Web Technologies

.....AND SECURITY

Stones Dalitso Chindipha

Chapters 16: Security

Security: Security Principles & Authentication

Objectives

1. Security Principles
2. Authentication
3. Cryptography
4. HTTPS
5. Security Best Practices
6. Common Threat Vectors

Security Overview

- Developer's often consider security only toward the end of a project, and by then it is too late.
  - hosting configuration errors, code design, policies, and implementation can infiltrate through the application.
- Filling these holes takes time, and the patched systems are often less elegant and manageable, if the holes get filled at all.
- The right way of addressing security is right from the beginning and all along the way so that you can plan for a secure system and hopefully have one in the end.
- This chapter will guide you in that never-ending quest to proactively defend your data and systems, which you will see, touches all aspects of software development.
- **The principal challenge with security is that threats exist in so many different forms.**
- **The most dangerous threat to secure system (read web application in this case) is not a malicious hacker on a tropical island a threat but a sloppy programmer, a disgruntled manager, or a naive secretary**

Security Principles

- **Security principles** are set of standards that are designed to minimize the vulnerability of systems and services to attackers who may obtain unauthorized access to sensitive data and misuse it.
  - They are building blocks to identifying the type of attack and solution for that.
- In the main book the list of security principles are:
  - **Information security**
  - **Risk Assessment and Management**
  - **Security Policy**
  - **Business continuity**
  - **Secure by design**
  - **Social engineering**

Information Security

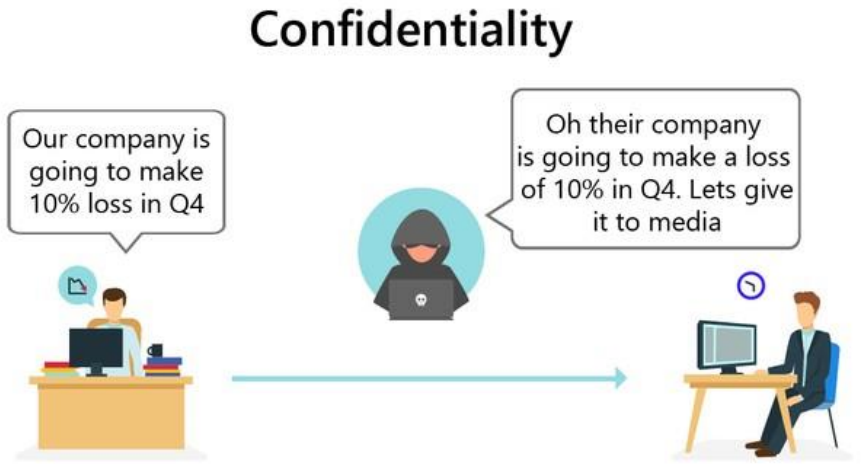
- **Information security** is the holistic practice of protecting information from unauthorized users.
  - Computer/IT security is just one aspect of this holistic thinking, which addresses the role computers and networks play.
- The other is **information assurance**, which ensures that data is not lost when issues do arise.
- **Security Standards:** ISO standards ISO/IEC 27002-270037 speak directly about security techniques and are routinely adopted by governments and corporations the world over.
  - These standards are very comprehensive, outlining the need for risk assessment and management, security policy, and business continuity to address the **CIA triad**.
- This chapter touches on some of those key ideas that are most applicable to web development.

InfoSec Goals in an Organization (CIA)

- There are three main objectives protected by information security, collectively known as CIA
  - **Confidentiality**
  - **Integrity**
  - **Availability**

Confidentiality

- Is to protect information from accidental or malicious disclosure.
- Prevents unauthorized users from accessing information to protect the privacy of information content.
  - Confidentiality is maintained through access restrictions.
  - Breaches of confidentiality can occur due to human error, intentional sharing, or malicious entry.

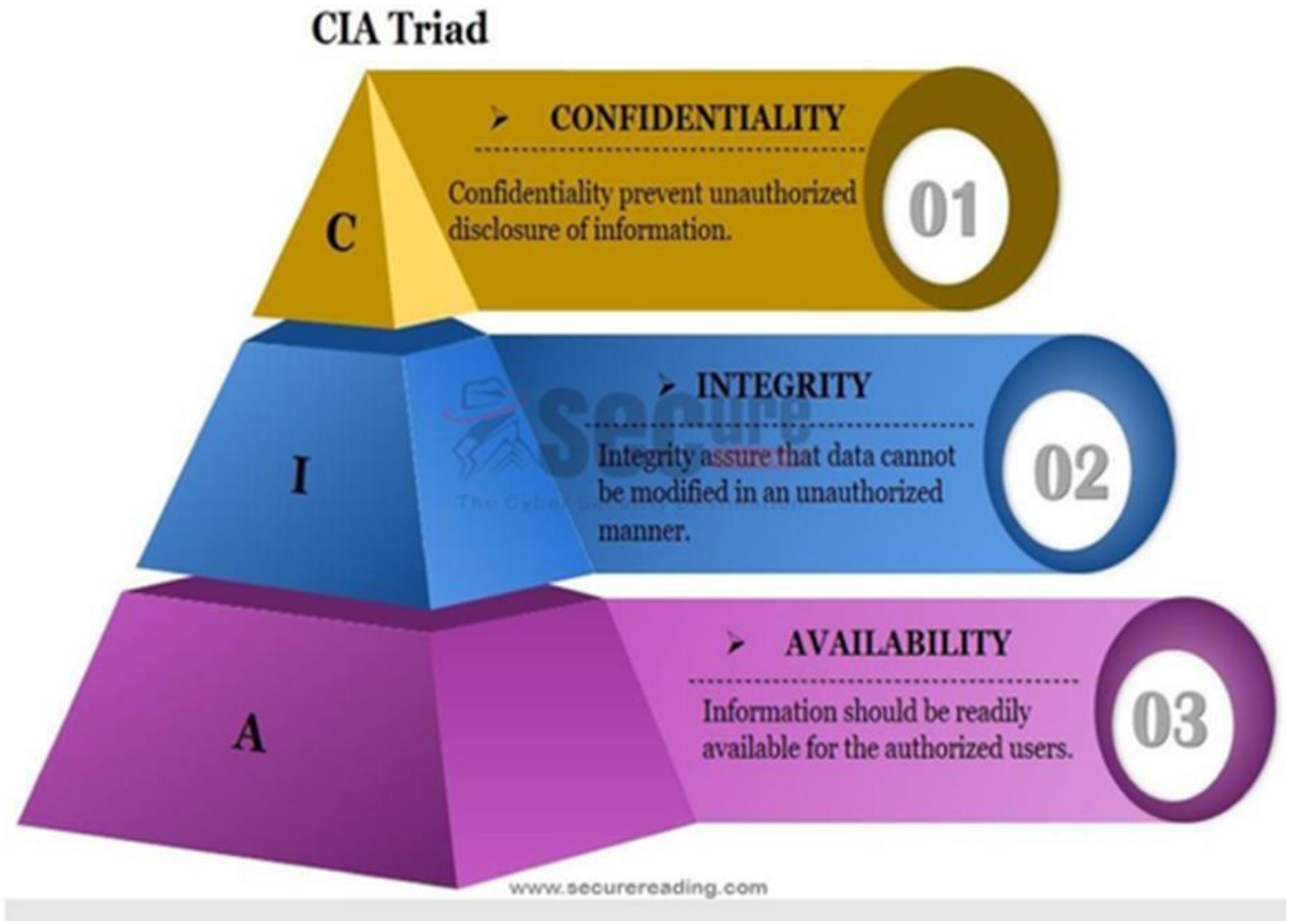


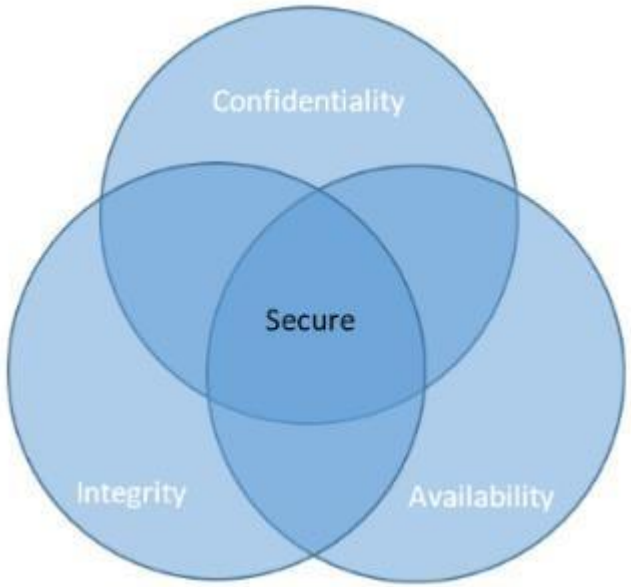
Integrity

- Is to protect information from accidental or intentional (malicious) modification.
- Ensures the authenticity and accuracy of information.
  - Integrity is maintained by restricting permissions for editing or the ability to modify information.
  - Loss of integrity can occur when analog information is not protected from environmental conditions, digital information is not transferred properly, or when users make unapproved changes.

Availability

- This is to ensure that information is available to those who need it and when they need it.
- Ensures that authorized users can reliably access information.
  - Availability is maintained through continuity of access procedures, backup or duplication of information,
  - Maintenance of hardware and network connections.
  - Loss of availability can occur when networks are attacked due to natural disasters, or when client devices fail.
  - Denial of Service (DoS)


CIA Triad
CONFIDENTIALITY
Confidentiality prevent unauthorized disclosure of information.
01
C
INTEGRITY
Integrity assure that data cannot be modified in an unauthorized manner.
02
I
AVAILABILITY
Information should be readily available for the authorized users.
03
A
www.securereading.com


Confidentiality
Secure
Integrity
Availability

Risk Assessment and Management

- The ability to assess risk is crucial to the web development world.
- Risk is a measure of how likely an attack is, and how costly the impact of the attack would be if successful.
- Knowing which ones to worry about allows you to identify the greatest risks and achieve the most impact for your effort by focusing on them
- Risk assessment uses the concepts of **actors, impacts, threats**, and **vulnerabilities** to determine where to invest in defensive countermeasures.

Risk Assessment: Actors

- **Actors** refers to the people who are attempting to access your system.
  - **Internal actors:** the people who work for the organization.
    - Although they account for a small percentage of the attacks, they are especially dangerous due to their internal knowledge of the systems.
  - **External actors:** the people outside of the organization.
    - More than three quarters of external actors are affiliated with organized crime or nation states.
  - **Partner actors:** people affiliated with an organization that you partner or work with.
    - Quite often partners are granted some access to each other's systems

Risk Assessment: Impact

- **What systems were infiltrated and what data was stolen or lost?**
  - The impact of an attack depends on what systems were infiltrated and what data was stolen or lost.
- The impact relates back to the CIA triad since impact could be the loss of availability, confidentiality, and/or integrity.
- **A loss of availability** prevents users from accessing some or all of the systems.
  - This might manifest as a denial of service attack, or a SQL injection attack (described later), where the payload removes the entire user database, preventing logins from registered users.
- **A loss of confidentiality** includes the disclosure of confidential information to a (often malicious) third party.
  - This could manifest as a cross-site script attack where data is stolen right off your screen or a full-fledged database theft where credit cards and passwords are taken.
- **A loss of integrity** changes your data or prevents you from having correct data.
  - This might manifest as an attacker hijacking a user session, perhaps placing fake orders or changing a user's home address.

Risk Assessment: Threats

- **A cybersecurity threat** is a **malicious** and **deliberate attack** by an individual or organization to gain unauthorized access to another individual's or organization's network to damage, disrupt, or steal IT assets, computer networks, intellectual property, or any other form of sensitive data.
- **Short version**: A security threat is a malicious act that aims to corrupt or steal data or disrupt an organization's systems or the entire organization.
  - A flood destroying your data center is a threat just as much as malicious SQL injections, buffer overflows, denial of service, and cross-site scripting attacks.
- Broadly, threats can be categorized using the **STRIDE** mnemonic, developed by Microsoft, which describes six areas of threat

Categorize threats with STRIDE

- **Spoofing:-** The attacker uses someone else's information to access the System.
- **Tampering:-** The attacker modifies some data in nonauthorized ways.
- **Repudiation:-** The attacker removes all trace of their attack, so that they cannot be held accountable for other damages done.
- **Information disclosure:-** The attacker accesses data they should not be able to.
- **Denial of service:-** The attacker prevents real users from accessing the systems.
- **Elevation of privilege:-** The attacker increases their privileges on the system thereby getting access to things they are not authorized to do.

Types of Cybersecurity Threats

StealthLabs

Malware

Phishing

Spear Phishing

Man in the Middle Attack

Denial of Service Attack

SQL Injection

Zero-day Exploit

Advanced Persistent Threats

Ransomware

DNS Attack

Risk Assessment: Vulnerabilities

- **A Security Vulnerability** is a weakness, flaw, or error found within a security system that has the potential to be leveraged by a threat actor in order to compromise a secure network.
  - Simply put: The holes in your system
- Some vulnerabilities are not fixed because they are unlikely to be exploited, while others are low risk because the consequences of an exploit are not critical.
- There are a number of Security Vulnerabilities, but some common examples are:
  - Broken Authentication
  - SQL Injection
  - Cross-Site Scripting
  - Cross-Site Request Forgery
  - Security Misconfiguration

Common Security Vulnerability Examples

- **Broken Authentication:** When authentication credentials are compromised, user sessions and identities can be hijacked by malicious actors to pose as the original user.
- **SQL Injection:** SQL injections attempt to gain access to database content via malicious code injection.
  - A successful SQL injection can allow attackers to steal sensitive data, spoof identities, and participate in a collection of other harmful activities.
- **Cross-Site Scripting:** Much like an SQL Injection, a Cross-site scripting (XSS) attack also injects malicious code into a website.
  - However, a Cross-site scripting attack targets website users, rather than the actual website itself, which puts sensitive user information at risk of theft.
- **Cross-Site Request Forgery:** A Cross-Site Request Forgery (CSRF) attack aims to trick an authenticated user into performing an action that they do not intend to do.
  - This, paired with social engineering, can deceive users into accidentally providing a malicious actor with personal data.
- **Security Misconfiguration:** Any component of a security system that can be leveraged by attackers due to a configuration error can be considered a "Security Misconfiguration."

Open Web Application Security Project (OWASP)

TOP 10:2021 Vulnerabilities

2017	2021
A01:2017-Injection	A03:2021-Injection
A02:2017-Broken Authentication	A07:2021-Identification and Authentication Failures
A03:2017-Sensitive Data Exposure	A02:2021-Cryptographic Failures
A04:2017-XML External Entities (XXE)	(New) A04:2021-Insecure Design
A05:2017-Broken Access Control	A01:2021-Broken Access Control
A06:2017-Security Misconfiguration	A05:2021-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)	A03:2021-Injection
A08:2017-Insecure Deserialization	(New) A08:2021-Software and Data Integrity Failures
A09:2017-Using Components with Known Vulnerabilities	A06:2021-Vulnerable and Outdated Components
A10:2017-Insufficient Logging & Monitoring	A09:2021-Security Logging and Monitoring Failures*
	(New) A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

Assessing Risk

- **A security risk assessment** identifies, assesses, and implements key security controls in applications. It also focuses on preventing application security defects and vulnerabilities.
- Carrying out a risk assessment allows an organization to view the application portfolio holistically-from an attacker's perspective.
- It supports managers in making informed resource allocation, tooling, and security control implementation decisions.
- **For our purposes, it will suffice to summarize that in risk assessment you would begin by identifying the actors, vulnerabilities, and threats to your information systems.**
- The probability of an attack, the skill of the actor, and the impact of a successful penetration are all factors in determining where to focus your security efforts.

4 steps of a successful security risk assessment model

1. **Identification:** Determine all critical assets of the technology infrastructure.
   - Next, diagnose sensitive data that is created, stored, or transmitted by these assets.
   - Create a risk profile for each.
2. **Assessment:** Administer an approach to assess the identified security risks for critical assets.
   - After careful evaluation and assessment, determine how to effectively and efficiently allocate time and resources towards risk mitigation.
   - The assessment approach or methodology must analyze the correlation between assets, threats, vulnerabilities, and mitigating controls.
3. **Mitigation:** Define a mitigation approach and enforce security controls for each risk.
4. **Prevention:** Implement tools and processes to minimize threats and vulnerabilities from occurring in your firm's resources.

A visual way of assessing threats

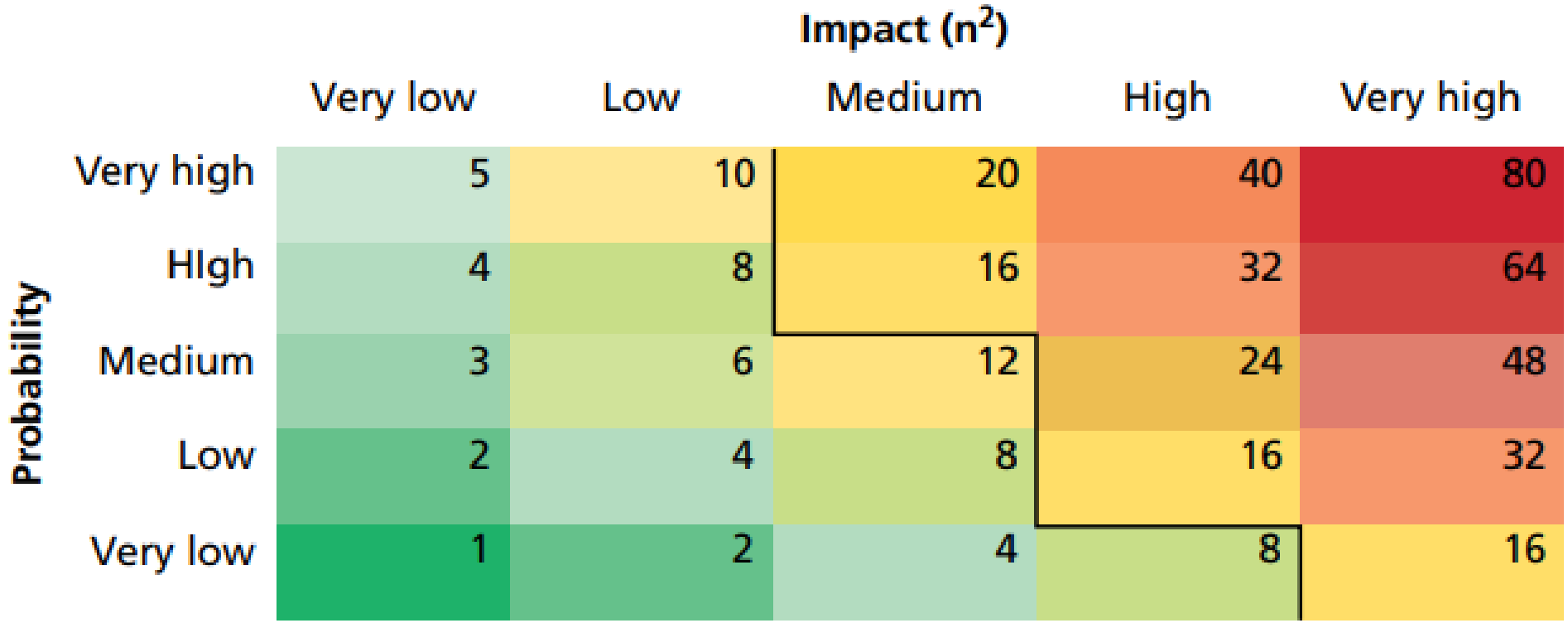


TABLE 16.1 Example of an Impact/Probability Risk Assessment Table Using 16 as the Threshold.

Policies

- Clearly articulate policies to users of the system to ensure they understand their rights and obligations.
- Good policies aim to modify the behaviour of internal actors, but will not stop foolish or malicious behaviour by employees.
  - However, as one piece of a complete security plan, good policies can have a tangible impact.
- These policies typically fall into three categories:
  - **Usage policy** defines what systems users are permitted to use, and under what situations. A company may, for example, prohibit social networking while at work. Usage policies are often designed to reduce risk by removing some attack vector.
  - **Authentication policy** controls how users are granted access to the systems. These policies may specify where an access badge is needed, a biometric ID, or when a password will suffice.
  - **Legal policies** define a wide range of things including data retention and backup policies as well as accessibility requirements (like having all public communication well organized for the blind).

Business Continuity

- Part of a secure system is being able to access it in the case of the unforeseen.
- You should consider
  - Administrator Password Management
  - Backups and Redundancy
  - Geographic Redundancy
  - Stage Mock Events
  - Auditing you systems

Secure By Design

- Secure by design is a software engineering principle that tries to make software better by acknowledging and addressing that there are malicious users out there.
- By continually distrusting user input (and even internal values) throughout the design and implementation phases, you will produce more secure software than if you didn't consider security at every stage.
- Some techniques that have developed to help keep your software secure include:
  - code reviews
  - pair programming
  - security testing
  - Secure by default
- Techniques can be applied at every stage of the software development life cycle to make your software Secure By Design

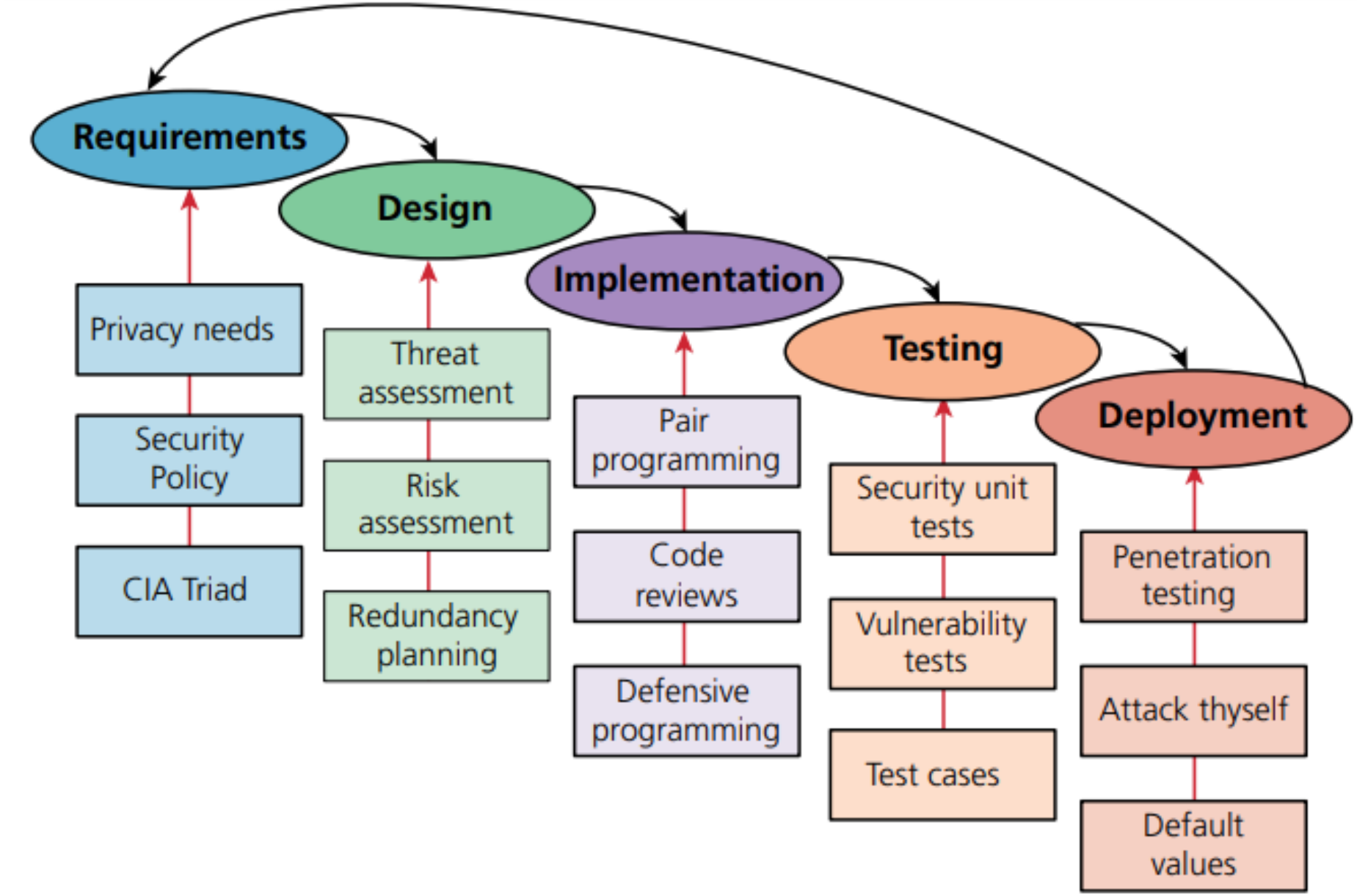


FIGURE 16.2 Some examples of security input into the SDLC

Secure By Design : Code Reviews & Unit Testing

- **Code Reviews:** In a code review system, programmers must have their code peer-reviewed before committing it to the repository.
  - New employees are often assigned a more senior programmer who uses the code review opportunities to point out inconsistencies with company style and practice.
- **Unit testing** is the principle of writing small programs to test your software as you develop it.
  - Usually the units in a unit test are a module or class, and the test can compare the expected behaviour of the class against the actual output.
  - If you break any existing functionality, a unit test will discover it right away, saving you future headache and bugs.

Secure By Design :Pair programming &Security testing

- **Pair programming** is the technique where two programmers work together at the same time on one computer.
  - One programmer drives the work and manipulates the mouse and keyboard while the other programmer can focus on catching mistakes and high-level thinking.
  - After a set time interval the roles are switched and work continues.
- **Security testing** is the process of testing the system against scenarios that attempt to break the final system.
  - Usually includes penetration testing where the company attempts to break into their own systems to find vulnerabilities as if they were hackers.
  - Whereas normal testing focuses on passing user requirements, security testing focuses on surviving one or more attacks that simulate what could be out in the wild.

Secure By Design: Secure by default

- Systems are often created with default values that create security risks (like a blank password).
- Secure by default aims to make the default settings of a software system secure:
  - **Objective:** minimise those type of breaches even if the end users are not very knowledgeable about security.
- **What could possibly go wrong with this?**

Social Engineering

- **Social engineering** is the tactic of manipulating, influencing, or deceiving a victim in order to gain control over a computer system, or to steal personal and financial information.
- It uses psychological manipulation to trick users into making security mistakes or giving away sensitive information.
- Social engineering attacks happen in one or more steps.
  - A perpetrator first investigates the intended victim to gather necessary background information, such as potential points of entry and weak security protocols, needed to proceed with the attack.
  - Then, the attacker uses a form of pretexting such as impersonation to gain the victim's trust and provide stimuli for subsequent actions that break security practices, such as revealing sensitive information or granting access to critical resources.

Social Engineering

- Social engineering attacks come in many different forms and can be performed anywhere where human interaction is involved
- Common ones include:
  - Phishing, Baiting, piggybacking, Scareware, Dumpster Diving, Quid pro quo
- Social Engineering Prevention:
  - Don't open email attachments from suspicious sources.
  - Use Multi-Factor Authentication (MFA)
  - Be wary of tempting offers
  - Avoid plugging an unknown USB into your computer
  - Clean up your social media.
- Some organizations go so far as to set up false phishing scams that target their own employees to see which ones will divulge information to such scams.
  - Those employees are then trained or terminated.

Security Theather

- Security theater is when visible security measures are put in place without too much concern as to how effective they are at improving actual security.
- The visual nature of these theatrics is thought to dissuade potential attackers.
- This is often done in 404 pages where a stern warning might read:
  - **Your IP address is XX.XX.XX.XX. This unauthorized access attempt has been logged. Any illegal activity will be reported to the authorities.**
- This message would be an example of security theater if this stern statement is a site's only defense.
  - When used alone, security theater is often ridiculed as a serious technique, but as part of a more complete defense it can contribute a deterrent effect.

Authentication

- **Authentication** is the process of verifying whether someone (or something) is, in fact, who (or what) it is declared to be.
  - Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.
- Authentication in action:
  - Getting entrance to an airport, getting past the bouncer at the bar, or logging into your web application
  - Whether you are logging into your computer system at the office, checking your account balance on your bank website
  - or visiting your favourite social media feeds
- The process of authentication helps these sites determine that you are the correct person trying to gain access.

Authentication Factors

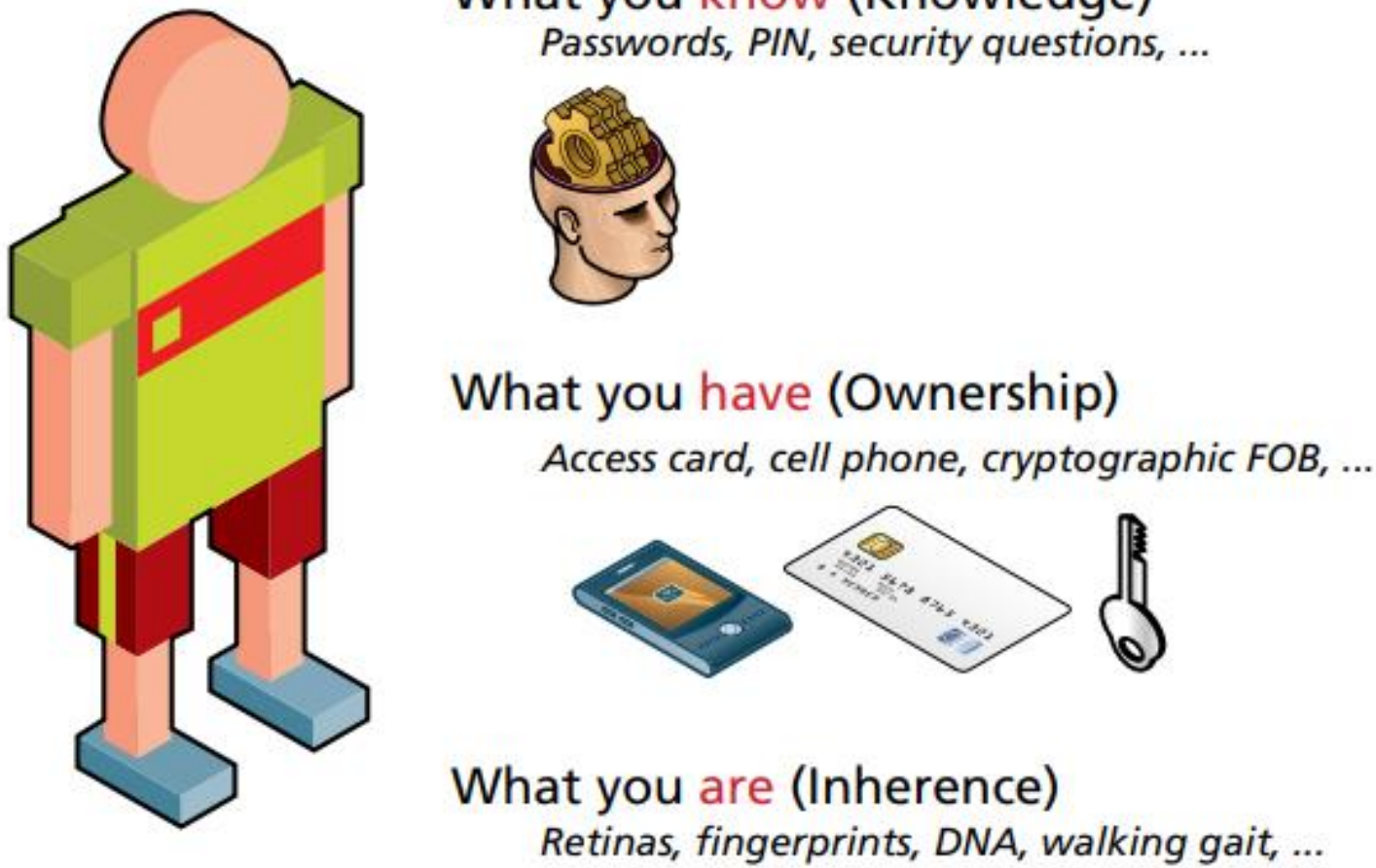


FIGURE 16.3 Authentication factors

How many factors do you need?

Authentication factors are the things you can ask someone for in an effort to validate that they are who they claim to be.

Single Factor Authentication

- Single-factor authentication is the weakest and most common category of authentication system where you ask for only one of the three factors.
  - Know a password
  - Posses an access card
  - Fingerprint access on your mobile phone
- When better authentication confidence is required, more than one authentication factor should be considered

Multi Factor Authentication

- Multifactor authentication is where two or more distinct factors of authentication must pass before you are granted access.
- The way we all access an ATM machine is an example of two- factor authentication:
  - you must have both the knowledge factor (PIN) and
  - the ownership factor(card)
- Multifactor authentication is becoming prevalent in consumer products as well:
  - your cell phone is used as the ownership factor alongside
  - your password as a knowledge factor.
  - Email login on new machine?
  - Accessing application on your phone

Third Party Authentication

- Let someone else worry about it
- Many popular services allow you to use their system to authenticate the user and provide you with enough data to manage your application.
- Third-party authentication schemes like **OpenID** and **Open authorization (OAuth)** are popular with developers and are used under the hood by many major websites
  - including Amazon, Facebook, Microsoft, and Twitter, to name but a few.

OAuth => Pg 723 - 725

Authorization

- Authorization defines what rights and privileges a user has once they are authenticated.
- **Authentication grants access while Authorization defines what the user with access can (and cannot) do.**
- The **principle of least privilege** is a helpful rule of thumb that tells you to give users and software only the privileges required to accomplish their work.
- Starting out a new user with the least privileged account and adding permission as needed not only provides security but allows you to track who has access to what systems.
  - Even system administrators should not use the root account for their day-to-day tasks, but rather escalate their privileges when needed.

Applications of Authorization

- Some examples in web development where proper authorization increases security include:
  - Using a separate database user for read and write privileges on a database
  - Providing each user an account where they can access their own files securely
  - Setting permissions correctly so as to not expose files to unauthorized users
  - Ensuring Apache is not running as the root account