WEB SECURITY

UNIT -2

2.1 Privacy Protection Techniques:-

There are several privacy-protecting techniques and best practices you can consider:

Encryption: Use strong encryption for data at rest and data in transit to prevent unauthorized access.Anonymization: Remove or obfuscate personally identifiable information (PII) from datasets to protect user identities.

Data Minimization: Collect and store only the data necessary for a specific purpose to reduce the risk of data breaches.

Access Control: Implement strict access controls to limit who can access sensitive data and what they can do with it.

Tokenization: Replace sensitive data with tokens that are meaningless without the proper decryption key.

Privacy by Design: Incorporate privacy considerations into the design of systems and applications from the beginning.

User Consent: Obtain clear and informed consent from users before collecting and using their data.

Regular Audits: Conduct regular privacy audits and assessments to identify and rectify potential vulnerabilities.

Data Retention Policies: Define and adhere to data retention policies to ensure data is not kept longer than necessary.

Two-Factor Authentication (2FA): Require additional authentication steps beyond a password to access sensitive information.

Secure APIs: Protect APIs that handle user data with proper authentication and authorization mechanisms.

Privacy Enhancing Technologies (PETs): Explore technologies like differential privacy and federated learning to protect user privacy in data analysis.

Incident Response Plan: Have a well-defined plan in place for responding to data breaches and privacy incidents.

Regular Updates: Keep software and systems up-to-date with security patches to mitigate vulnerabilities.

Employee Training: Educate employees on privacy policies and best practices to prevent accidental data leaks.

Data Portability: Allow users to easily access and transfer their own data between services.

Privacy Policies: Clearly communicate your privacy practices to users through easily accessible privacy policies.

Third-Party Audits: Vet third-party vendors for their privacy and security practices, especially if they handle user data.

Secure Disposal: Properly dispose of data that is no longer needed, ensuring it cannot be easily recovered.

Legal Compliance: Stay informed about relevant data protection laws and regulations (e.g., GDPR, CCPA) and ensure compliance.

2.2 Backup and Anti-Theft:-

Backups and anti-theft measures are important components of web security:

Backups: Regularly backing up your website's data and files is crucial. This helps in case of data loss due to hacking, server crashes, or accidental deletions. You should have both on-site and off-site backups, and they should be tested to ensure they can be restored when needed.

Anti-theft: Anti-theft measures primarily apply to mobile devices and laptops. These measures include:

Device Locking: Enable password, PIN, or biometric locks on devices to prevent unauthorized access.

Find My Device: Services like Apple's "Find My iPhone" or Android's "Find My Device" help locate and remotely wipe your device if it's lost or stolen.

Remote Wipe: In case of theft, you can remotely erase all data on the device to prevent unauthorized access to sensitive information.

Tracking: Some anti-theft solutions allow you to track the device's location in real-time, aiding in recovery.

Encryption: Encrypting the data on your device makes it much harder for thieves to access your information.

In the context of web security, it's important to protect not only your website but also the devices you use to access and manage it. Regularly updating software and using strong, unique passwords for all accounts are additional best practices to enhance security.

To ensure the security of your data and protect your device from theft, here are some steps you can take:

Backup Your Data: Regularly back up your device's data to a secure location, such as cloud storage (e.g., Google Drive, iCloud) or an external hard drive. This ensures you can recover your data if your device is lost or stolen.

Enable Device Encryption: Encrypt your device's storage to prevent unauthorized access to your data. This feature is often available in device settings under "Security" or "Privacy."

Set Up a Strong Lock Screen: Use a secure PIN, password, or biometric authentication (fingerprint or face recognition) to lock your device. This helps prevent unauthorized access.

Install an Anti-Theft App: Consider installing a reputable anti-theft app that can help you track your device's location, remotely lock or wipe it, and even take photos of potential thieves if they try to unlock your device.

Activate Find My Device (Android) or Find My iPhone (iOS): These built-in features allow you to locate, lock, or erase your device remotely. Make sure they are enabled and linked to your account.

Use Two-Factor Authentication (2FA): Enable 2FA for your accounts, especially on apps and services that contain sensitive information. This adds an extra layer of security.

Be Cautious with Public Wi-Fi: Avoid connecting to unsecured public Wi-Fi networks, as they can be vulnerable to data theft. Use a VPN for added protection when necessary.

Keep an Eye on Your Device: Be mindful of your surroundings and don't leave your device unattended in public places.

Register Your Device: Keep a record of your device's serial number (IMEI for smartphones) and other identifying information. This can be helpful for reporting a stolen device to the authorities.

Report a Theft: If your device is stolen, report it to the police and provide them with all relevant information. Also, contact your service provider to report the theft and potentially lock the device.

By following these steps, you can enhance the security of your device and data, as well as improve your chances of recovering a lost or stolen device.

2.3 Backup Types:

Local Backup:

This involves creating a copy of your data on a physical storage device like an external hard drive or a computer. It's a good option for those who want complete control over their backups.

Cloud Backup:

Cloud services like Google Drive, iCloud, Dropbox, and OneDrive offer the ability to automatically back up your data to remote servers. This ensures your data is accessible from anywhere with an internet connection.

Manual Backup:

You can manually copy and paste important files and folders to a backup location, such as an external drive. This gives you control over what gets backed up.

Automated Backup:

Many operating systems and backup apps provide automated backup solutions that run at specified intervals. They back up your data without requiring manual intervention.

2.4 Anti-Theft Measures:

Remote Tracking and Locking: Anti-theft apps and built-in features like "Find My Device" (Android) and "Find My iPhone" (iOS) allow you to track your device's location and remotely lock it if it's lost or stolen.

Remote Data Wipe: You can remotely erase all data on your device to prevent unauthorized access to your personal information. This feature is available through anti-theft apps and platform-specific services.

Alarm and Siren: Some anti-theft apps enable you to trigger loud alarms or sirens on your device to draw attention to it in case it's lost nearby.

Photographic Evidence: Certain anti-theft apps can capture photos or videos using your device's camera to identify potential thieves.

SIM Card Lock: Lock your device to the SIM card. If someone tries to insert a different SIM card, the device may become inoperable or send alerts.

Biometric Locks: Enable biometric authentication methods like fingerprint or face recognition to secure your device, making it difficult for unauthorized users to access it.

Two-Factor Authentication (2FA): Enable 2FA for your accounts and devices to add an extra layer of security.

IMEI Blocking: Report your device's International Mobile Equipment Identity (IMEI) to your mobile carrier if it's stolen. They can block the device from accessing their network.

Lock Screen Messages: Display a message with contact information on your device's lock screen, so anyone who finds it can contact you.

2.5 Web Server Security

Web server security is crucial to protect your website and data from various threats. Here are some key practices:

Keep Software Updated: Regularly update your web server software, web applications, and plugins to patch known vulnerabilities.

Use Strong Authentication: Implement strong passwords and consider two-factor authentication for server access.

Firewall: Use a firewall to control incoming and outgoing traffic, allowing only necessary services to communicate.

HTTPS: Use SSL/TLS certificates to encrypt data in transit, ensuring secure communication between the server and clients.

Regular Backups: Back up your website and server data regularly, and store backups in a secure offsite location.

Access Control: Restrict access to directories and files, granting only necessary permissions to users and services.

Intrusion Detection and Prevention: Employ intrusion detection and prevention systems to monitor for suspicious activity and block potential threats.

DDoS Mitigation: Implement DDoS protection to prevent or mitigate Distributed Denial of Service attacks.

Web Application Firewall (WAF): Use a WAF to filter and monitor incoming web traffic for malicious activity and common web application vulnerabilities.

Security Headers: Set up security headers in your web server configuration to enhance security, like Content Security Policy (CSP) and HTTP Strict Transport Security (HSTS).

Regular Security Audits: Conduct security audits and vulnerability assessments periodically to identify and address weaknesses.

Error Handling: Customize error messages to reveal minimal information to potential attackers.

File Uploads: If your site allows file uploads, validate and sanitize user input and use a secure storage location for uploaded files.

Monitoring and Logging: Implement robust logging and monitoring to detect and respond to security incidents promptly.

User Education: Educate users and administrators about security best practices and the importance of strong password management.

Remove Unused Services: Disable or remove unnecessary services and features to reduce the attack surface.

Security Patch Management: Stay informed about security vulnerabilities and apply patches promptly.

Security Headers: Use security headers, like Content Security Policy (CSP) and X-Content-Type-Options, to enhance protection against common web vulnerabilities.

Network Segmentation: Segment your network to limit access to sensitive data and services.

Incident Response Plan: Develop an incident response plan to handle security breaches effectively.

2.6 Physical security for servers

Physical security for servers is crucial to protect sensitive data and ensure the uninterrupted operation of your IT infrastructure. Here are some key measures:

Access Control: Limit physical access to authorized personnel only. Use secure locks, card readers, or biometric scanners to control entry to server rooms or data centers.

Surveillance: Install security cameras to monitor server areas, and retain footage for review if necessary.

Environmental Controls: Maintain appropriate temperature, humidity, and fire suppression systems to safeguard servers from environmental hazards.

Rack Security: Secure servers within racks using locking mechanisms to prevent tampering or theft.

Alarms and Sensors: Employ intrusion detection systems and sensors to alert personnel of unauthorized access or environmental issues.

Visitor Logs: Maintain logs of all personnel entering server rooms, including the purpose of their visit and the duration.

Biometric Access: Implement biometric authentication for highly sensitive areas, such as fingerprint or retina scans.

Redundancy: Have backup power sources (e.g., uninterruptible power supplies) to ensure servers stay operational during power outages.

Cable Management: Keep cables organized and labeled to facilitate quick identification and maintenance.

Physical Location: Choose server room locations with security in mind, away from exterior walls or windows.

Security Policies: Develop and enforce strict security policies for server access, with clear guidelines and consequences for violations.

Regular Audits: Conduct periodic security audits and assessments to identify vulnerabilities and address them promptly.

Employee Training: Train staff on security procedures and the importance of physical security.

Vendor Access: If third-party vendors require access to your servers, closely monitor and restrict their access as needed.

Data Encryption: Encrypt data on servers to protect it in case of unauthorized physical access.

Remote Monitoring: Implement remote monitoring and management tools to reduce the need for physical access to servers.

2.7 Host Security for servers

Server security is a key aspect of server management for web hosting providers and server administrators. Here, we look at ten techniques for hardening servers and monitoring them for security vulnerabilities.

1. Use Public Key Authentication For SSH

Remove unencrypted access. No one should use telnet, ftp or http to manage servers anymore. SSH, SFTP and https are the accepted standards. For even better security, get rid of password authentication on SSH altogether. Instead, use SSH keys. Each user has a public key and a private key. The private key is kept by the user. The public key is kept on the server. When the user tries to login, SSH makes sure the public key matches the private key. Once password logins are disabled, there's no risk of a successful brute force attack against a weak password.

2. Strong Passwords

A security hardened server is a challenge for criminals, but you would be surprised how many server administrators leave the front door wide open. People—including those who should know better—tend to choose easily guessed passwords. Last year, brute force attacks against servers with weak SSH passwords resulted in a spate of ransomware attacks. Use long and random passwords—long passphrases are better and finally restrict users with login type access.

3. Install And Configure The CSF Firewall

The Config Server Firewall is a feature-rich, free firewall that can protect a server against a wide variety of attacks. Its features include stateful packet inspection, authentication failure rate limiting, flood protection, directory watching, and the use of external block lists. CSF is a fantastic tool, and is a lot easier to manage than iptables.

4. Install And Configure Fail2Ban

Every server on the web is plagued by bots looking for weaknesses. Fail2Ban trawls through your server's logs in search of patterns that indicate malicious connections, such as too many failed authentication attempts or too many connections from the same IP. It can then block connections from those IPs and notify an administrator account.

5. Install Malware Scanning Software

Ideally, you want to keep malicious individuals out of your server, but if they do manage to breach the server's security, you want to know about it as soon as possible. ClamAV is an excellent malware scanning tool for Linux, and rkhunter is useful for finding rootkits. In combination, there's a good chance they will find any malware a hacker might install on a server. AIDE can be used to generate a hashed table of files on the system and then compare the hash count of the files daily to confirm no changes have been made to system-critical files.

6. Keep Software Up-To-Date

Out-of-date software is likely to contain security vulnerabilities that are known to hackers, as Equifax recently discovered to everyone's cost. If you ignore all the other advice in this article —which you should not—you should at the very least update using your Linux distribution's package manager.

7. Backup Regularly

You may not think of backups as a security measure, but the main reason we secure a server is to keep the data stored on it safe. It's impossible to guarantee that a server will never be compromised, so data should be encrypted and backed-up to a offsite location. Regular testing of recovery from comprehensive backups will neuter ransomware attacks.

8. Monitor Logs

Logs are a vitally important security tool. A server collects enormous amounts of information about what it does and who connects to it. Patterns in that data often reveal malicious behavior or security compromises. Logwatch is an excellent daily summary tool that can analyze, summarize, and generate reports about what's happening on your server. Logsentry can be used for hourly reports for more active monitoring of ingress.

9. Turn Off Unnecessary Services

Any internet-facing software that isn't essential to the server's function should be disabled. The fewer points of contact between the server's internal environment and the outside world, the better. Most Linux distributions—including CentOS and Ubuntu—include a tool for managing services.

This also applies to the web server engine itself, turn of modules you don't need, remove language modules not in use, disable web server status, and debugging pages. The less information you provide about your underlying infrastructure the smaller the footprint becomes to attack you with.

10. Install ModSecurity

Mod Security is a Web Application Firewall—it operates at a higher level than the CSF firewall and is designed to deal with threats against the application layer. In a nutshell, it stops many types of attack against web applications, including content management systems like Word Press and eCommerce stores like Magento. ModSecurity used to be an Apache module, but it is now available for NGINX too.

Securing Web Applications:-

web applications are a critical aspect of business and everyday life. By using web applications, both businesses and individuals can simplify and get more things done with fewer resources, achieving objectives much faster than they could before.

- They no longer need a warehouse full of meticulously organized paperwork.
- There is little or no need to rely on actual physical mail now for communication.
- Most marketing efforts are now highly web-focused.
- Even customer service is now pointing you to websites instead of 1-800 phone numbers.

Following are the tips developers should remember to protect and secure information:

1. Maintain Security During Web App Development

Before you run out and hire a team of security consultants, realize that you can maintain security in your web applications during the actual development of those tools.

2. Be Paranoid: Require Injection & Input Validation (User Input Is Not Your Friend)

A good rule of thumb is to consider all input to be hostile until proven otherwise. Input validation is done so that only properly-formed data passes through the workflow in a web application. This prevents bad or possibly corrupted data from being processed and possibly triggering the malfunction of downstream components.

Some types of input validation are as follows:

- Data type validation (ensures that parameters are of the correct type: numeric, text, et cetera).
- Data format validation (ensures data meets the proper format guidelines for schemas such as JSON or XML).
- Data value validation (ensures parameters meet expectations for accepted value ranges or lengths).

There is a whole lot more to input validation and injection prevention, however, the basic thing to keep in mind is that you want to validate inputs with both a syntactical as well as a semantic approach. Syntactic validation should enforce correct syntax of information (SSN, birth date, currency or whole numbers) while semantic validation should enforce the correctness of their values within a very specific business context (end date is greater than the start date, low price is less than high price).

3. Encrypt your data

Encryption is the basic process of encoding information to protect it from anyone who is not authorized to access it. Encryption itself does not prevent interference in transmit of the data but obfuscates the intelligible content to those who are not authorized to access it.

Not only is encryption the most common form of protecting sensitive information across transit, but it can also be used to secure data “at rest” such as information that is stored in databases or other storage devices.

When using Web Services and APIs you should not only implement an authentication plan for entities accessing them, but the data across those services should be encrypted in some fashion. An open, unsecured web service is a hacker’s best friend (and they have shown increasingly smarter algorithms that can find these services rather painlessly).

4. Use Exception Management

Another development-focused security measure is proper exception management. You would never want to display anything more than just a generic error message in case of a failure. Including the actual system messages verbatim does not do the end-user any good, and instead works as valuable clues for potentially threatening entities.

When developing, consider that there are generally only three possible outcomes from a security standpoint:

1. Allow the operation.
2. Reject the operation.
3. Handle an exception.

Usually, in the case of an exception or error, you will revert to rejecting the operation. An application that fails securely will prevent operations from unintentionally being allowed. For example, if an ATM failed you would prefer it to display a simple, friendly message to the user (not spill money out onto the ground).

5. Apply Authentication, Role Management & Access Control

Implementing effective account management practices such as strong password enforcement, secure password recovery mechanisms and multi-factor authentication are some strong steps to take when building a web application. You can even force re-authentication for users when accessing more sensitive features.

When designing a web application, one very basic goal should be to give each and every user as little privileges as possible for them to get what they need from the system. Using this principle of minimal privilege, you will vastly reduce the chance of an intruder performing operations that could crash the application or even the entire platform in some cases (thus adversely affecting other applications running on that same platform or system).

Other considerations for authentication and access control include things such as password expiration, account lock-outs where applicable, and of course SSL to prevent passwords and other account-related information being sent in plain view.

6. Don't Forget Hosting/Service-Focused Measures

Equally important as development-focused security mechanisms, proper configuration management at the service level is necessary to keep your web applications safe.

7. Avoid Security Misconfigurations

Given the endless amount of options that contemporary web server management software provides, this also means that there are endless ways to really muck things up:

- Not protecting files/directories from being served
- Not removing default, temporary, or guest accounts from the webserver
- Unnecessarily having ports open on the webserver
- Using old/defunct software libraries
- Using outdated security level protocols
- Allowing digital certificates to expire

Have a well-documented process for not only setting up new websites but also for setting up the web servers and the software used to serve those websites.

The modular nature of web server features allows for more granular control over resources and security. Although, this can make your applications less secure if you

are not careful when using them. Be extremely cautious and careful when managing more high-risk security options and features.

8. Implement HTTPS (and Redirect All HTTP Traffic to HTTPS)

We had discussed encryption previously with development-focused approaches. Encryption at the service level is also extremely helpful (and sometimes necessary) preventative measure that can be taken to safeguard information. This is typically done by using HTTPS (SSL or Secure Sockets Layer).

SSL is a technology used to establish an encrypted link between a web server and a browser. This ensures that the information passed between the browser and the webserver remains private. SSL is used by millions of websites and is the industry standard for protecting online transactions.

In addition, blanket use of SSL is advised not only because it simply will then protect your entire website, but also because many issues can crop up with resources like stylesheets, JavaScript or other files if they aren't referenced via HTTPS over an SSL.

9. Include Auditing & Logging

We are also concerned with auditing and logging at the server level. Thankfully, much of this is built into the content serving software applications such as IIS (Internet Information Services) and is readily accessible should you need to review various activity-related information.

Not only are logs often the only record that suspicious activity is taking place, but they also provide individual accountability by tracking a user's actions.

Different from Error Logging, Activity or Audit Logging should not require really much setup at all since it is generally built into the webserver software. Be sure to leverage it to spot unwanted activities, track end user's actions, and to review application errors not caught at code-level.

In extremely rare cases, logs may be needed in legal proceedings. As I am sure you well know, in these cases the handling of the log data is critical.

10. Use Rigorous Quality Assurance and Testing

If your situation at all allows you to, utilizing a third-party service that specializes in penetration testing or vulnerability scanning as an addition to your own testing efforts is a great idea. Many of these specialized services are very affordable.

It is better to be overly cautious when possible, and not rely on only your own in-house quality assurance process to uncover every little hole in every little web application you are using. Adding another layer of testing to catch a few holes here and there that were perhaps not identified by other means of testing is never a bad thing.

To make security upgrades and routine testing efforts go more smoothly, have a well-defined and easily replicable process in place, as well as a thorough inventory of all web applications and where they exist. Nothing is more frustrating than trying to fix security bugs with a specific code library, but to only then have no idea which web applications are even using it!

Your web applications should also be free of any vulnerabilities or breaches that would fail any PCI or HIPAA guidelines. To be certain of this, you should be diligent in all these areas with your approach and design. Whenever possible, you should consult with a party that specializes in adherence to these guidelines so that you can be fully confident that you have everything in place to not only thwart attacks but to simply follow the rules put forth by governing agencies as well.

11. Be Proactive to Keep Up With the Bad Guys

When I talk to people about cybersecurity I often use military analogies and phraseology, since cybersecurity seems to me like an arms race. Threats are constantly evolving and developing new attacks and tactics are constantly being developed. Businesses with an online presence must counter these threats to keep up with the 'bad guys' out there.

Like a good military strategy, the key to cybersecurity is proactivity.

You should have a well-defined blueprint for a security plan for all your sensitive web applications. This means prioritizing your more high-risk applications. It can be easier to identify if you have an inventory or repository of all the web applications that your business uses or provides to its end users.

As security threats evolve, so should your approach and plan for handling them. Increasingly sophisticated adversaries and ever-expanding soft spots as we turn to web applications to solve more and more of even our most tenable business needs is a concern that requires a full-time effort.