc. It will only depend of the content of your word list and the scenario.

fuzzing with burp intruder:- Burp Suite comes with an integrated HTML Fuzzer, commonly termed as a Burp Intruder. This burp intruder gives us several opportunities to fuzz the injection points in the most customizable way we can.

In order to make a fuzzing attack possible, we need to add up a dictionary as a payload list. However, Burp Suite's Professional Edition gives us an option to opt the predefined lists containing the most common fuzz strings according to the attack types.

- First, we need to intercept the HTTP Request, therewith that we'll thus share it with the Intruder.
- As soon as we do so, we'll define the parameters or the injection points where the fuzzing needs to be done.
- Now, at last, the attack type and payloads list need to be defined up with that.

Q20. Explain OWAPS top 10 attack with prevention mechanism?
Ans:- OWASP basically stands for the Open Web Application Security Project, it is a non-profit global online community consisting of tens of thousands of members and hundreds of chapters that produces articles, documentation, tools, and technologies in the field of web application security.
The top 10 OWASP vulnerabilities in 2020 are:

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities (XXE)
5. Broken Access control
6. Security misconfigurations
7. Cross-Site Scripting (XSS)
8. Insecure Deserialization
9. Using Components with known vulnerabilities
10. Insufficient logging and monitoring.

1. **Injection**
   Injection vulnerabilities occur when an attacker uses a query or command to insert untrusted data into the interpreter via SQL, OS, NoSQL, or LDAP injection.

   Injection attacks can be prevented by
   - Using safer API which avoids the use of the interpreter
   - Using parameterized queries when coding

2. **Broken Authentication**
   Broken Authentication is a vulnerability that allows an attacker to use manual or automatic methods to try to gain control over any account they want in a system.

   Broken authentication attacks can be prevented by
   - Implementing multi-factor authentication
   - Protecting user credentials

3. **Sensitive Data Exposure**
   This vulnerability is one of the most widespread vulnerabilities on the OWASP list and it occurs when applications and APIs don't properly protect sensitive data such as financial data, social security numbers, usernames, and passwords, or health information, and this enables attackers to gain access to such information and commit fraud or steal identities.

   Sensitive data exposure attacks can be prevented by
   - Using the secure URL's
   - Using strong and unique passwords

4. **XML External Entities (XXE)**
   This vulnerability occurs for web applications that parse XML input.

   XXE attacks can be prevented by
   - Using less complex data formats such as JSON
   - Keeping XML processors and libraries upgraded

5. **Broken Access Controls**
   This vulnerability occurs when there is broken access to resources, it means there are some improperly configured missing restrictions on authenticated users which allows them to access unauthorized functionality or data like access to others accounts, confidential documents, etc.

   Broken access control attacks can be prevented by
   - Deleting accounts that are no longer needed or are not active
   - Shutting down unnecessary services to reduce the burden on servers

6. **Security Misconfiguration**
   This vulnerability refers to the improper implementation of security intended to keep application data safe.

   Security misconfiguration attacks can be prevented by
   - Using Dynamic application security testing (DAST)
   - Disabling the use of default passwords

7. **Cross-Site Scripting (XSS)**
   XSS vulnerability allows a hacker to inject malicious client-side scripts into a website and then use the web application as an attack vector to hijack user sessions, or redirecting the victim to malicious websites.

   Cross-site scripting attacks can be prevented by
   - Using appropriate response headers

- Filtering the input and encoding the output

8. **Insecure Deserialization**
   Insecure Deserialization vulnerability allows an attacker to remotely execute code in the application, tamper or delete serialized (written to disk) objects, conduct injection attacks, replay attacks, and elevate privileges.

   Insecure Deserialization attacks can be prevented by
   - Implementing digital signatures
   - Using penetration testing

9. **Using Components with known vulnerabilities**
   It also occurs because developers frequently don’t know which open source and third-party components are present in their applications and this makes it difficult for developers to update components when new vulnerabilities are discovered in their current versions.

   This attack can be prevented by
   - Removing all unnecessary dependencies
   - Using virtual patching

10. **Insufficient Logging and Monitoring**
    Insufficient logging and ineffective integration of the security systems allow attackers to pivot to other systems and maintain persistent threats.

    Insufficient logging and monitoring attacks can be prevented by
    - Implementing logging and audit software
    - Establishing an effective monitoring system