AI FOR CYBER SECURITY WITH IBM QRADAR

Name: J. Manasa

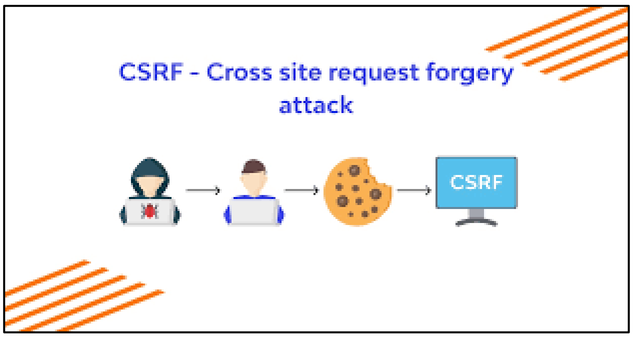
TASK-4: 10 Web Application Vulnerabilities (other than Top 10)

1. Directory indexing

When a user types in a request for a page on a web site, the web server processes the request, searches the web document root directory for the default file name, and then sends this page to the user. If the server cannot find the page, it will issue a directory listing and send the output in HTML format to the user.

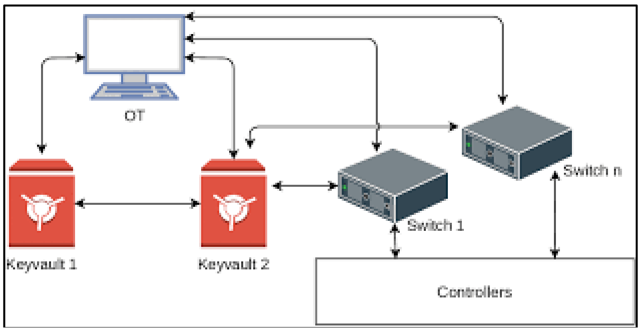
2. Cross-site request forgery (CSRF)

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. With the help of social engineering, an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.



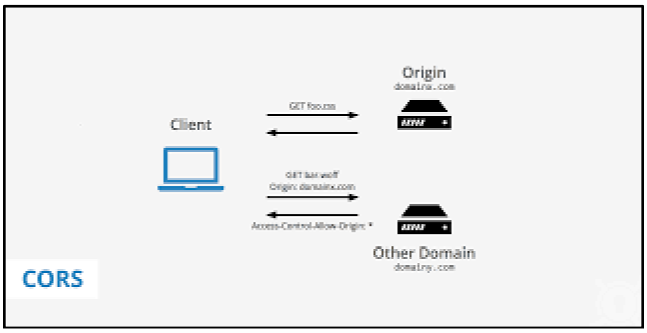
3. Credentials management vulnerability

Credential based attacks occur when attackers steal credentials to gain access, bypass an organizations security measure, and steal critical data. This attack is done on a device via controllers, keyvaults etc.



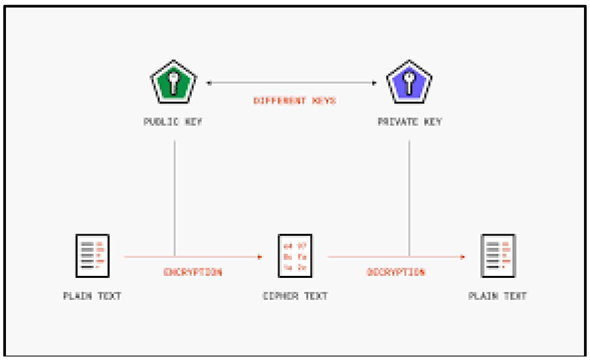
4. Cross-Origin Resource Sharing (CORS) Policy

Cross-origin resource sharing (CORS) is a mechanism for integrating applications. CORS defines a way for client web applications that are loaded in one domain to interact with resources in a different domain.



5. Cipher transformation insecure

Cipher transformation insecurity refers to vulnerabilities in cryptographic algorithms and their configurations, especially in the context of Java Cryptography Architecture (JCA) and Java Secure Socket Extension (JSSE). When implementing encryption or decryption in Java applications, developers often specify cipher transformations to determine the cryptographic algorithm, mode, and padding to be used.

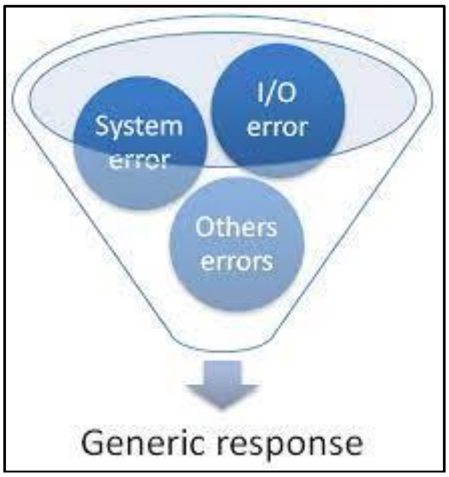


6. Encapsulation

Encapsulation refers to a programming approach that revolves around data and functions contained, or encapsulated, within a set of operating instructions. Applications become vulnerable to an attack when they fail to separate or differentiate critical data or functionality within components. When an encapsulation vulnerability exists, bad code creeps across software components or "leaks" from an application. This problem can also lead to cross-domain attacks. Without strong and clearly defined boundaries between control spheres, attackers can gain unauthorized access to data and functions.

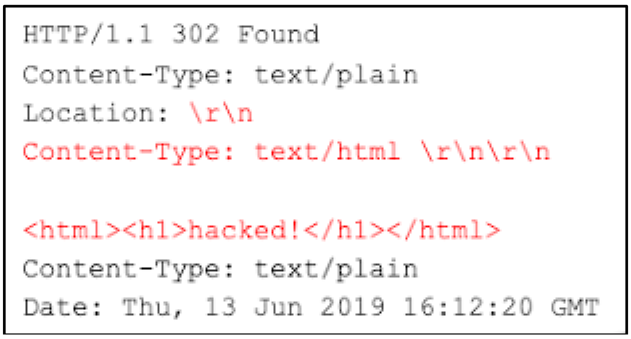
7. Error handling

Error handling is a crucial aspect of software development that deals with how an application handles unexpected situations or errors. Inadequate error handling can lead to security vulnerabilities as attackers might exploit error messages to gain insights into the system's architecture or sensitive information. Proper error handling involves providing informative but not overly revealing error messages and handling exceptions gracefully to prevent information leakage.



8. HTTP response splitting

HTTP response splitting is a web security vulnerability where an attacker inserts newline characters into an HTTP response. This can lead to malicious manipulation of the response, potentially allowing an attacker to inject their own content or conduct various forms of attacks. To prevent this vulnerability, developers should properly sanitize and validate user input before generating HTTP responses.

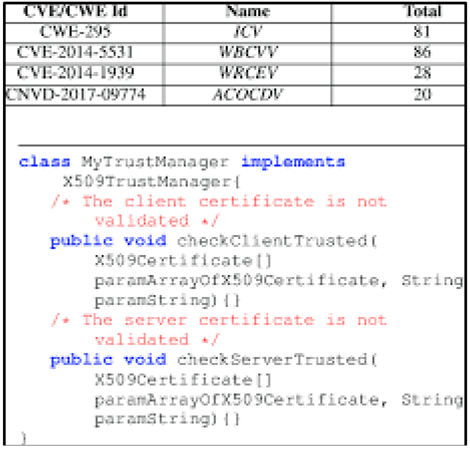


```
HTTP/1.1 302 Found
Content-Type: text/plain
Location: \r\n
Content-Type: text/html \r\n\r\n

<html><h1>hacked!</h1></html>
Content-Type: text/plain
Date: Thu, 13 Jun 2019 16:12:20 GMT
```

9. Improper certificate validation

Improper certificate validation occurs when an application fails to correctly validate SSL/TLS certificates during secure communications, such as HTTPS. This can allow attackers to intercept or manipulate data exchanged between the client and server. Developers should ensure that their applications validate certificates correctly, checking for validity, issuer trust, and hostname matching, to prevent this type of attack.

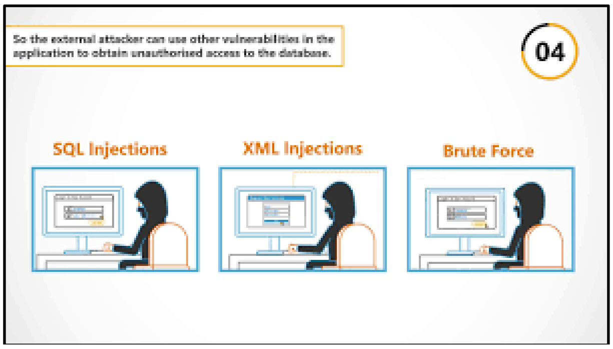


CVE/CWE Id	Name	Total
CWE-295	*ICV*	81
CVE-2014-5531	*WBCVV*	86
CVE-2014-1939	*WRCEV*	28
CNVD-2017-09774	*ACOCDV*	20

```
class MyTrustManager implements
    X509TrustManager{
   /* The client certificate is not
       validated */
   public void checkClientTrusted(
       X509Certificate[]
       paramArrayOfX509Certificate, String
       paramString){}
   /* The server certificate is not
       validated */
   public void checkServerTrusted(
       X509Certificate[]
       paramArrayOfX509Certificate, String
       paramString){}
}
```

10. Insecure cryptographic storage

Insecure cryptographic storage refers to the improper handling and storage of sensitive data like passwords or encryption keys. If these are stored in an insecure manner, such as using weak encryption algorithms, inadequate key management, or unhashed passwords, it can lead to data breaches and unauthorized access. Developers should follow industry best practices for secure storage and encryption to protect sensitive information.



So the external attacker can use other vulnerabilities in the application to obtain unauthorised access to the database.