SP 2025

بسم الله الرحمن الرحيم

KHAWAJAGAN Technologies

Secure Software Design and Development

SP 2025

CYC386

KHAWAJAGAN Technologies

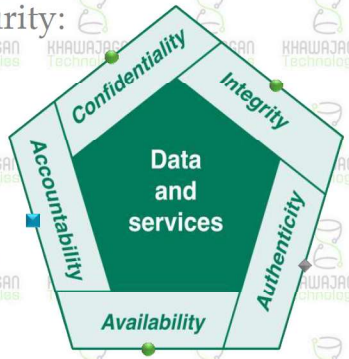
FAISAL SHAHZAD

Secure Software Development

- To develop secure software, a reasonable knowledge of security principles is required.
- Secure Software Concepts, comprises a collection of principles, tenets, and guidelines from the information security domain.
- Understanding these concepts as they apply to software development is a foundation of secure software development.
- We will start by understanding Security Concepts for their later implementation in the domain of software development.

Secure Software Development

- Actions associated with security:
  - Confidentiality
  - Integrity
  - Availability
  - Authentication
  - Authorization
  - Accountability
  - Non repudiation



CIA Triad

Confidentiality

- Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information

Integrity

- Guarding against improper information modification or destruction, including ensuring information nonrepudiation and authenticity

Availability

- Ensuring timely and reliable access to and use of information

CIA Triad

Authentication

- The property of being genuine and being able to be verified and trusted
- Verifying users are who they say they are
- Each input arriving at the system came from a trusted source

Authorization

- To verify if the person has authority to perform specific task on the data or system
- To authorize the activity to the object on the subject

Accountability

- Requirement for actions of an entity to be traced uniquely to that entity.
- This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action.

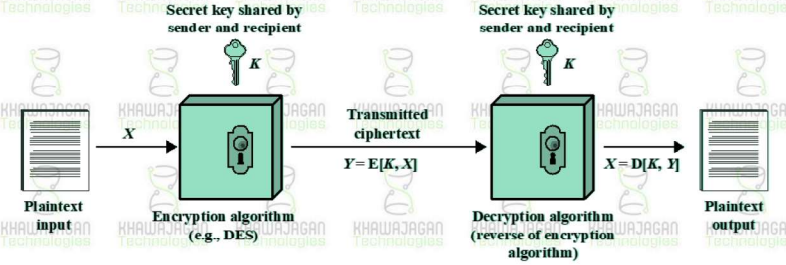
SP 2025

Confidentiality

Confidentiality

- Confidentiality is the concept of preventing the disclosure of information to unauthorized parties.
- Keeping secrets secret is the core concept of confidentiality.
- The technique employed to achieve confidentiality depends upon whether the data is at rest, in transit, or in use.
- Access controls are used for data in use and at rest
- Encryption is common for data in transit and at rest.

Encryption Process - Simplified



Confidentiality – Key Questions

- Who is authorized to see what specific data elements?
- What mechanism should be employed to enforce confidentiality?
- What are the business requirements for data collection, transfer, storage, and use with respect to confidentiality?

Confidentiality – Implementation

- Confidentiality is the protection of information from observation by unauthorized parties.
- This is done primarily through the use of encryption to make the data unreadable.
- Encryption acts by scrambling the data in a manner that hides the true data values from observation.
- Encryption is a process that requires an algorithm and a key, and those who possess the correct keys can access the data.

Confidentiality – Implementation

- Encryption elements
  - Algorithm
  - A key.
- Failure to use algorithms from approved cryptographic libraries, i.e., rolling your own crypto, almost always leads to failure.
- Failing to protect the key always leads to failure.
- Methods
  - Covert
  - Overt
- Covert Methods : once found, they are no longer covert, The secrecy value is lost.
- Overt Method : Implementation of approved encryption algorithms, Secure the key.

SP 2025

Integrity

Integrity

- Integrity refers to protecting the data from unauthorized alteration.
- Controlling alterations, including deletions, can be an essential element in a system's stability and reliability.
- Integrity can also play a role in the determination of authenticity.

Integrity – Key Questions

- Who is authorized to alter which specific data elements?
- What mechanism should be employed to detect errors and enforce integrity?
- What are the business requirements with respect to data collection, transfer, storage, and use with respect to integrity?

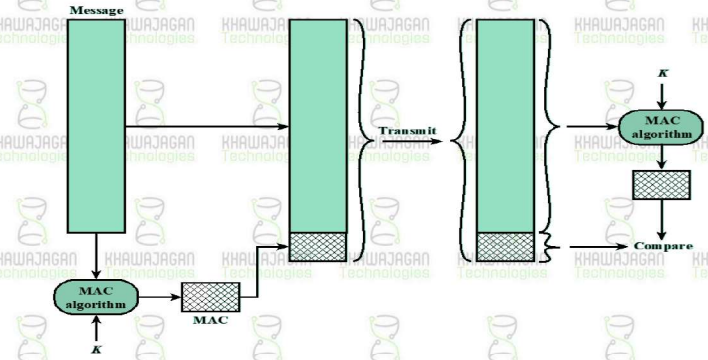
Integrity – Implementation

- Forms of alteration
  - Changing a value
  - Deleting a value
- Based on Confidentiality
  - If data is to be altered or deleted, then it is probably also visible to the user account.
- Protecting integrity can be done through the use of access control mechanisms.
- Individual specific levels of access with respect to read, write, and delete can be defined.

Integrity – Implementation - Hashing

- To verify the integrity of a data element, the cryptographic function of hashing is used.
- Hash functions can determine whether single or multiple bits of data change from the original form.
- Method: Take the original data, run it through a hash function, and record the result.
- Verification: Running the current value of the data through the same hash function will produce a current digest value. If the Value matches with original, the data is unaltered.

Integrity – Implementation - Hashing



Integrity – Implementation - Hashing

- The design issue is,
  - How does one implement this functionality?
  - When are hash digests calculated?
  - How are the digests stored and transmitted by a separate channel to the party checking?
- These decisions need to be made at the design phase so that the coding phase can implement the desired behavior.

SP 2025

Availability

Availability

- Access to systems by authorized personnel can be expressed as the system's availability.
- Availability is concerned with two issues:
  - Ensuring systems are available for authorized users when they require those systems
  - Denying access to unauthorized users at all other times.

Availability - Challenges

- The challenge in system definition and design is to determine the correct level of availability for the data elements of the system.
- Common availability issues include denying illegitimate access to systems as well as preventing availability attacks such as denial of service.

CIA Summary

- CONFIDENTIALITY
  - "Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information…"
  - A loss of confidentiality is the unauthorized disclosure of information.
- INTEGRITY
  - "Guarding against improper information modification or destruction, and includes ensuring information nonrepudiation and authenticity…"
  - A loss of integrity is the unauthorized modification or destruction of information.
- AVAILABILITY
  - "Ensuring timely and reliable access to and use of information…"
  - A loss of availability is the disruption of access to or use of information or an information system.

SP 2025

Authentication

Authentication

- Authentication is the process of determining the identity of a user.
- All processes in a computer system have an identity assigned to them so that a differentiation of security functionality can be employed.
- Authentication provides the means to define the separation of users by allowing the differentiation between authorized and unauthorized users.

Authentication

- Authentication is the process of determining the identity of a user.
- All processes in a computer system have an identity assigned to them so that a differentiation of security functionality can be employed.
- Authentication provides the means to define the separation of users by allowing the differentiation between authorized and unauthorized users.

Authentication

The four means of authenticating user identity are based on:

Something the individual knows	Something the individual possesses (token)	Something the individual is (static biometrics)	Something the individual does (dynamic biometrics)
Password, PIN, answers to prearranged questions	Smartcard, electronic keycard, physical key	Fingerprint, retina, face recognition	Voice pattern, handwriting, typing rhythm

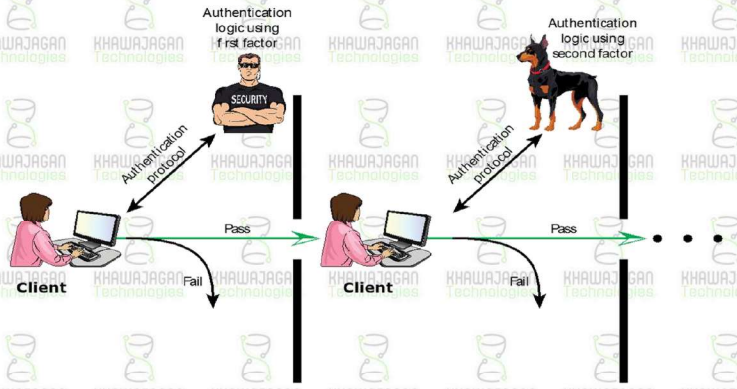
Authentication

- In U.S. government's "rainbow series" of manuals on computer security, the categories of shared "secrets" are as follows:
  - What users know (such as a password)
  - What users have (such as tokens)
  - What users are (static biometrics such as fingerprints or iris pattern)
- Because of technological advances, new categories have emerged, patterned after subconscious behaviors and measurable attributes:
  - What users do (dynamic biometrics such as typing patterns or gait)
  - Where a user is (actual physical location)

Authentication - Multifactor

- Multifactor authentication is simply the combination of two or more types of authentication.
- Two-factor authentication combines any two of these before granting access.
  - A card reader that turns on a fingerprint scanner—if your fingerprint matches the one on file for the card, you are granted access.
- Three-factor authentication would combine all three types.
  - A smart card reader that asks for a PIN before enabling a retina scanner.

Authentication - Multifactor



Authentication - Multifactor

- Multifactor authentication methods greatly enhance security by making it difficult for an attacker to obtain all the correct materials for authentication.
- They also protect against the risk of stolen tokens, as the attacker must have the correct biometric, password, or both.
- More important, multifactor authentication enhances the security of biometric systems by protecting against a stolen biometric.
- Changing the token makes the biometric useless unless the attacker can steal the new token.

Authentication – Identity Management

- Identity management is a set of processes associated with the identity lifecycle, including the provisioning, management, and deprovisioning of identities.
- The provisioning step involves the creation of a digital identity from an actual identity.
- The source of the actual identity can be a person, a process, an entity, or virtually anything.
- The identity process binds some form of secret to the digital identity so that at future times, the identity can be verified.
- The secret that is used to verify identity is an item deserving specific attention as part of the development process.

Authentication – Identity Attributes

- Identity attributes are the specific characteristics of an identity—name, department, location, login ID, identification number, e-mail address, and so on—that are used to accurately describe a specific entity.
- These elements are needed if one is to store identity information in some form of directory, such as an LDAP directory.
- The particulars of a schema need to be considered to include attributes for people, equipment (servers and devices), and services (apps and programs), as any of these can have an identity in a system.

Authentication – Identity Management

- Determining a user's identity requires an authentication process.
- Authentication is an identity verification process that attempts to determine whether users are who they say they are.
- This requires a previous identification step, where the original record of the user is created.
- The identification step requires the following issues to be determined by requirements:
  - Mechanism to be used for identity
  - Management of identities, including reaffirmations

Authentication – Identity Provider

- Identity provider (IdP) denotes a system or service that creates, maintains, and manages identity information.
- IdPs can range in scale and scope—from operating for a single system to operating across an enterprise.
- Additionally, they can be operated locally, distributed, or federated, depending on the specific solution.
- Multiple standards cover these services, including those built on the Security Assertion Markup Language (SAML), OpenID, and OAuth.

Identity Mgmt – Certificates

- Certificate-based authentication is a means of proving identity via the presentation of a certificate.
- Certificates offer a method of establishing authenticity of specific objects such as an individual's public key or downloaded software.
- A digital certificate is a digital file that is sent as an attachment to a message and is used to verify that the message did indeed come from the entity it claims to have come from.
- Using a digital certificate is a verifiable means of establishing possession of an item (specifically, the certificate).

Identity Mgmt – Identity Tokens

- An Identity Token is a security token that represents claims about the authentication of an entity by a specific authentication scheme.
- An access token is a form of identity token in physical form that identifies specific access rights and, in authentication, falls into the "something you have" factor.
- The risk of theft of the token can be offset by the use of multifactor authentication.
- A less expensive alternative is to use hardware tokens in a challenge/response authentication process.

Identity Mgmt – SSH Keys

- SSH keys are access credentials used by the Secure Shell (SSH) protocol.
- They function like usernames and passwords, but SSH keys are primarily used for automated processes and services.
- SSH keys are also used in implementing single sign-on (SSO) systems used by system administrators.
- SSH keys are exchanged using public key cryptography, and the keys themselves are digital keys.

Identity Mgmt – Smart Card

- Smart cards are devices that store cryptographic tokens associated with an identity.
- The form factor is commonly a physical card, credit card sized, that contains an embedded chip that has various electronic components to act as a physical carrier of information.

Authentication – Implementation

- Authentication is a validation that the user is presenting the known shared secret.
- This requires the establishment of a specific authentication method and a means of protecting the shared secret from disclosure.
- An authentication system may be as simple as a plaintext password system.
- For systems where more security is needed, a more complicated system such as the Kerberos system may be employed with its encrypted secrets.

Authentication – Implementation

- The requirement for authentication depends upon the level of security needed, the scale of users, and the management of users, coupled with the specific uses of the authentication.
- For a given system, multiple methods may be used; for example, high value customers may use two-factor authentication, whereas visitors may have only a simple single factor.
- System administrators may require stronger authentication based on their higher access levels.
- The specifics for all of these individual circumstances can be decomposed from policy-driven requirements.

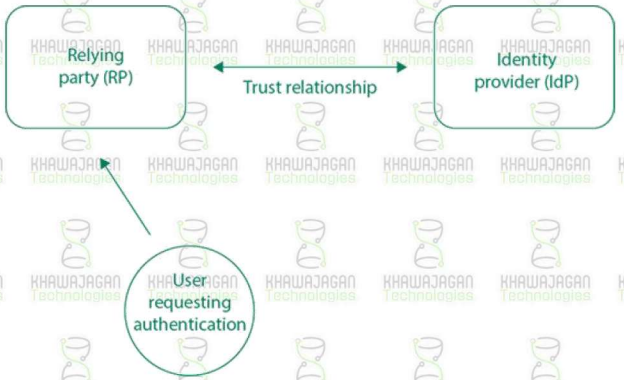
Authentication – Implementation

- Best-practice safeguards that should be included in a system.
  - Escalating time lock-out after a given number of successive failures
  - Logging of all attempts (both successful and failed)
  - Integration with the authorization system once authentication is successful
  - Password/token reset
  - Account recovery
  - Periodic password changes
  - Password strength

Authentication – Implementation

- Numerous technologies are in use for authentication and authorization.
- Federated ID systems allow users to connect to systems through known systems.
  - The ability to use your Facebook account to log in to another system is a useful convenience for many users.
  - A known user experience (UX) interface and simple-to-use method for users to have a single sign-on environment, a federated ID system can use best-practice authentication systems.
- There are two main parties in these systems:
  - A relying party (RP)
  - An identity provider (IdP).
- The user wants access to an RP and has credentials established on an IdP.

Authentication – Implementation



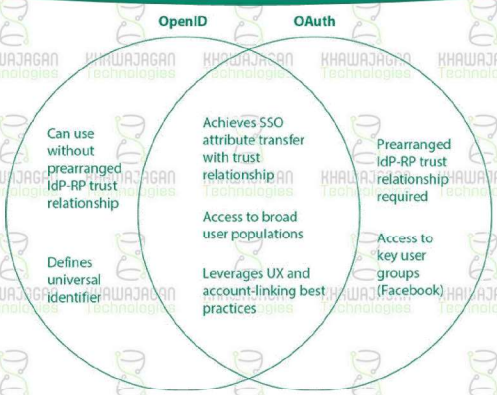
Authentication – Implementation

- Two of the more prevalent systems are OpenID and OAuth.
- OpenID was created for federated authentication, specifically to allow a third party to authenticate your users for you by using accounts that users already have.
- The OpenID protocol enables websites or applications (consumers) to grant access to their own applications by using another service or application (provider) for authentication.
- This can be done without requiring users to maintain a separate account/profile with the consumers.

Authentication – Implementation

- OAuth was created to eliminate the need for users to share their passwords with third-party applications.
- The OAuth protocol enables websites or applications (consumers) to access protected resources from a web service (service provider) via an application programming interface (API), without requiring users to disclose their service provider credentials to the consumers.

Authentication – Implementation



Authentication – Credential Mgmt

- Numerous methods of authentication have their own set of credentials (identifying information) that requires management.
- These credentials can be in the form of a password, digital strings held by hardware tokens or devices, biometrics, and certificates etc.
- Each of these forms has advantages and disadvantages.
- Each set of credentials, regardless of the source, requires safekeeping on the part of the receiving entity.
- Managing these credentials includes tasks such as credential generation, storage, synchronization, reset, and revocation.

Authentication – Credential Mgmt

- Numerous methods of authentication have their own set of credentials (identifying information) that requires management.
- These credentials can be in the form of a password, digital strings held by hardware tokens or devices, biometrics, and certificates etc.
- Each of these forms has advantages and disadvantages.
- Each set of credentials, regardless of the source, requires safekeeping on the part of the receiving entity.
- Managing these credentials includes tasks such as credential generation, storage, synchronization, reset, and revocation.

SP 2025

Authorization

Authorization

- Authorization is the process of applying access control rules to a user process and determining whether a particular user process can access an object.
- After the authentication system identifies a user, the authorization system takes over and applies the predetermined access levels to the user.
- Three elements are used in the discussion of authorization: a requestor (subject), the target (object), and the type or level of access to be granted.

Authorization – Access Control

- Access is the ability of a subject (individual /process) to interact with an object (such as a file or hardware device).
- Once the individual has verified their identity, access controls regulate what the individual can actually do on the system.
- Just because a person is granted entry to the system does not mean they should have access to all the data the system contains.

SP 2025

Accountability

Accountability

- Accounting is a means of measuring activity.
- Accountability is the recording of actions and the users performing them.
- In IT systems, this can be done by logging crucial elements of activity as they occur.
- With respect to data elements, accounting is needed when activity is determined to be crucial to the degree that it may be audited at a later date and time.

Accountability - Auditing

- Auditing is a form of recording historical events in a system.
- Operating systems have the ability to create audit structures, typically in the form of logs that allow management to review activities at a later point in time.
- One of the key security decisions is the extent and depth of audit log creation.
- Auditing takes resources, so by default it is typically set to a minimal level.
- It is up to a system operator to determine the correct level of auditing required based on a system's criticality.

Accountability - Auditing

- Organizational characteristics that can drive auditing include items such as organizational history, the business environment, and supervisory issues.
- When decomposing policies, the following organizational security elements should be explored for auditing:
  - Roles and responsibilities
  - Separation of duties
  - Training and qualifications
  - Change management
  - Control management

Accountability - Logging

- Presence of security logs is an important element in any security system.
- Logs enable personnel to examine information from a wide variety of sources to provide information about what actions transpired, with which accounts, on which servers, and with what specific outcomes.
- Many compliance programs require some form of logging and log management.
- The challenges in designing log programs are what to log and where to store it.

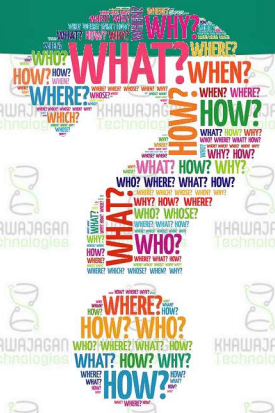
SP 2025

Non Repudiation

Nonrepudiation

- Nonrepudiation is the concept of preventing a subject from denying a previous action with an object in a system.
- Nonrepudiation is a general concept, so security requirements must specify the subject, objects, and events for which nonrepudiation is desired, as this will affect the level of audit logging required.
- If complete nonrepudiation is desired, then every action by every subject on every object must be logged, and this could be a very large log dataset.

SP 2025



KHAWAJAGAN Technologies

SP 2025

Secure Development Lifecycle

Secure Development Lifecycle (SDL)

- SDLs contain a set of common components that enable operationalization of security design principles.
- These components are as under
  - Team-awareness and education program
  - Use of security gates
    - Bug Tracking
    - Threat Model
    - Fuzzing
  - Security Reviews

Secure Development Lifecycle (SDL)

- The first components is a current team-awareness and education program.
  - Having a team qualified to do the task in an SDL environment including security knowledge.
- The next component is the use of security gates to check compliance with security requirements and objectives.
  - These gates ensure that the security elements of the process are being used and are functioning to achieve desired outcomes.
- The final element is a security review
  - Results of the SDL process are reviewed to ensure that all of the required activities have been performed and completed to an appropriate level.

SDL - Software Team Awareness

- All team members should have appropriate training and refresher throughout their careers.
- Training should be focused on the roles and responsibilities associated with each team member.
- As the issues and trends associated with both development and security are ever changing, it is important for team members to stay current in their specific knowledge so that they can appropriately apply it in their work.
- The key element of team awareness and education is to ensure that all members are properly equipped with the correct knowledge before they begin to engage in the development process.

SDL - Gates & Security Requirements

- Periodic reviews conducted during development process are referred to as gates.
- To pass the security gate, a review of the appropriate security requirements is conducted.
- Missing or incomplete elements can prevent the project from advancing to the next development phase until these elements or issues are addressed.
- This process results in eventual behavior by the development team where the gates are successfully negotiated as a part of normal business.

SDL - Gates – Bug Tracking

- Bugs are elements of code that have issues that result in undesired behaviors.
- That behavior results in something that can be exploited, and this makes it a potential security bug.
- Bugs need to be fixed, and hence, they are tracked to determine their status.
- Some bugs may be obscure, impossible to exploit, and expensive to fix; thus, the best economic decision may be to leave them until the next major rewrite, saving costs now on something that is not a problem.
- Tracking all of the bugs and keeping a log so that things can be fixed at appropriate times are part of managing code development.

SDL - Gates – Threat Modeling

- Threat modeling is a design technique used to communicate information associated with a threat to the development team.
- The threat modeling effort begins at the start of the project and continues throughout the development effort.
- The purpose of threat modeling is to completely define and describe threats to the system under development.
- In addition, information as to how the threat will be mitigated can be recorded appropriately.
- Communicating this material among all members of the development team enables everyone to be on the same page with respect to understanding and responding to threats to the system.

SDL – Security Review

- Adding a series of security-related steps to the SDL can improve the software security, however, **You get what you measure** approach needs to have an audit function that verifies that the process is functioning as desired.
- Security reviews are sprinkled at key places like between stages in the SDL.
- The purpose of these reviews is not to test for security, but rather to examine the process and ensure that the security-related steps are being carried out and not being circumvented to expedite the process.
- Security reviews act as moments where the process can be checked to ensure that the security-related elements of the process are functioning as designed.
- It may seem to be a paperwork drill, but it is more than that.
- Security reviews are mini-audit events where the efficacy of the security elements of the SDL process is being checked.

Bugs Mitigation - Introduction

- Do Nothing
  - is a tacit acknowledgment that not all bugs can be removed if the software is to ship.
  - This is where the bug bar comes into play; if the bug poses no significant threat (i.e., is below the critical level of the bug bar), then it can be noted and fixed in a later release.
- Warn User
  - Warning the user is in effect a cop-out.
  - This pushes the problem on to the user and forces them to place a compensating control to protect the software.
  - This may be all that is available until a patch can be deployed for serious bugs or until the next release for minor bugs.
  - This does indicate a communication with the user base, which has been shown to be good business.
  - Most users hate surprises, as in when they find out about bugs that they feel the software manufacturer should have warned them of in advance.

SDL - Gates – Fuzzing

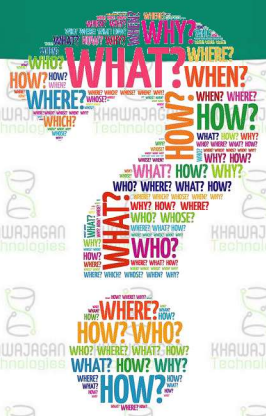
- Fuzzing is a test technique where the tester applies a series of inputs to an interface in an automated fashion and examines the outputs for undesired behaviors.
- This technique is commonly used by hackers to discover unhandled input exceptions.
- The concept is fairly simple: For each and every input to a system, a test framework presents a variety of inputs and monitors the results of the system response.
- System crashes and unexpected behaviors are then examined for potential exploitation.

Bugs Mitigation - Introduction

- When bugs are discovered, a natural tendency is to just fix the problem.
- But most bugs are not easily fixed with a simple change of the code.
- Why? these errors are caught much earlier and rooted out before they become a "security bug."
- When a security bug is determined to be present, four standard mitigation techniques are available to the development team:
  - Do nothing
  - Warn the user
  - Remove the problem
  - Fix the problem

Bugs Mitigation - Introduction

- Problem Removal
  - Removing the problem is a business decision, because it typically involves disabling or removing a feature from a software release.
  - This mitigation has been used when the feature was determined not to be so important as to risk an entire release of the product.
  - Best in case if a fix will take too long or involve too many resources.
- Problem Fixation
  - Fixing the problem, is the most desired method and should be done whenever it is feasible given time and resource constraints.
  - If possible, all security bugs should be fixed as soon as possible, for the cascading effect of multiple bugs cannot be ignored.



KHAWAJAGAN Technologies

Resilience

Software Resilience

- * Software resilience is the ability to reduce the magnitude and/or duration of disruptive events. The effectiveness of a resilient application or infrastructure software depends upon its ability to anticipate, absorb, adapt to, and/or rapidly recover from a potentially disruptive event.

- * Adaptation of the National Infrastructure Advisory Council (NIAC) definition of infrastructure resilience

Software Resilience

- New paradigms have rapidly replaced the Waterfall model of software development.
- Agile and Scrum SDLCs displaced the rigorous activities, specially the notion of "phase containment".
- Appsec professionals, in past, counted on "phase containment" as a reliable means to prevent defects from creeping into subsequent phases.

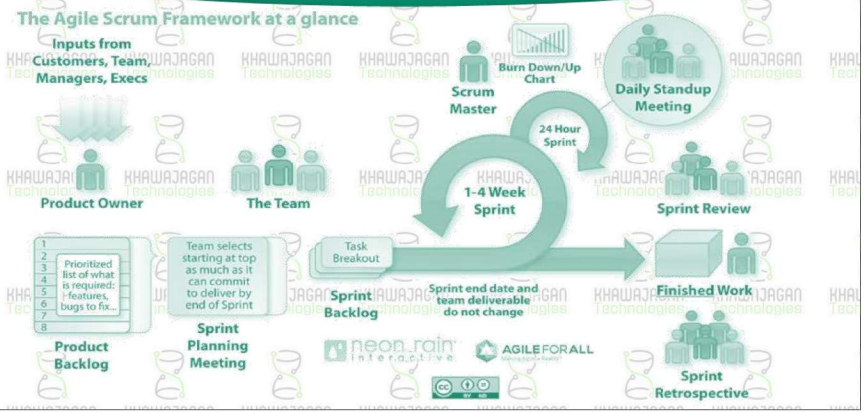
Software Resilience

- This new landscape includes Agile/Scrum, DevOps, Continuous Integration/Deployment (CI/CD) and Site Reliability Engineering (SRE).
- Changes in the software development paradigm forces change in the software security paradigm, which work hand-in-hand with what development teams are expected to do.
- Typically, software were inspected for security issues at the end of the development cycle (because of phase containment), this control point no longer exists.

Agile / Scrum

- Agile and Scrum are based on the following philosophy point
  - Software takes on a life of its own
  - Software are constantly being improved, extended, and enhanced
  - Changes in software can be delivered in hours, rather than weeks, months, or years

Agile / Scrum



Agile / Scrum

- This diagram encapsulates all the processes described by Scrum
- It also contains suggested time frames showing how it compresses time into digestible bites that continue to produce software.
- Some new roles are also indicated (e.g., product owner and Scrum master), and teams are composed of ALL the roles you formerly find on separate teams using Waterfall methods.
- In Scrum, one team is composed of the roles responsible for analysis, design, coding, testing, coordination, and ongoing delivery as new features are added, changes are made, or defects removed.
- It also means that work is not tossed over the wall to the next person in line to do "something." The team is responsible for all the effort and results.

Agile / Scrum

- A minimally viable product (MVP) is the first release of a new software application that's considered "bare bones" but has sufficient functionality for release to the market.
- The actions in each sprint typically follow the same activities as in the Waterfall model, but with more iterations and fewer phase gates that control when software is tested and released.
- Software is then changed regularly and is never really considered "complete."
- This creates severe challenges for appsec.

Agile / Scrum – Shift Left

- Shift left requires that development teams address software security from the very inception of a project (Build Security In) and in every step along the way to its manifestation.
- Everyone working on specification and development of new software "product" must understands their security obligations and prepared / able to meet those obligations.
- Security teams can no longer "do" security for development teams—the teams must be responsible and able to fulfill security requirements.

Agile / Scrum – Shift Left

- Shifting left in the development activity involves security checks built directly into integrated development environment (IDE)—for example, Visual Studio or Eclipse.
- Although these checks are on incomplete segments of an overall application, coding provides the first opportunity for security inspection and is needed to continue the cycle of appsec.
- Shifting left makes it possible to gain the assurance that applications are appropriately secure

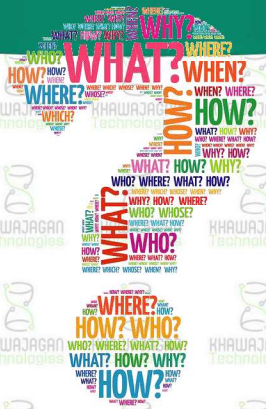
AppSec - Principles

- Secure application development is a PEOPLE issue, not a technical one.
- Every intervention into the SDLC affects people.
- Shift Left within the SDLC as much of the appsec work as you can
- The AppSec performed closest to the point of defects introduction is the best way of eliminating / preventing "defect creep" from one phase to the next.
- AppSec tools are great but of are questionable use if the people using them don't understand:
  - What the tool is telling them
  - Why their code is vulnerable
  - How their code is vulnerable
  - What do to about the vulnerability

AppSec - Principles

- People can only deal with some changes at one time – too many changes all at once leads to chaos and ultimately rebellion.
- Automate everything that you can (scanning, remediation planning, retesting, etc.).
- Make development team staff realize to not to treat security as a punishment or barrier
- Convince them that it empowers them, makes them more valuable as employees and helps in their product getting on shelf early

SP 2025



KHAWAJAGAN Technologies