23/08/23

**AI WITH CYBER SECURITY**

TASK-1

Top 10 cyber hackers

Why are they known so and what did they do ,what type of hat are they?

**1)Kevin Mitnick**

Once the 'Most Wanted Computer Outlaw,'

Best known for an audacious hacking spree in the 1990s involving the theft of data and credit card numbers, he later became a security consultant and public speaker.

He is a White hat hacker

**2)Gary Mckinnon**

who was accused in 2002 of perpetrating the "biggest military computer hack of all time", although McKinnon himself states that he was merely looking for evidence of [free energy suppression](https://en.wikipedia.org/wiki/Free_energy_suppression) and a [cover-up](https://en.wikipedia.org/wiki/Cover-up) of [UFO](https://en.wikipedia.org/wiki/UFO) activity and other technologies potentially useful to the public. On 16 October 2012, after a series of legal proceedings in Britain, then [Home Secretary](https://en.wikipedia.org/wiki/Home_Secretary) [Theresa May](https://en.wikipedia.org/wiki/Theresa_May) blocked [extradition](https://en.wikipedia.org/wiki/Extradition) to the United States.

He is a Black hat hacker

**3)Adrian lamo**

Security compromise

Lamo was a grey hat hacker who viewed the rise of the World Wide Web with a mixture of excitement and alarm. He felt that others failed to see the importance of internet security in the early days of the World Wide Web.

He is a Grey hat hacker

**4)Albert Gonzalez**

Albert Gonzalez is one of the many poster children for black hat hacking. In 2005, he organized a group of individuals to compromise poorly secured wireless networks and steal information.

He is a “Black hat” hackers

**5)Kevin Poulson**

Kevin Lee Poulsen (born November 30, 1965) is an American former black-hat hacker and a contributing editor at The Daily Beast.

He is a black-hat hacker

**6)Jonathan james**

American [hacker](https://en.wikipedia.org/wiki/Security_hacker) (a [gray hat](https://en.wikipedia.org/wiki/Gray_hat) [ethical hacker](https://en.wikipedia.org/wiki/Ethical_hacker)) who was the first juvenile incarcerated for [cybercrime](https://en.wikipedia.org/wiki/Cybercrime) in the United States.

He is a grey hat hacker

**7)Michael calce**

Mr Calce learned his lesson and now he is on the good side working as a white hat hacker, which means he helps companies identify weak spots in their technology and helps protect them. One of the key things he urges everyone to do is switch off functions many of us run.

He is a white hat hacker.

**8)Robert tappan morris**

The malware known as **Morris worm** was launched from a computer at Cornell University, by graduate student Robert Tappan Morris. The worm spread to all internet-connected computers and was designed to be undetectable. A design flaw led it to create more copies than Morris could control, which ultimately led to its detection.

He is a Black hat hacker

**9)Jeanson james Ancheta**

He became the first person to be charged for controlling large numbers of hijacked computers

He is a white hat hacker

**10)Julian Assange**

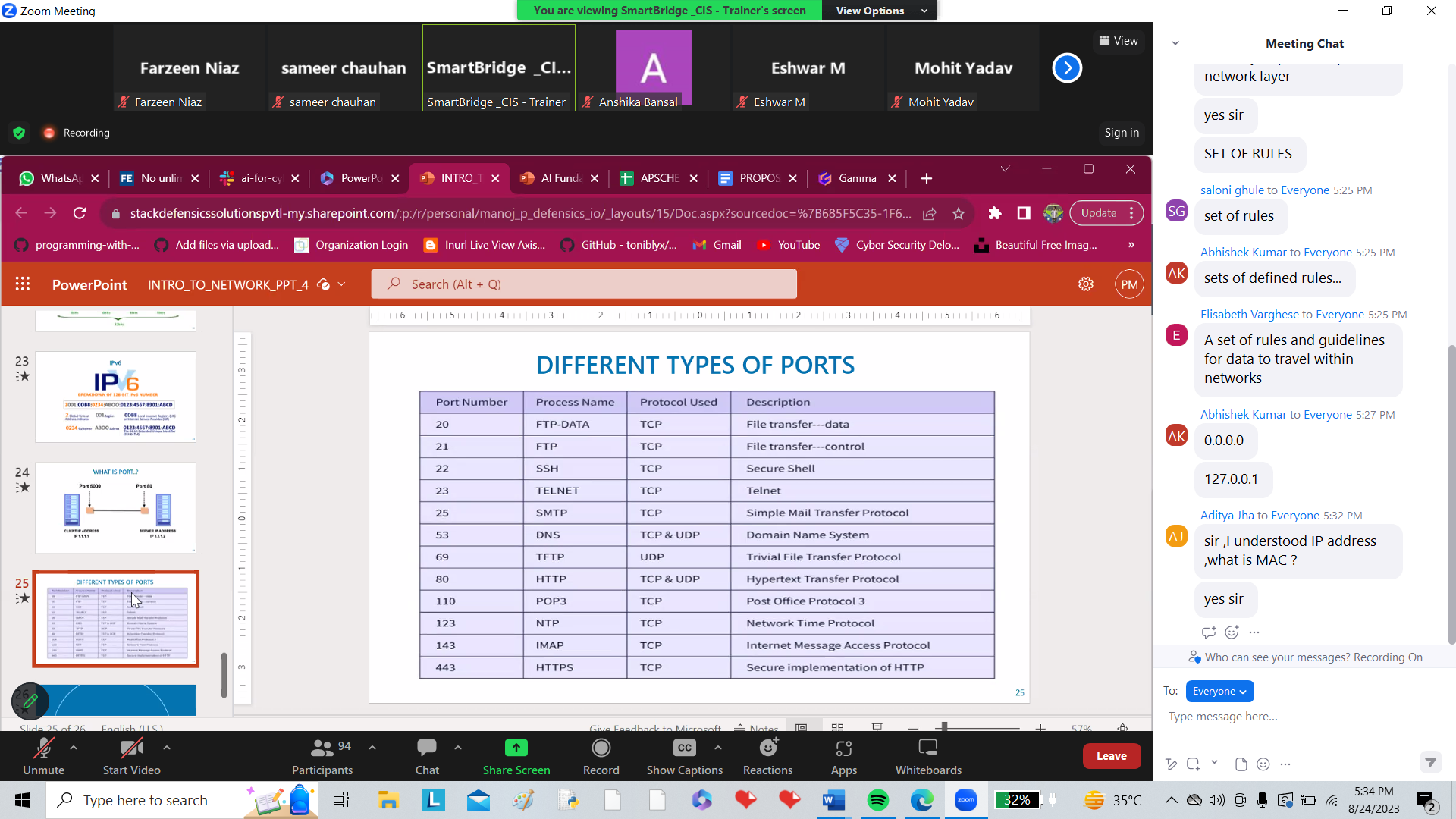
hat he hacked into Department of Defense computers in his twenties is not up for debate.

He is a White hat hacker

**Date:-24/08/23**

**Task-2**

**Port and vulnerabilities**



1)port number 20:-

If port number 20 is open the vulnerabilities will be

The most commonly abused ports are: FTP (Port 20 and 21):An insecure and outdated protocol, FTP doesn't have encryption for data transfer or authentication. Cybercriminals can easily exploit this port through cross-site scripting, brute-forcing passwords, and directory traversal attacks.

If port number 20 is open then the vulnerabilities we can perform are

* Cross-site scripting
* Brute-forcing passwords
* Directory traversal attacks
* Anonymous authentication. You can log into the FTP port with both username and password set to "anonymous".

[Port 20 and 21 are solely TCP ports used to allow users to send and to receive files from a server to their personal computers](https://www.makeuseof.com/vulnerable-ports-check-when-pentesting/)

**2)**[**Port 22 vulnerabilities include**](https://www.speedguide.net/port.php?port=22)

* [Tunneling random TCP traffic to other hosts on the network via Ruckus devices](https://www.speedguide.net/port.php?port=22).
* [Brute-force attacks on SSH servers with weak or default credentials](https://security.stackexchange.com/questions/226546/opened-ssh-22-port-cause-risks).
* [Enumeration of internal host services via Google Chrome](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-6082).

Port number 23 vulnerabilities

**Hardening the operating system** that is hosting the service, defining robust authentication policies, updating the system, and even using advanced systems such as SELinux among others

**3)Port number 25 vulnerabilities**

1. [Port 25 is a **Simple Mail Transfer Protocol (SMTP) port for receiving and sending emails1**](https://blog.netwrix.com/2022/08/04/open-port-vulnerabilities-list/). [Without proper configuration and protection, this TCP port is **vulnerable to spoofing and spamming1**](https://blog.netwrix.com/2022/08/04/open-port-vulnerabilities-list/). [If you intend to serve SMTP service, the open port 25 is a vulnerability since it exposes your service to external threats targeting this service**2**](https://security.stackexchange.com/questions/98419/does-having-these-ports-open-indicate-any-vulnerabilities). [An integer overflow in some web browsers allows remote attackers to bypass intended port restrictions on outbound TCP connections via a port number outside the range of the unsigned short data type, as demonstrated by a value of 65561 for TCP port 25](https://www.speedguide.net/port.php?port=25)

**4)Port number 53 vulnerabilities**

[**Port 53 vulnerabilities include1**](https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=port+53):

* Kerio Personal Firewall (KPF) 2.1.4 has a default rule to accept incoming packets from DNS (UDP port 53), which allows remote attackers to bypass the firewall filters via packets with a source port of 53.
* Avaya Argent Office allows remote attackers to cause a denial of service by sending UDP packets to port 53 with no payload.

[**Port 53 is also associated with Adobe Flash Player vulnerabilities2**](https://www.dshield.org/data/port.html). Adobe Flash Player before 10.2.154.27 on Windows, Mac OS X, Linux, and Solaris and 10.2.156.12 and earlier on Android; Adobe AIR before 2.6.19140; and Authplay.dll (aka AuthPlayLib.bundle) in Adobe Reader 9.x before 9.4.4 and 10.x through 10.0.1 on Windows, Adobe Reader 9.x before 9.4.4 and 10.x before 10.0.3 on Mac OS X, and Adobe Acrobat 9.x before 9.4.4 and 10.x before 10.0.3 on Windows and Mac OS X allow remote attackers to execute arbitrary code or cause a denial of service (application crash) via crafted Flash content

**5)Port number 69 vulnerabilities**

PORT 69 – Information  
  
Security Concerns: Remote attackers can download server files without auth. Can extend to sensitive files (eg: passwd file) if server is poorly configured. Since file transfer is cleartext, all boot info passed to clients is vulnerable.

**6)Port number 80 vulnerabilities**

Port 80 isn't inherently a security risk. However, if you leave it open and don't have the proper configurations in place, attackers can easily use it to access your systems and data. Unlike port 443 (HTTPS), port 80 is unencrypted, making it easy for cybercriminals to access, leak and tamper with sensitive data.

**7)Port number 110 vulnerabilities**

Port 110 is used by the POP3 protocol for unencrypted access to electronic mail. The port is intended for end-users to connect to a mail server to retrieve messages.

**8)Port number 123 vulnerabilities**

Network Time Protocol (NTP) - used for time synchronization [[RFC 5905](http://tools.ietf.org/html/rfc5905)]  
  
Security Concerns:  
It provides both information and possible avenue of attack for intruders. Info gathered can include system uptime, time since reset, time server pkt, I/O & memory statistics and ntp peer list. If a host is susceptible to time altering via ntp an attacker can possibly:  
1) Run replay attacks using captured OTP and Kerberos tickets before they expire.  
2) Stop security-related cron jobs from running or cause them to run at incorrect times.  
3) Make system and audit logs unreliable since time is alterable.  
  
Vodafone Sure Signal also uses this port

**9)Port number 143 vulnerabilities**

IMAP was designed with the goal of permitting complete management of an email box by multiple email clients, therefore clients generally leave messages on the server until the user explicitly deletes them. An IMAP server typically listens on port number 143.

**10)Port number 443 vulnerabilities**

What Are the Port 443 Vulnerabilities? Port 443 has the same exposure as the HTTPS and TLS protocols. Vulnerabilities can include the following: Man-in-the-middle (MITM) attacks, where a hacker intercepts the communication between the client and server to steal sensitive information

**25/08/23**

**Task -3**

**Ows top 10 from that select the top 5 n search cwe of each separately**

**Top 10 vulnerabilities we have to find n do research on it**

**We have to perform top 5 OWASP vulnerabilities’**

* [**A01:2021-Broken Access Control**](https://owasp.org/Top10/A01_2021-Broken_Access_Control/) moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.
* [**A02:2021-Cryptographic Failures**](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/) shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.
* [**A03:2021-Injection**](https://owasp.org/Top10/A03_2021-Injection/) slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.
* [**A04:2021-Insecure Design**](https://owasp.org/Top10/A04_2021-Insecure_Design/) is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to “move left” as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.
* [**A05:2021-Security Misconfiguration**](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/) moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it’s not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.
* [**A06:2021-Vulnerable and Outdated Components**](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/) was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.
* [**A07:2021-Identification and Authentication Failures**](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/) was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.
* [**A08:2021-Software and Data Integrity Failures**](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/) is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.
* [**A09:2021-Security Logging and Monitoring Failures**](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/) was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn’t well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.
* [**A10:2021-Server-Side Request Forgery**](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/) is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it’s not illustrated in the data at this time.

**28/08/23**

**Task-4**

**Understanding web application vulnerabilities**

### 1. Injection Attacks

An injection attack occurs when a hacker conducts an unvalidated input to a web application. Most often, injections target the app’s most vulnerable and insecure components. In particular, SQL (Structured Query Language) injection attacks are some of the most common and dangerous cyber threats.

SQL is a technology allowing you to establish communication and management of databases. When an attack occurs, SQL injection (SQLI) uses malicious code to manipulate backend databases and access sensitive or confidential information like credentials, user names, and passwords. The attackers can steal, remove, or change this data, which puts businesses at enormous risk.

**How to avoid injection attacks?**

* Add filters to input fields
* Use parametrized queries, such as prepared statements
* Avoid using inputs directly .

### 2. Data Leakage

According to an [IBM annual report](https://www.ibm.com/security/data-breach), the average data breach cost in 2022 is nearly $4.5 million. So every business should strive to mitigate information leakage risks. However, unfortunately, such attacks often occur with web applications.

Most commonly, attackers can steal sensitive data during its transition to the users or in case of insufficient storage encryption. Therefore, companies should do their best to keep data protected when it’s at rest and in transition.

**How to avoid sensitive information exposure?**

* Analyze website scripts during the development lifecycle
* Implement ciphers (encryption algorithms) for incoming data
* Turn off caching of confidential information
* Ensure reliable database encryption and keep the keys separately
* Remove outdated information

### 3. Security Misconfiguration

Misconfiguration occurs when a vulnerability in web applications is exceptionally high. Hence, it’s much easier for an attacker to approach your app’s security. The following situations put your website at the greatest risk:

1. Default passwords and accounts
2. Insecure passwords
3. Unpatched flaws
4. Insecure file and directory
5. Unused pages or other components
6. [Obsolete software](https://devoxsoftware.com/blog/application-modernization-strategy-full-guide/)

Security misconfiguration vulnerabilities are extremely dangerous since they may lead to attacks targeting the weakest points of your [web application or website](https://devoxsoftware.com/blog/web-app-vs-website-whats-the-difference/).

**How to avoid security misconfigurations?**

* Enhance web application’s architecture
* Don’t use default usernames and passwords
* Conduct access control verification
* Use a deployment protocol for updates

### 4. Broken Access Control

Access control issues are some of the most common web application security vulnerabilities ranked first in the latest OWASP report. This particular challenge was [the most critical](https://owasp.org/Top10/A01_2021-Broken_Access_Control/) in 2021.

The issue involves a failure in user-data or user-resource interactions. It means that unauthorized access to specific actions or information may occur. Normally, the users aren’t supposed to go beyond the interface and interact with the backend of your web application. However, if the broken access control takes place, hacker attacks or breaches are inevitable.

For example, imagine that your web app users can change payment and order details instead of just viewing them. Malicious actors will use this opportunity to manipulate or use this confidential information.

**How to avoid broken access control?**

* Keep the app’s authentification configured
* Manage role-based access control

### 5. Broken Authentification

Broken authentification is another common vulnerability leading to unwanted access to your web app’s internal system. Most often, it occurs when the hackers steal passwords or keys get permission to manipulate your data. For example, they can interfere with the functioning of your networks or software.

This vulnerability takes place due to several reasons, including incorrect password hashing algorithms, poor password timeout management, and the use of insecure passwords.

**How to avoid broken authentification?**

* Adopt multi-factor authentification
* Create secure passwords and regularly update them
* Configure password timeout length correctly

### 6. Cross-site Scripting

Cross-site scripting (XSS) is a cyberattack that involves inserting malicious scripts into the web app’s code to infect users’ browsers. Usually, hackers hide such scripts in links, images, or videos, so when a user opens the website, the malicious code installs into their browsers. Later, it can spread malware, redirect users to illegitimate websites, access their browsing history, or steal sensitive data.

The most vulnerable objects that an XSS attack may affect are your web app’s unsanitized input fields.

**How to avoid cross-site scripting?**

* Sanitize input fields
* Filter input and encode output data
* Use proper response header

### 7. Cross-site Request Forgery

Cross-site request forgery (CSRF) is a malicious attack applying social engineering techniques encouraging users to change their account information like username and password. If doing so, the user gets tricked by the attacker who takes control over the session and can access confidential data or even perform a financial transaction.

Web security vulnerability to CSRF attacks is often due to the lack of additional user authentification or random tokens. With their help, you can prevent malicious behavior even if a user has logged into the application.

**How to avoid cross-site request forgery?**

* Use random tokens and cookies
* Implement two-factor authentification or re-authentification

### 8. Unverified Redirects and Forwards

Web applications use redirects and forwards to transport users to another URL automatically. Usually, this technique helps users quickly access specific information and effortlessly navigate the website. However, hackers and malicious actors can use this method to redirect or forward users to illegitimate pages, steal data, or install malware on their devices.

Web application security vulnerabilities leading to such attacks are lack of identifiers for request parameters or too many destination URLs.

**How to avoid unverified redirects and forwards?**

* Minimize the number of redirects and forwards in your app
* Validate destination parameters

### 9. Insecure Direct Object References

Insecure Direct Object References (IDOR) is another web app vulnerability related to unwanted approaches to databases containing sensitive information. This risk occurs if your website’s URLs expose the format or pattern for entering your system directly. At the same time, there is no additional access control or authorization check. As a result, hackers can easily bypass authorization and enter the storage system.

This vulnerability exposes your files or keys that the attacker can change, remove or utilize to steal data.

**How to avoid insecure direct object reference?**

* Provide input validation on the server-side of your app
* Adopt additional access control checks
* Conduct consistent permission management

### 10. Remote File Inclusion (RFI)

In a web application, file inclusion refers to the “include” functionality the developers use to put data from one file to another. It’s a standard function for the PHP programming language. However, file inclusion is also a significant vulnerability in web applications. It allows hackers to access sensitive files, execute malicious code, or even perform cross-site scripting.

Remote file inclusion (RFI) is typical for websites or servers running PHP. The attacker aims to upload malware to the website server applications from a remote URL. Once the malicious script is uploaded, hackers can access and harm the web app’s resources.

**How to avoid remote file inclusion attacks?**

* Disable the remote inclusion function in your PHP configuration
* Whitelist and sanitize input fields

**28/08/2023**

**Task-5**

**10 web vulnerabilities apart from top 10**

## **11. Directory traversal**

Also called directory climbing, dot-dot-slash, and backtracking attack, the directory traversal method leverages the way in which an application gets data from the webserver. Generally, Access Control Lists (ACLs) limit user access to specific files within a root directory.

Consider a set of nested folders that follow this order:

* Root directory: My Very Sensitive Data (MVSD)
* Inside MVSD folder: Protecting from H@x0rs (PfH) folder
* Inside PfH folder: My Password is Bad (MPiB) folder
* Inside MPiB folder: H@x0rs Stole My Info file

Now, you might have an additional set of folders outside that root folder including Pictures, Videos, and Downloads. Unless you have access to each of these other root folders, you can’t access the information they contain.

Web applications organize information the same way, even if you don’t see it. In a directory traversal attack, malicious actors figure out the URL structure that the application uses to request files. Using the hypothetical above, that URL might be:

www.myinsecurewebapp.com/MyPas… “.asp?item=” indicates that this URL pulled the file “H@x0rsStoleMyInfo” from the “My Password is Bad” folder. Now, they know the structure of folders and how to start getting different files.

Using this structure, they add “../” at the end. The “../” indicates moving from one folder to one just above it in the hierarchy. The new request might look like this:

www.myinsecurewebapp.com/MyH@cking.asp?item=../

They keep adding the ../ until they gain access to another file. If they know the name of the file, such as an operating system file name, they might do this:

www.mywebsiteinfo.com/MyPasswordisBad.asp?item=../genericoperatingsystemfile

At this point, they just keep adding more “../” after the equal sign until they get to the folder level and file they want.

## **12. Encapsulation**

Unlike some of the other vulnerabilities that leverage web browser access to applications, encapsulation vulnerability exploits focus on weaknesses in the way a developer coded the application. The programming term encapsulation refers to bundling data and actions that can be taken on that data into a single unit. Encapsulation protects data by hiding details about how the code works which creates a better user interface. Users don’t need to know how the application brings them data; they just need access to it.

For example, a developer can bundle access controls, like read/write permissions, into an application’s ability to retrieve data. When the user requests information in the application, it returns only the data that they have permission to access.

However, if the developers fail to clearly define the boundaries between the data and the actions taken across different areas of the application, the application has an encapsulation vulnerability. Attackers exploit this by sending the application a request that they know will result in an error message. The error message gives them information about how the application works, enabling additional attack types such as a denial of service.

## **13. Error handling**

Several different attack methods rely on how an application responds to abnormal inputs or conditions. One example of an error message is the “404 not found” message when you try to access a website. For most enterprise applications and systems, error messages provide valuable information about how to fix a problem.

However, for web applications, too much information returned through an error message can give malicious actors that same information. Often, attackers send the web application a query that they know will return an error message. They usually do this during the reconnaissance phase, where they try to get as much information as possible so they can find exploitable vulnerabilities.

## **14. Failure to restrict URL access**

As with many other web application vulnerabilities, this one also aligns with access control rights. Applications use URL restrictions to prevent non-privileged users from accessing privileged data and resources. Every clickable button in a web application directs to a URL. A failure to restrict access vulnerability means that while clicking the button in the application would prevent access, directly using the URL into the browser allows access. When an application fails to restrict URL access, malicious actors can use “forced browsing” for an attack.

For example, a web application might have a URL structure that looks like this:

www.insecurewebapp.com/failure… the attackers know that the last item in that URL is the data type, they can try to take guesses at the URL structure for a specific type of sensitive information.

www.insecurewebapp.com/failure… the application has a failure to restrict URL access vulnerability, plugging that URL directly into the browser gives the attacker access.

## **15. HTTP response splitting**

HTTP response splitting is a type of CRLF injection attack. HTTP is the way that a browser sends queries and a server sends back responses. In an HTTP response splitting attack, the malicious actors use the CR and LF notations to manipulate how the browser and server “talk” to one another that sends a request but asks the server to “split” the response into different parts. Splitting the response into two parts gives the attacker control over what data the server sends in response to the second part of the request. When that requested data is sensitive or user ID data, the malicious attacker has completed the attack.

## **16. HTTP verb tampering**

HTTP is the protocol that lets applications respond to requests and retrieve data. An HTTP verb is one of several actions that the application can use when querying the server. Common ones HTTP verbs include:

* GET: retrieves data from specified source
* HEAD: requests preview of specified resource
* POST: submits entity to specified resource, such as editing data
* PUT: transmits new data to the specified resource replacing the old information
* DELETE: deletes the specified resource entirely

Most web applications use HTTP verbs to authenticate users and manage access privileges. Malicious actors can bypass authentication and access controls intended to protect privileged information.

## **17. Improper certificate validation**

SSL certificates bind a domain name, server name, or hostname to a company and location. For example, GoodSecureCo installs the SSL certificate data files on its US web servers. Every time a browser asks for data from the US web server, the SSL certificate checks to make sure that the user’s browser connects with an approved owner. The two securely connect if the answer is yes.

When software refuses to validate or incorrectly validates the certificate, it has an improper certificate validation vulnerability. Most often, attackers create a false trusted entity that tricks the server or application into thinking the certificate is valid so it accepts the data transfer as legitimate. Often, malicious actors use improper certificate validation vulnerabilities as a way to install malware on endpoints.

## **18. Injection flaw**

An injection flaw enables a variety of different attack methods. Any application that enables users to update a database, shell command, or operating system call can have an injection flaw. In computing, an interpreter is a program that takes a command, generates an instruction, and performs the action within the application.

Malicious actors use injection flaws to change the commands which leads to new and unintended actions within the application. Leveraging these flaws, attackers can create, read, update, or delete data.

## **19. Insecure cryptographic storage**

Encrypting stored data is a common best practice for preventing unauthorized access to or use of sensitive information. Encryption takes information stored in a readable format, such as PlainText, then uses mathematical algorithms to scramble it, making it unreadable. Encryption typically requires an encryption key, which is the technology that applies the algorithm that scrambles the data and is also used to make the information readable again. However, if someone finds the encryption key, the protection no longer works.

The insecure cryptographic storage vulnerability means you have a problem with one or more of the following:

* Not encrypting all sensitive data
* Improper key storage and management
* Easy to crack encryption algorithms
* Internally-designed, untested algorithm

## **20. Insecure deserialization**

Applications handle complex data structures. Serialization converts the structures into an object that can be stored and transmitted easily. For example, think about different actions that go into making a peanut butter and jelly sandwich:

1. Get plate
2. Get bread
3. Open bread
4. Take out bread 1
5. Put bread 1 on plate
6. Take out bread 2
7. Put bread 2 on plate
8. Get knife
9. Get peanut butter
10. Open peanut butter
11. Get jelly
12. Open jelly
13. Get peanut butter on knife
14. Put peanut butter on bread 1
15. Get jelly on knife
16. Put jelly on bread 2
17. Smoosh bread 1 and bread 2 together with covered sides facing

You need all of these things to happen as part of making the sandwich, but they aren’t necessarily step-by-step in this order. Having to send all 17 of these data points, like individual messages, every time someone asks for a peanut butter and jelly sandwich can be time-consuming to write down and send. Most likely, you’d group them in a document as “Peanut Butter and Jelly Sandwich” that you send when someone asks, similar to serialization. When the person opens the document, they can see each individual data point, similar to deserialization.

Deserialization is the process of reconstructing the original, expanded data structure. With a deserialization vulnerability, malicious actors can change the application logic or execute code remotely, one of the most serious attack types.

**29/08/23**

**Task-6**

**10 web server attacks**

### 1)URL INTERPRETATION ATTACK

This attack is also called URL poisoning as the attackers manipulates the URL by changing its semantics but keeping the syntax intact. The parameters of the URL are adjusted so that information beyond what is intended can be retrieved from the web server. This type of attack is quite common with CGI-based websites.

To understand how this type of attack is perpetrated, we can take an example of an email application where the users can reset their email password by answering a security question. After the security question is answered correctly, the application opens a page where the user can set an alternative email address. The page that receives the request lets the users reset the password and has all the login credentials of the user. Using URL interpretation, the URL that carries the request to fetch the details of a user and send it to the alternative email id provided, can be modified to fetch the detail of another user. Thus, URL interpretation attack makes the information of other users vulnerable.

URL interpretation attack can be prevented by implementing a fix that is usually supplied by a vendor and also through in-depth checking and verification of the web server configuration.

### 2)SQL INJECTION ATTACK

As the name suggests, SQL injection attack aims to modify a database or extract information from it. An SQL query with parameters from the URL is fed to the database that has the ability to alter the data. The stored procedures in the database can also be executed through SQL injection and database can be made to do things, it is intended to do only when desired by the authorized personnel.

When this attack is conducted, there are chances of backend database server to be compromised and it can be catastrophic for a company. The vulnerability that this type of attack exploits is the scenario where the SQL query is permitted to be executed without validating the input data. Websites that are most likely to be attacked using this type of attack are e-commerce websites that have huge database comprising users™ information.

SQL injection or SQL poisoning is an attack that also does not have an easy fix and it requires a thorough review of the source code, following least privilege for DB applications and deleting redundant and unnecessary database users and procedures.

### 3)INPUT VALIDATION ATTACK

Input validation attack is an attack on the web server where the server executes a code injected by a hacker to the web server or the database server. There are many input types that need to be validated before execution including data type, data ranges, and others. By executing the code with inputs that are not validated, information can be retrieved or modified by the attacker.

This attack is done by bypassing the client side checking using JavaScript code. Tampering of the hidden file is also possible with this attack. Data to the web server may be sent in a variety of ways including the URL, HTTP headers, POST requests, and cookies. Any negligence while writing the code and trusting the data received from the users can lead to such attacks.

When it comes to input validation attack, there are really no countermeasures. The only preventive measure is following a good coding practice. The code should have the provision for validating all the inputs like data types, data ranges, meta characters and buffer sizes.

### 4)BUFFER OVERFLOW ATTACKS

Buffer Overflow attack implies the deliberate overflowing of the buffer memory that is reserved for the users™ input. When an application awaits a users™ input, it allocates a stack with a memory location where the input data by the user is fed. The attackers flood this space by writing arbitrary data so that the memory stack is full and the users deny the service. This is one of the ways to perform denial of service attack which is dealt with in more detail further.

Another aspect of this type of attack is that the hacker can feed an executable command in the stack. Although, the surety of the execution of the command is dependent on the return address that is specified by the hacker. After the stack recovers from the crash, it goes to the return address and if it has been changed and replaced with one that falls within the desired range, the command may execute and grant entry to certain sections of the web server.

In the case of buffer overflow attacks, the best way to mitigate is the vendor supplied specific fixes. However, checking the bounds within the application can be effective. Buffer overflow testing and source code review are often regarded as good countermeasures.

### 5)IMPERSONATION ATTACKS

Impersonation attack is also called IP spoofing where the hacker pretends to be accessing the web server with an IP that is actually impersonating an IP that has the access to the web server. There are special programs that the hackers make use of, to create an IP packet that appears to be originating from the intranet and hence gain entry to the section of the web server that is intended to be accessed only by the authorized personnel.

Impersonation attacks exploit the vulnerability of the authentication protocols and get unauthorized access to the web servers and the databases. Such attacks can be countered by having a strong authentication module and identification of the traffic coming towards the server.

Countermeasures for impersonation attacks are locking down of web configurations. Also there should be a firewall that tracks the source of IP from where the request is being directed to the web server. In case the impersonation is being done using cookie, obfuscating them is a good solution so that manipulating the cookies is not possible.

### 6)PASSWORD-BASED ATTACKS

The authentication system of a web server is often based on the password that identifies a valid user and grants access to the web server. If the hacker can, by any means, get your username and password, he or she can access the information that only you are supposed to access. The older applications do not have strong authentication system and this makes it easy for the eavesdroppers to get through the authentication process.

Breaking a password is not easy and there are certain algorithms that the hackers deploy to guess the password and gain access to the network. Once hackers get access to the network, they can modify the configuration to make it easy for them to hack the network again if normalcy is restored in the website by attempts from the network engineers.

Preventing password based attacks is keeping the password long and complex and also have additional security measures to protect the database that stores the password. Cryptographic storage of password is also highly recommended.

### 7)DENIAL OF SERVICE ATTACKS

Denial of service attack (DOS) is an attack where the server denies serving the users with a response to their request. This attack is performed by several means and buffer flow is one of them. It is an effective and naturally one of the most popular ways of attacking the web server. The attackers after gaining access to the network randomize the attention of the security system experts so that they do not become aware of the attack immediately so that they can exploit the web server in other ways.

DOS attacks are performed by overwhelming the web server in numerous ways including sending invalid data as input that causes application termination, flooding the web server with automated request causing a crash, blocking the traffic resulting in loss of access to the users. DOS attacks are categorized under volume attack, protocol attack, and Application layer attack.

Prevention of DOS attacks from anonymous sources can be ensures by implementing a web server firewall that inspects the entire HTTL traffic and stop any data packet that appears malicious and generating from a source that is not authorized. Network audit trail must be maintained so that the changes done over a period of time can be tracked with ease. The network must also be tested locally as well as on the internet.

### 8)BRUTE FORCE

Brute Force, as the name suggests, implies cracking the username, password combination by using all possible iterations. This is a basic form of web server attack and is implemented when the hackers have a clue that weak passwords have been used in the authentication. The chance of brute force working is maximum when no other security measures are there besides password authentication.

The trick to prevent this type of attack is to create passwords that are long and comprise complex characters not used commonly. Also, there should be provision in the network to limit the number of attempts of unsuccessful login. After a certain number of unsuccessful attempts of login, the account may be locked. However, this is not seen as a practical solution. CAPTCHA can also be used in order to add an extra layer of security to prevent brute force attack.

### 9)SOURCE CODE DISCLOSURE

Through Source code disclosure attack, the attackers are able to retrieve the application files without using any parsing. The source code of the application is recovered and then it is analyzed to find loopholes that can be used to attack the web servers. It is often caused when the application is designed poorly or there are errors in the configuration.

Source code disclosure is said to be taken place when the attacker is able to access the source code of the server-side scripting language such as PHP or ASP. Net. These codes are not meant to be seen by any person other than the authorized programmers.

Attack through implementing source code disclosure attack can be nipped in the bud by conducting a thorough check on the web server proxy configuration. Also, care should be exercised while creating URL mappings to the internal servers.

### 10)SESSION HIJACKING

HTTP is a stateless protocol while all the web applications have states. When the tracking of these states is based on poor mechanism, session hijacking becomes easy for the hackers. It is also called cookie hijacking because a web server determines the session with a user based on the cookie. The cookie stored on the users™ computer is stolen by the hijacker by either intercepting it through the access to the network or through a previously saved cookie. Sniffing programs are used to perform this attack in an automated manner.

Preventive measures to tackle session hijacking attack are using server side tracking id, matching every connection with time stamps and associated IP address. Sessions Ids if generated cryptographically will be tough to decipher. Use of server session management API is also quite useful in the prevention of session hijacking attacks.

### 11)IMPACT OF WEB SERVER ATTACKS

The impact of these attacks can range from website defacement to information theft. Intrusion in the web server may have some other serious implications like the modification of the data and especially, users™ information. All this might lead to bad name for a company and the customers losing faith in it. Prospective users will be scared of sharing personal data with the brand again considering their inability to keep the data confidential in the past.

Attacks such as source code disclosure can be catastrophic for the websites as the source code sometimes contains login names and password along with the business logic that is devised after a lot of brainstorming and expenditure of resources. Denial of Service (DOS) attacks also have serious implications on the website. The credibility of the website is marred and users detest such websites because of their bitter experience of denial.

Not all attacks are for theft of information, but for the defacement of the website. Hackers do it to disrupt the normal operation of the website and driving the users away from it by exposing its vulnerability.

### 12)CLOSING THOUGHTS

In order to safeguard the information that is stored on the web server, these web server attacks need to be prevented at all costs. As they say, complete information is the precursor to prevention, almost all types of web server attacks that threaten the web security have been mentioned and the methods of prevention of these attacks have also been discussed in detail. These points will go a long way in setting up a web server that is secured from all kinds of attacks.