CO5:- Determine the applicable methods of Cryptography, Steganography and Vulnerability Assessment.

Syllabus:- Cryptography → Cryptography, Digital signature, Hash functions, Steganography → Steganography Process, watermarking, Steganography methods & Attacks, Steganography tools, Vulnerability Assessment → Vulnerability, The open web Application Security Project (OWASP), Prevention, Damn vulnerable web application (DVWA), installation & testing of DVWA.

Cryptography:- Cryptography is a technique which is used for the secured communication by changing the message (or) information into encoded format. Some special algorithms are used to encrypt the information & the information exchange became secure. The algorithms used in Cryptography are mathematical algorithms & converts the plain text information into unreadable coded format which is known as cipher text. while the process of Cryptography, the information is encrypted by using a KEY which makes it more secure. Receiver would not be able to decrypt the information without knowing the key used to encrypt the information.

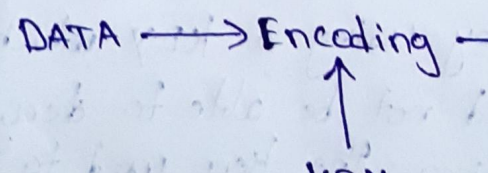
* These keys are transmitted between the sender & receiver, and are Generally of two types:

1. Public Key:- In the Public Key, the data is encrypted using the recipient's public key & it can't be decrypted without the matching Private Key.

2. Private Key:- In the Private Key, the data is encrypted using the Private key & can be decrypted only by the matching Public Key. The keys will not work if interchanged. In case the locking key is Private, it used to verify that the document is locked by the owner & hence it is mostly used in making of digital signatures. The information can be decrypted by the user having the matching Public key of the cipher text.

* There are two types which play an important role in cryptography:-
  1. Encryption
  2. Decryption.

a. Encryption:- In the encryption, the information present in plain text is encoded using specific algorithm by means of a KEY. The information which obtained as an output is known as encrypted data.



b. Decryption:- Decryption is the Process of Converting the cipher text into plain text. In the Process of decryption, the receiver decodes the encrypted information by using the key shared. To decrypt the information, Presence of Private key is

and & without it decryption cannot be done. Both of encryption & decryption make up the whole term Cryptography.

Digital Signature:- Digital Signature is used for defining the authenticity of the digital documents. It is based upon the Private key encryption because the user locks the document by using his digital signature. Digital signature is the electric signature of the user which is used for secure digital purposes & used for authenticating confidential information.

To generate a random Private key, a key generation algorithm is used which select a random key from the possible keys. Generated Private key & information is combined by using the signing algorithm, finally signature verifying algorithm is used which checks whether the Public key is matching (or) not. In case, the Public key matches with the Private key, authenticity is defined. Digital signatures are widely used in government sector related works.

Hash Functions:- Hash functions are also defined as one way cryptography. In the hash functions there is no involvement of the key during the encryption process. The plain text information is converted into hash by the suitable algorithm. The plain text is non-recoverable from the hash function because the hashes are one-way Property.

Important cryptography algorithms:-

1. Rivest Shamir Adleman (RSA)
2. MD5
3. Secure Hashing Algorithm (SHA)

1. RSA:- RSA is the very first Public key based cryptographic system & which is widely used for the secured data transmission. In the RSA, a user generates a public key based upon two large Prime numbers having an auxilary value. User publishes the generated public key keeping the Prime numbers secret.

The published public key can be used by anyone for encrypting the information. A user having the knowledge of Prime number can efficiently break the encrypted message & decode it. Decoding the RSA encryption is generally known as RSA Problem. RSA is one of the slowest algorithms and due to this is not much used for encrypting the user data.

2. MD5:- MD5 is widely used hashing algorithm for generating 128 bit hash. It is generally used as a data verification checksums against unintentional corruptions. MD5 is one-way hash function but it can be cracked (or) reversed by using brute force attacks. MD5 is the advanced series of message-digest functions. Looking at security point of view, MD5 hash security had compromised many times. Using collision attacks made it possible to crack MD5 hash.

SHA:- SHA is a hashing algorithm which takes an input of arbitrary length. The out of SHA is 160 bit and it is Quite slow than MD5. Secured hashing algorithm is generally used for the authentication related encryption purposes. It is also used for integrity checksum & for secured web Connections. SHA is generally bigger than MD5 in length.

Versions of SHA are following

1. SHA-1
2. SHA-2
3. SHA-3

Practical:-

True Crypt:- Download:- www.truecrypt.org/downloads

Note:- True Crypt is not used in latest version of windows. Bit locker is introduced for encryption & it is Pre-loaded.

Using True Crypt:-

* Download & install truecrypt. It will show list of drives.
* Select drive & click on mount. you can choose specific files to encrypt them.

Online MD5 Encryption:-

1. open your web browser & visit to www.md5encryption.com.
2. Input message which you want to encrypt & click on encrypt it.

For Ex:- Hello

3. The MD5 hash will be generated.

For Ex:- f8148937777bcc2295fff05f00e508da6.

4. This word used in example is a normal dictionary word & can be easily cracked using brute force.

3. Online MD5 Decryption:-

1. Open your web browser & visit to http://md5decrypt.net/en/ Input MD5 string which you want to decrypt & click on decrypt it. For Ex: 8b1a9953c4611296a827abf8c47804d7.

2. The MD5 hash will be decrypted & original message would be retrieved

   For Ex:- Hello

3. This word used in example is a normal dictionary word & can be easily cracked using brute force.

4. Using SHA-1:-

1. Open your browser & go to "www.sha1-online.com".
2. Now it will open a website from where you can Convert your simple text into sha-1 hash.
3. Insert your text, for ex: text is "I want to be a hacker of ethics" & click on hash button.
4. Now you can have your Sha1 hash. It is alphanumeric hash.

Note:- You can choose other hash from drop-down list.

5. Using Gost:- 1. Open your browser & go to "www.sha1-online.com".
2. Now it will open a website from where you can Convert your simple text into Gost hash.
3. Insert your text for ex: text is "I want to be a hacker of ethics" & click on hash button.

Now you can have your Gost hash. It is alphanumeric hash

Note:- You can choose other hash from drop-down list.

Steganography:- Steganography is an art of hiding the information within the files. Sensitive messages & information are hided into the multimedia files without being detected by the Process of steganography. It allows anonymous & secure interchange of information without being detected easily. The hidden information is not visible with naked eyes & hence the chance of being detected just by seeing the files containing the hidden information is near to impossible. Steganography can be done with media, files & folders.

For example an attacker hides some confidential information within a Picture using the steganography. The new image which contains the information hidden with in will be now infected image. Attacker use a brand new car's Picture & upload it over the social media to secretly without suspiciousness & the receiver will download the Picture & extract the information using the tools. Now the attacker will remove the picture from social media. In this whole Process, attacker Pretended to share the Pictures of his new car but actually the secret information was shared without knowledge of anyone.

Steganography Process:-

1. The target message is first encrypted & then combined with the target file by the means of special tools which have permissions to modify the files.
2. The encrypted data is appended with the target file by using special algorithms which makes the data hidden into the file & makes it invisible to naked eyes.
3. The information is visible to the some special exceptional programs which are designed for steganography analysis.

Terms Associated with Steganography:-

Cover Medium:- The medium in which the information (or) the target message is to be hidden is known as Cover medium. Cover medium is initial face of deciding where the information should be hidden in the medium.

Stego-medium:- The medium in which the information (or) the target message is hidden is known as the stego medium. In the stego-medium, the information has been already hidden somewhere into the medium & this is the next phase after Cover medium.

Information:- The plain text (or) data which is to be hidden with in any particular datatype is known as information. Everything is performed for the security & confidentiality of information.

Watermarking :- watermarking is a similar process to the steganography, which is used for the protection of the documents by keeping a copy right of the owner. Its primary goal is not be destroyed (or) extracted. watermarking is generally used with multimedia files to protect the intellectual property rights. watermarks are also used with documents which are visible watermarks. It may be used to make information temper proof by using as fingerprint to the information for detection of changes.

Steganography methods :-

1. Traditional methods
   a. Hidden tattoos
   b. Using wax paper
   c. Using the news articles by highlighted text method.
   d. Microdots & Symbolic Communication.
2. Modern methods
   a. Plain text
   b. Hyper text
   c. Image
   d. Video
   e. Audio
   f. Executable.
   g. Network Packets.

* Modern methods are widely used for the steganography & secure information exchanges. In this book, we will discuss about some basics of modern methods.

1. Plain text:- One of the common methods of steganography is using plain text. Plain text steganography can be done using the letters present in a paragraph (or) sentence. special shapes (or) text highlighting is used for this method.

2. Hypertext:- Steganography based on hypertext is similar to plain text. Generally the message is hidden with in the file using the comics which is generally not visible to a normal user & hence can be viewed by the inspection of source code & hence might be used for steganography. In this case the method is not much secure because an advanced user can easily detect this steganography. Sometimes it may present within the phrases, images (or) any other page content settings.

3. Audio:- Audio steganography is one of the most commonly used techniques. It can be done by digitally embedding a file into audio files (or) hiding any information directly within the audio file which can be extracted directly. It can be done by creating some additional music nodes (or) additional music sequence on the sheet.

Digital embedding is a process of embedding a message into audio files. Redundant bits are used for embedding the message into the files. Since the audio is one of the inaccurate data formats, slight changes in the bits can't be easily detected. Generally least significant bits are used for redundant bits.

4. Video:- Messages & information can be hidden in the videos by the help of steganography. Videos steganography is widely used for the secret information interchange. It is somewhat.

similar to the hypertext steganography. unlike the audio & video steganography, messages can also be hidden using slight different colours which can't be detected easily by visual lookup.

5. Image:- Most common & widely used technique is image steganography. An image is used to hide the data & information within it. From the close inspection (or) naked eye inspection, the hidden information can't be detected. Just like audio & video steganography, some redundant bits are used for the steganography. The original image & the infected image (steganography performed image) are exactly same on looking with naked eyes (or) visual look up. It can only be detected using the named technique steganalysis.

6. Executable files:- steganography can be done with the help of executable files. Some specially designed tools are used to hide information with in the executable files. It also uses redundant bits for the steganography. The executable file is not effected by the hidden data & also there is no visual detection of steganography in it.

Steganalysis:- steganalysis is the process of analysing & detecting steganography. Some special techniques & tools are used for steganalysis. Generally the statistical analysis is used for the steganography detection.

Steganalysis attacks:-

1. Stego only attack:- In this type of attack only the stego file is available to the attacker. It means that an attacker can

only read the stego file to retrieve the hidden message.

2. Cover Attack:- In Cover attack, An attacker Compares the original file with stegofile to detect the Pattern differences. For example an original & stego image is Compared to know the Pattern variance in that to find whether the steganography is done (or) not.

3. Visual Detection:- steganography can also be detected by using visual lookup. Sometimes the unusual variance & Patterns can lead to the failure & detection of the steganography. Generally due to lack of Proper encrypting with in the image, it is detected by viewing the image. Specially, in case when the steganography is done using Colour variance.

Steganography using tools:-

1. Net tools:- Download net tools using the URL http://mabsoft.com/net tools.html

using Net tools:-

1. Net, tool is one solution for beginers as well as intermediate users & it Contains more than 100 tools for hacking. Download & run net tools.
2. There is a drop down named tools from their choosen steganography.
3. Now click on load image to open a image which we want to hide a message.
4. In the message box enter message we want to hide & click on hide text. Now save the image output & it will Contain your secret message.

If we want to extract message just load the image file & click on extract message. If image Contains any message it will get seperated from the image & display it.

2. Quick Stego:- To download this tool follow the URL ie, www.Cybernescence.com/ [redacted]

Using Quick Stego:-

1. Download & run Quick stego
2. Click on open image & select the image which we want to hide the information & click on open text to open the text file which we need.
3. Click on Save image & it will Contains Secret message.
4. To extract the message load the image & click on extract message. If the image Contains any hidden message it will extract from the image.

Vulnerability Assessments:-

Vulnerability:- In simple terms, vulnerability is weakness present in any System. Vulnerability gives attacker advantage to use it to exploit the target System. Just like human gets a disease because of deficiency (or) weakness in immune System, this weakness is actually vulnerability in immune System & a disease uses that weakness to spread into human body. Similarly, vulnerability is a weakness which leads to the exploitation of the target System.

The open web Application Security Project (OWASP):- OWASP is international open source foundation. OWASP declares the list of top vulnerabilities on the basis of threat level & risk factor. This list is known as OWASP TOP 10. OWASP TOP 10 vulnerabilities are recognized as the standard vulnerability lists. Threat from these vulnerabilities is very high & cause potential damage to the web application.

OWASP also declares the list for mobile vulnerability with the name of OWASP mobile Security Project. OWASP Zed Attack Proxy (ZAP) is one of the open source tool used for Penetration testing. OWASP zap is available online for free. It helps the user to automatically find security vulnerabilities in the target website. This is mostly useful when you want to test developing web applications. OWASP ZAP is also used for manual Penetration testing & Generally used by Professionals for manual testing. OWASP ZAP Comes Pre-installed in kali linux.

OWASP TOP 10-2021:- OWASP TOP 10 is a flagship Project of OWASP foundations. It is the list of 10 most threatening vulnerabilities which are found in web applications. OWASP redefines the TOP 10 list from every three years. Along with the list of TOP 10 Critical vulnerabilities, it Provides the whole documentation to learn & test for these security vulnerabilities in web applications. This Project is Completely open-source. All the Penetration tester & bug bounty hunters follow OWASP TOP 10 vulnerability standards while testing web applications. OWASP Projects are open source & for awareness Purposes.

Following are the top 10 vulnerabilities:-

1. Broken Access Control
2. Cryptographic failures
3. Injections
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable & outdated Components
7. Identification & Authentication failures
8. Software & data integrity failures
9. Security logging & monitoring failures
10. Server-Side Request forgery (SSRF)

These are Top 10 vulnerabilities according to the OWASP Top 10, 2021 edition. The open web application Security Project top 10 vulnerability documentations are also present in various languages like English, Japanese, French, Turkish, Korean, Spanish & other various languages.

1. Broken Access Control:-

Description:- Users cannot behave outside of their intended permissions because access control policies are enforced. However failures in the proper implementation of these policies result in unauthorized information exposure, data change (or) deletion, (or) executing a business function beyond the user's capabilities.

Application flaws that allow unauthorized attackers to create, read, update (or) delete actions in the application (or) data related to it are known as Broken Access Control. This can conclude data from other users (or) stored at the system level, such as

Password files. Even in the API domain, challenges with broken access control are standard. This may be due to lack of understanding & implementation of Authentication & Authorization! That is driving these problems to become more prevalent even in 2022.

Approach:- "Direct object Access' is a typical example of a broken access control system. It happens when a query Parameter in a URL Contains something like a database, Customer ID (or) UUID. When an attacker notices the ID in the query Parameter, the attacker manipulates (or) Probes for other numerical values.

* you are changing your Cookie / JWT token to spoof another user's session.
* Modifying the URL to access details of other users:
  http://testing.com/user/1234/ address http://testing.com/user/1235/ address.
* Changing the URL to gain access to Pages only an admin can see.
* http://testing.com/user/details.
* http://testing.com/admin/secretdetails.

Impact:- The Consequences can be served depending on the same vulnerability. The worst case scenario occurs when an unauthorized person obtains access to a Privileged function. This could allow them to change (or) delete material on the website (or), even worse, get Complete Control of the online application.

Prevention:- These Problems Could be resolved by appropriately establishing access Controls.

* Model access Controls should guarantee record ownership instead of allowing the user to Create, read, edit (or) delete any record.
* To reduce the risk of automated attack tools such as brute-force sessions, set API & Controller access rate limits.

2. Cryptographic failures:-

Description:- In OWASP's TOP 10 2017 list, Cryptographic failures was named Sensitive data Exposure. When Sensitive data is not saved Securely, insecure Cryptographic storage is a typical issue. Insecure cryptographic storage is a set of weakness rather than a single flaw.

The first step is to analyze the security requirements for data in transit & at rest. For example, passwords, credit card numbers, health records, personal information, Corporate Secrets, etc., require additional security, especially if the data is subject to Privacy regulations & laws.

Approach:-

* Encryption is not enforced while communicating with it & Server Certificates are not validated.
* Data is sent in clear text via Protocols including HTTP, SMTP, FTP & TLS enhancements like STARTTLS. "External internet traffic Poses a threat".
* Randomness that was not meant to meet cryptographic standards was used for cryptographic Purposes.
* Examine whether the received Server Certificate & the trust

chain have been appropriately validated.

* Is it possible to exploit cryptographic error messages (or) side channel information by Padding oracle attacks?

Impact:- when a Company is the victim of a data breach, it suffers. Even when violations are fixed, users begin to recognize them as untrust worthy (or) unprotected, making them more unwilling to share sensitive information. Client Confidence is critical to an Organizations success, & without it, it would almost certainly fail. when a data breach reaches mass Proportions & affects millions of People, it attracts media attention exposure to media damages Company's reputation, which last for year

Prevention:- * when sending sensitive data, avoid using older Protocols like FTP & SMTP.

* Instead of just encryption, Consistently implement authenticated encryption.
* Ensure that standard algorithms, Protocols & key are up to date & robust; utilize Correct key management.
* check out whether you're using any known bad algorithms.

Injection:-

Description:- Injection slides down to the third Position. Again 94% of the applications were tested for some form of injection with a max incidence rate of 19%. an average incidence rate of 3%. & 274k Occurrences. An attacker against a Parser (or) interpreter that relies on user-supplied input known as "Injection".

user input is one of the sources of injection attacks. If it is implicitly trusted, without any sanitization, filtering (or) escape

malicious user might control the application's response & possibly even the server. This occurs when the user's data is mixed with the interpreter. SQL injection is a classic example in which standard user input is manipulated to generate several SQL statements.

Any input field can be used as a source of injection. Injections attacks come in various forms like SQL injection, LDAP, Xpath, XML Phrases, SMTP headers, & others.

Approach:-

* The application does not validate, filter, (or) sanitize user supplied data.
* In the interpreter, dynamic queries (or) non-parameterized calls without context aware escaping are used directly.
* To extract additional, sensitive records, hostile data is employed with in object relational mapping (ORM) search criteria.

Impact:- Injection harm the web application. An attacker injects queries into the login field (or) any other field to access the data base (or) sensitive data disclosure. Injection vulnerabilities are most commonly observed in LDAP, OS Command, & SQL. As a result, it is the most dangerous vulnerability since data loss (or) theft can result in significant financial loses.

Prevention:-

* As a secondary defence perform allowlist input validation
* Escaping all user supplied input

* Continuous monitoring of SQL statements.
* Enforce Prepared statements & Parameterization.

Insecure Design:-

Description:- It's a new add on to the OWASP top 10. Insecure design a board term that describes a variety of flaws & is defined as "missing (or) poor control design". The other top 10 risk categories are not caused by insecure design. There is a clear differentiation between insecure design & execution. We distinguish between design flaws & implementation defects for a reason: their root causes & remediation are different.

Even if a design is secure, implementation flaws can lead to vulnerabilities that can be exploited. Because needed security controls were never created to defend against specific attacks, Perfect implemention cannot fix an insecure design.

Approach:-

* Lousy Password Policies are another typical insecure design vulnerability. While specific Password standards are constantly Evolving & debatable, restricting the maximum size exposured an application to a brute-force attack.

* utilizing unsafe APIs (or) functions can lead to security issues: Consider using random numbers without seed (or) extracting an archive without considering the absolute (or) relative Path that the contained files may have.

* Applications that have more rights then are required.

Impact:- Insecure application design can have serious business consequences, as it can allow attackers to tamper with the application logic, exposing critical information (or) Compromising a web application.

Recent IDOR vulnerabilities in wordPress plugins show how simple it is to take control of an online web application.

Prevention:-

* Detailed code reviews are required to prevent any vulnerable code from making it into Production.
* Creating a solid treat model & reference architecture.
* user (or) Service resource Consumption should be limited.
* user stories should include security language & Controls.

Security Misconfiguration:-

Description:- As security misconfiguration occurs when a component is vulnerable to attack due to an insecure configuration. Misconfigurations are frequently viewed as an easy target since they are easy to identify on misconfigured web servers, cloud services & apps & they can be exploited, causing severe harm.

Insecure default configurations, incomplete (or) ad-hoc configurations, unprotected cloud storage, misconfigured HTTP headers, superfluous HTTP methods, overly permissive Cross-origin resource sharing (CORS) & verbose error messages are all instances of security misconfigurations.

Approach:-

* Default accounts & passwords remain active & unmodified.
* The server does not send security headers & directives, (or) they

not set to secure values.
* The software is either outdated (or) insecure.
* Insufficient firewall Protection.

Impact:- The impact varies & is determined by the type of misconfiguration. In the worst-case scenario, it could result in a complete takeover, resulting in the theft of critical data & the subsequent cost of recovery.

During the reconnaissance phase, attackers can take advantage of security misconfiguration to learn about the application & API components.

Attackers can also use misconfigurations to pivot their attacks against APIs, as in an authentication bypass caused by misconfigured access control measures.

Prevention:- * unnecessary features that aren't needed in Production environments should be disabled.
* Conduct frequent scans & audits to detect future misconfigurations
* In all environments, they are using an automated method to verify the effectiveness of the setups & settings.

6. Vulnerable & outdated Components:-

Description:- This a hazardous category because of the frequency of vulnerable & outdated components & the ease with which attacks can be launched utilizing this vector. Open-source Packages are used by almost all current Programs/applications & information regarding vulnerabilities in these Packages is Publicly available.

Attackers who figure out the vulnerable Packages you're using can employ Publicly known exploits.

Approach:- * client & Server-Side Code with vulnerable components.
* Insecure Software Configuration.
* The dependency chain of the components uses old/unpatched dependencies.
* If software developers do not test updated, upgraded, (or) patched libraries for Compatibility.

Impacts:- The Potential impact is impossible to evaluate because it depends entirely on vulnerable Component & the vulnerability to which it is exposed. using Components with known vulnerabilities can have a more severe business impact.

Prevention:-
* Install the Components using trusted channels & double-check their integrity.
* Remove any unwanted dependencies, features, Components, files (or) documentation.
* If Patching is not an option, try using a virtual Patch to track, identify, & Protect against the Problem.
* Keep an eye out for libraries & Components that haven't been updated in a while (or) that don't have security fixes for older versions.

Identification & Authentication failures:-

Description:- Broken Authentication has moved down the list to Position #7. To guard against authentication-related threats, it's essential to Confirm the user's identity, authenticate them, & manage their sessions.

Broken Authentication refers to various issues caused by errors in authentication & session management implementations.

Everything from a login without a timeout, which means that users who forget to logout on a Public Computer can be hacked to more technical flaws like session fixation, is included in this category.

Approach:-

* Flaws in the password reset (or) recovery flow
* The session identification is exposed in the URL.
* Passwords like "Password1" (or) "admin/admin" are default, weak, (or) well-known.
* Is multi-factor Authentication missing (or) ineffective?
* Missing multi-factor Authentication

Impact:- An attacks Purpose is to gain Control of one (or) more accounts & grants the attacker the same rights as the targeted user. For example if an attacker successfully hijacks an admin account, the attacker will be able to do all of the functions of a regular admin, which, depending on the Programme, Could have a significant impact.

Prevention:-

* whenever possible, multifactor authentication should be used to Prevent automated Credential stuffing, brute force & stolen Credential reuse threats.
* Make sure that default Passwords are only used once & changed the user logs in.
* It's essential to follow updated Password guidelines.
* limit the no. of login attempts & increase the duration

...from two (or) more tries for critical endpoints such as login & payment (password)

* Software & Data integrity failures:-

Description:- Code & infrastructure that do not protect against integrity violations are software & data integrity failures. A Programme that uses plugins, libraries (or) modules from untrusted sources, repositories (or) content delivery networks is an example of this (CONs).

unauthorized access, malicious code, (or) system compromise can all be risks of an unsecured CI/CD Pipeline. Finally, many Programmes now have auto-update capabilities, allowing updates to be obtained without necessary integrity checks & applied to previously trusted applications. As a result, attackers might theoretically distribute & run their updates across all systems.

Impact:- Attackers can tamper with critical data used by the Programme if it is not checked, leading to major concerns such as the insertion of malicious code into the software.

Many Programmes now have automatic software updates, raising worries regarding data integrity during the update process. Furthermore, if attackers can use a MitM attack to inject malicious code into a Programme during the update process, such updates must never be installed, (or) the application will be compromised.

Prevention:-

* Always be sure the app you're using is trustworthy & adheres to the same security standards.

* Always sign your application components to ensure that the software & data you're using are from a reliable source & haven't been tampered with.
* Determine that the libraries & dependencies, such as npm (or) Maven, use trusted repositories.
* Ensure your CI/CD Pipelines are secure & that no malicious code enters.

- Security logging & monitoring failures:-

Description: This category which returns to the OWASP top 10 2021, is designed to detect, escalate & respond to active breaches. Breach detection is impossible without logging & monitoring. It can happen at any time. insufficient logging, detection, monitoring & active response.

There is no direct vulnerability that can occur due to these flaws, but logging & monitoring are significant in general & their absence (or) failure can directly impact visibility, incident alerting & forensics. As a result its critical to have a working logging & monitoring system in place to collect logs & issue alerts if any malfunctions (or) problems occur; otherwise, they could go unreported for a long time, causing a lot more damage.

Approach:-

* If the app server that holds the logs locally fails, the records are not backed up.
* Warnings & errors provide no, insufficient (or) confusing log messages.

Monitoring systems are incapable of detecting suspicious activities (or) raising alarms in real time.

Logins, unsuccessful logins, & high value transactions, are not logged.

Impact:- without effective recording & monitoring procedures, organizations will have a much harder time detecting & mitigating breaches, which will cost them time & money.

Proper logging & monitoring techniques can identify users & processes engaging with a system can be identified more efficiently. However it's impossible to track out the source of a message/request without proper recording techniques. This makes tracking the source of a threat difficult, which encourages system attacks.

Prevention:-

To prevent injections (or) attacks on logging (or) monitoring systems, make sure log data is encoded correctly.

Determine that the logs contain all essential information & are well-formatted for consumption by other tools (or) log management solutions.

DevSecOps teams should setup good monitoring & alerting to swiftly recognize & deal with suspicious activity.

Server-Side Request Forgery (SSRF):-

Description:- when a web application fetches a remote resource without validating the user-supplied URL, an SSRF fault occurs. Even if the programme is secured by a firewall, VPN, or another sort of network access control list, an attacker can force it to send a forged request to an unexpected location (ACL).

An attacker can get access to internal (or) external services to read (or) transfer confidential information such as Aws temporary credentials assigned to EC2 instances, internal HTTP resources (dev servers), execute internal network port scanning, & so on.

Approach:

* Import function: Ensure to intercept every request when importing photos from the server to your target.
* web hooks:- A webhook sends data to other programmes in real-time so you get it right away.
* An application that sends GET/POST requests to the webhook URI provided.
* Reading the cloud metadata: The data can be retrieved from the cloud instance. The access key, secret keys, & other credentials can all be retrieved.

Impact:- The true impact of SSRF could be determined by how much data is viewed (or) read. The severity of SSRF ranges from mild to severe. The attacker can scan the network's ports. In addition, he can scan any other website port through the victim's network.

Unauthorized activities (or) access to data within the business can often arise from a successful SSRF attack, either in the vulnerable application itself (or) on other back-end systems with which the programme can interface. Additionally, the SSRF vulnerability could allow an attacker to execute arbitrary commands in some circumstances.

An SSRF vulnerability that establishes connections with external third-party systems could lead to malicious onward attacks that appear to come from the business that hosts

the vulnerable application.

Prevention:- To minimize the effect to SSRF, split remote resource access functions into separate networks.

* Allowlist Domains in DNS.
* Sanitize & validate inputs.
* Enforce the URL schema, Port & destination with a positive allow list.
* Enable authentication on all services:- Even if they don't require it, ensure that any service running inside your network has Authentication enabled.

Damn vulnerable web application (DVWA):- DVWA is a specially designed vulnerable web application which is used to learn real time vulnerability assessment. DVWA contains most of the vulnerabilities. A tester can perform testing on it. It is completely open source project. There are many other web applications which are available to check vulnerability assessment & penetration testing skills & some live bootable image files are also available which can be run as virtual machine.

Download DVWA :- http://www.dvwa.co.uk/

Installing DVWA on local host:-

1. Download the DVWA package from its website.
2. Download XAMPP to run DVWA on local host. Download XAMPP: https://www.apachefriends.org/download.html.
3. Install & run the XAMPP Control Panel.
4. Install Apache & MySQL Server from XAMPP Control Panel &

allow them through firewall. From the Control Panel start both servers.

Extract the DVWA archive downloaded & Put the folder into "c:\xampp\htdocs".

Now DVWA will run on your local host.

Open your browser & type "127.0.0.1" (or) "local host" to open the local host server. This is generally used for the testing web application on local server.

Navigate to "127.0.0.1/dvwa/login.php" (or) "localhost/dvwa/login.php". username: admin Password: Password.

A mysql error will be encountered. Now navigate to: "c:\xampp\htdocs\dvwa\Config\Config.inc.php".

Open this file using any text editor & find the line: "$DVWA ['dbPassword']='PASSWORD'".

Change this line to the following: "$DVWA['dbPassword']="";

Now again visit to the dvwa login Page & this time no error would be encountered.

Testing with DVWA:-

1. SQL injection:- SQL injection is one of the common & most threating injection vulnerability. An attacker injects the SQL queries into data field like form fields (or) login Page in order to bypass the security & get access to the databases. Sometime it leads to the Complete Host takeover.

Run DVWA on local host & login into it.

Click on DVWA Security & set it to Low.

click on sql injection button in left sidebar.
Input any sql queries into the USER ID field to check whether it is working (or) not. For ex: input '3'. Now if it shows the user details present at id = 3 then it is working.
To see the all users into database input this query: " % ' or 0=0 union select null, user() # ".
Last detail in which only username is shown as "root@localhost" indicates the user who injected the query.
Download sql cheat sheet from the internet & try to taste different queries & analyse their behaviour.
•XSS:- Cross-Site Scripting is already explained above & is the one of the critical vulnerability.
Run DVWA on local host & login into it.
click on DVWA Security in left sidebar & set DVWA Security to Low.
click on XSS. For the beginning, start with stored XSS.
In the Name field input the name & in message field input javascript. For ex: Name: Harsh message: <Script> alert ("Hacked!") </Script>
click on Sign Guestbook. The javascript will get stored. Now again input name & message with anything & click on Sign Guest book. The javascript which was submitted earlier will get executed & a Popup will be shown up.
Download xss cheat sheet from the internet & try executing different xss.

3. Cross - Site Request Forgery :-

1. Run DVWA on local host & login into it.
2. click on DVWA Security from the left side bar & set the Security as Low.
3. click on CSRF from the left side bar.
4. Input the New Password & Confirm New Password with any Password & click on change. For ex: harsh@harsh.
5. Password changed massage will be shown below the change button.
6. Now check the URL. There will be two strings which are seperated by the "&" : Password_new = harsh@harsh Password_Conf = harsh@harsh.
7. These strings Contain the Password which has been set as new Password.
8. Change the Password Present in both the strings like: Password_new = xroot password_Conf = xroot.
9. Now reload the Page & the Password will be changed again.
10. Now logout from DVWA & try to login with changed Password & login will be successfully.
11. Visit www.owasp.org & read about more advanced uses of CSRF.