UNIT-3

Access Control

- **Access control** is the process of identifying a person and determining their level of security access to either electronic systems or physical sites based on the policies and procedures set by the organization. Modern access control systems identify a person, authenticate them by looking at their identification, then give that person only the key to the door or digital resource that they need to access.
- Access is granted using a set of steps to make sure this user can access the requested resources. These steps are typically:
- Identification
- Authentication
- Authorization

Types of Access Control

Access control types include:

- **Administrative**
- **Physical**
- **Technical**

Administrative Access Control

- Administrative access control sets the access control policies and procedures for the whole organization, defines the implementation requirements of both physical and technical access control, and what the consequences of non-compliance will be. Some examples are: supervisory structure, staff and contractor controls, information classification, training, auditing, and testing.

Physical Access Control

- Physical access control is critical to an organizations security and applies to the access or restriction of access to a place such as property, building or room. Some examples are: fences, gates, doors, turnstiles, etc. using locks, badges, biometrics (facial recognition, fingerprints), video surveillance cameras, security guards, motion detectors, mantrap doors, etc. to allow access to certain areas.

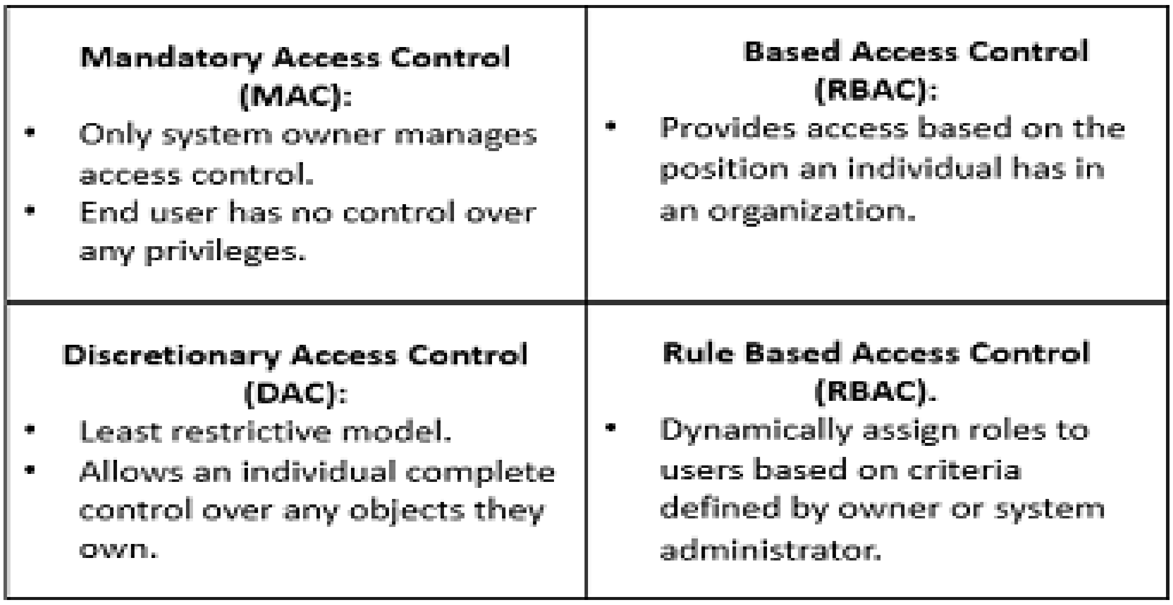
Technical or Logical Access Control

- Technical or logical access control limits connections to computer networks, system files, and data. It enforces restrictions on applications, protocols, operating systems, encryptions mechanisms, etc.

- In today's increasingly digital world, modern access control systems combine both administrative, physical and technical access control to limit access to sensitive data and physical locations, providing a much higher level of security. Some examples are: access control lists, intrusion detection systems, and antivirus software.

Access Control Models

- How does someone grant the right level of permission to an individual so that they can perform their duties? Access control models define how permissions are assigned.
- Access control models have four flavours:
- Mandatory Access Control (MAC)
- Role Based Access Control (RBAC)
- Discretionary Access Control (DAC)
- Rule Based Access Control (RBAC)



Mandatory Access Control (MAC): • Only system owner manages access control. • End user has no control over any privileges.	**Based Access Control (RBAC):** • Provides access based on the position an individual has in an organization.
Discretionary Access Control (DAC): • Least restrictive model. • Allows an individual complete control over any objects they own.	**Rule Based Access Control (RBAC).** • Dynamically assign roles to users based on criteria defined by owner or system administrator.

Common issues and mistakes
(Access control)

Appropriate role-based access

- Users should only be given access to systems that they need to access, and at a level that’s appropriate to their role. Good practice is to ensure that access privileges (and changes) are approved by a sufficiently senior Director or Manager.
- New employees or those changing role should have approved/documented access rights and they should be revoked across all systems for any leavers without delay. Finally, access privileges should be reviewed regularly and amended as part of a process of security governance.

Poor password management

Password management is one of the most common mistakes when it comes to access control. When there are a lot of different systems that require a password to access then it's not uncommon for employees and even business owners to use the same password across the board.

- A great solution to this problem is to encourage the use of password keeper apps. This allows you and your employees to use unique, strong passwords for each system without the need to have them all memorised. But of course, you need to ensure that your 'keeper app' is secure or you open up all the doors to your systems!

Poor user education

- One of the most important aspects of improving the security of company data is educating your employees about risk. Your employees could easily be doing things that are putting your data at risk. For example, people will often try to find a quicker and easier way to accomplish something, often not being aware of the risk they could be creating. This is why good training about risk is vital.
- Human error is always one of the biggest security risks for any company so you should be very aware of this and take any steps you can to educate your employees, including risk-training programs.

Web Security

- Web security is also known as "Cybersecurity". It basically means protecting a website or web application by detecting, preventing and responding to cyber threats.
- Websites and web applications are just as prone to security breaches as physical homes, stores, and government locations. Unfortunately, cybercrime happens every day, and great web security measures are needed to protect websites and web applications from becoming compromised.
- There are different types of technologies available for maintaining the best security standards. Some popular technical solutions for testing, building, and preventing threats include:
- Black box testing tools
- Fuzzing tools
- White box testing tools
- Web application firewalls (WAF)
- Security or vulnerability scanners
- Password cracking tools

Web Security(threats)

- Your website or web application's security depends on the level of protection tools that have been equipped and tested on it. There are a few major threats to security which are the most common ways in which a website or web application becomes hacked. Some of the top vulnerabilities for all web-based services include:
- SQL injection
- Password breach
- Cross-site scripting
- Data breach
- Remote file inclusion
- Code injection
- Preventing these common threats is the key to making sure that your web-based service is practicing the best methods of security.

Web Security(prevention)

- Resource assignment – By assigning all necessary resources to causes that are dedicated to alerting the developer about new web security issues and threats, the developer can receive a constant and updated alert system that will help them detect and eradicate any threats before security is officially breached.
- Web scanning – There are several web scanning solutions already in existence that are available for purchase or download. These solutions, however, are only good for known vulnerability threats – seeking unknown threats can be much more complicated. This method can protect against many breaches, however, and is proven to keep websites safe in the long run**.**

Browser Isolation

- Remote browser isolation is an advanced cybersecurity technique that provides an additional layer of protection for users and organizations. Browser isolation separates browsing activity from endpoint hardware, thereby reducing the device's attack surface. When a user accesses a web page or app, it is loaded onto a remote browser that serves a rendering of the webpage to the user. The page operates normally, but only pixels are delivered to the user. There is no active content downloaded, so malicious code that may be hidden is kept at bay.

Benefits of remote browser isolation

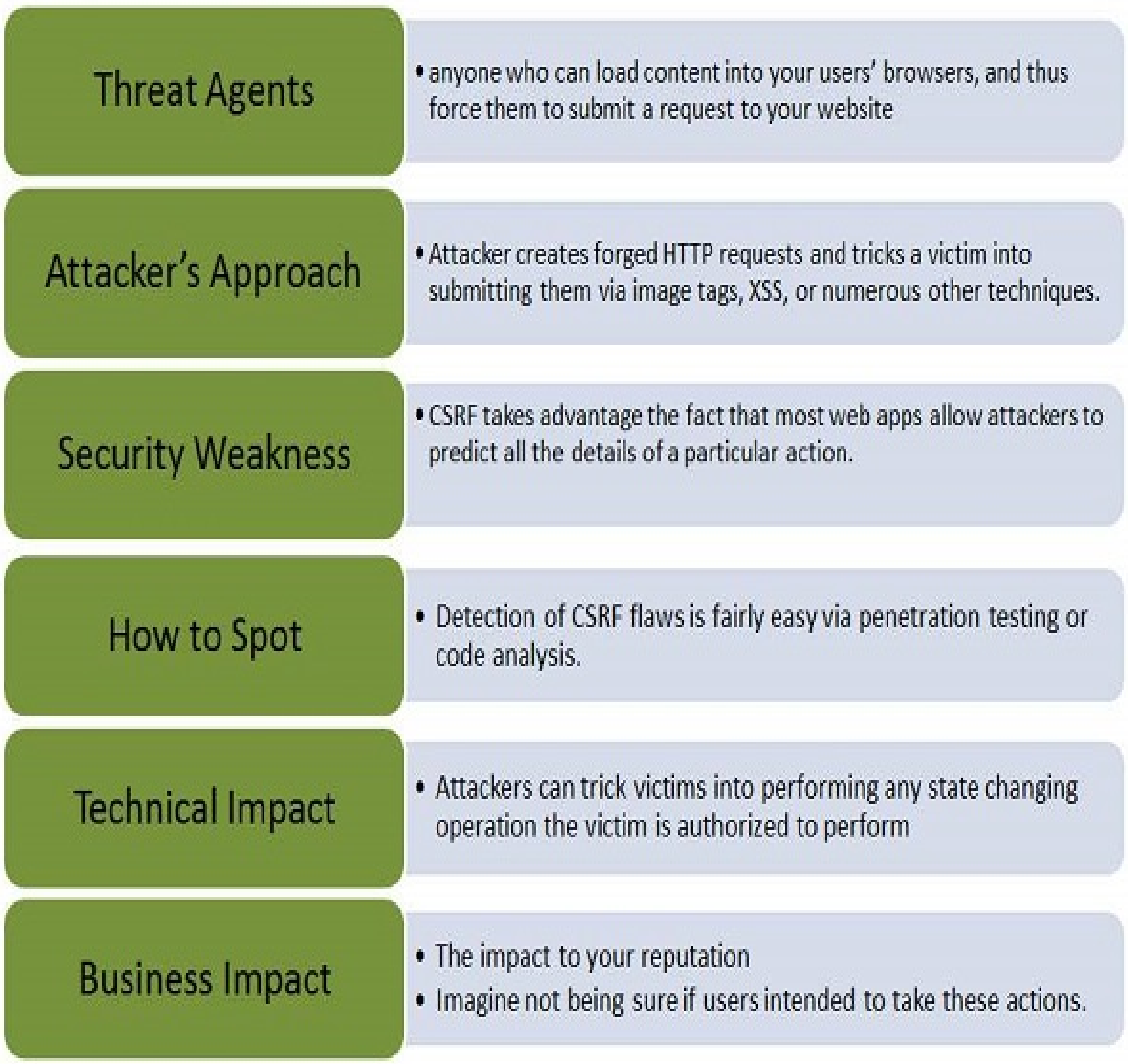
In today's digitally connected world, spending time on the internet is a requirement. To make that time safer, remote browser isolation:

Enables safe access to risky web content: It isolates users from web apps and delivers a safe rendering of web content—without requiring an endpoint agent on every device.

- **Protects sensitive data:** It protects users and executives from targeted attacks hidden in web pages, downloadable web content, and vulnerable plug-ins.
- **Removes the threat of data exfiltration:** It eliminates the ability for a webpage to exfiltrate data or compromise a user's machine, even if the browser is outdated, vulnerable, or has unsafe plug-ins installed.
- **Allows more open internet policies:** It enables organizations to minimize policy complexity, reduce risk, and implement more open policies for internet access.

Cross-Site Request Forgery(CSRF)

- A CSRF attack forces an authenticated user (victim) to send a forged HTTP request, including the victim's session cookie to a vulnerable web application, which allows the attacker to force the victim's browser to generate request such that the vulnerable app perceives as legitimate requests from the victim.
- Let us understand Threat Agents, Attack Vectors, Security Weakness, Technical Impact and Business Impacts of this flaw with the help of simple diagram.



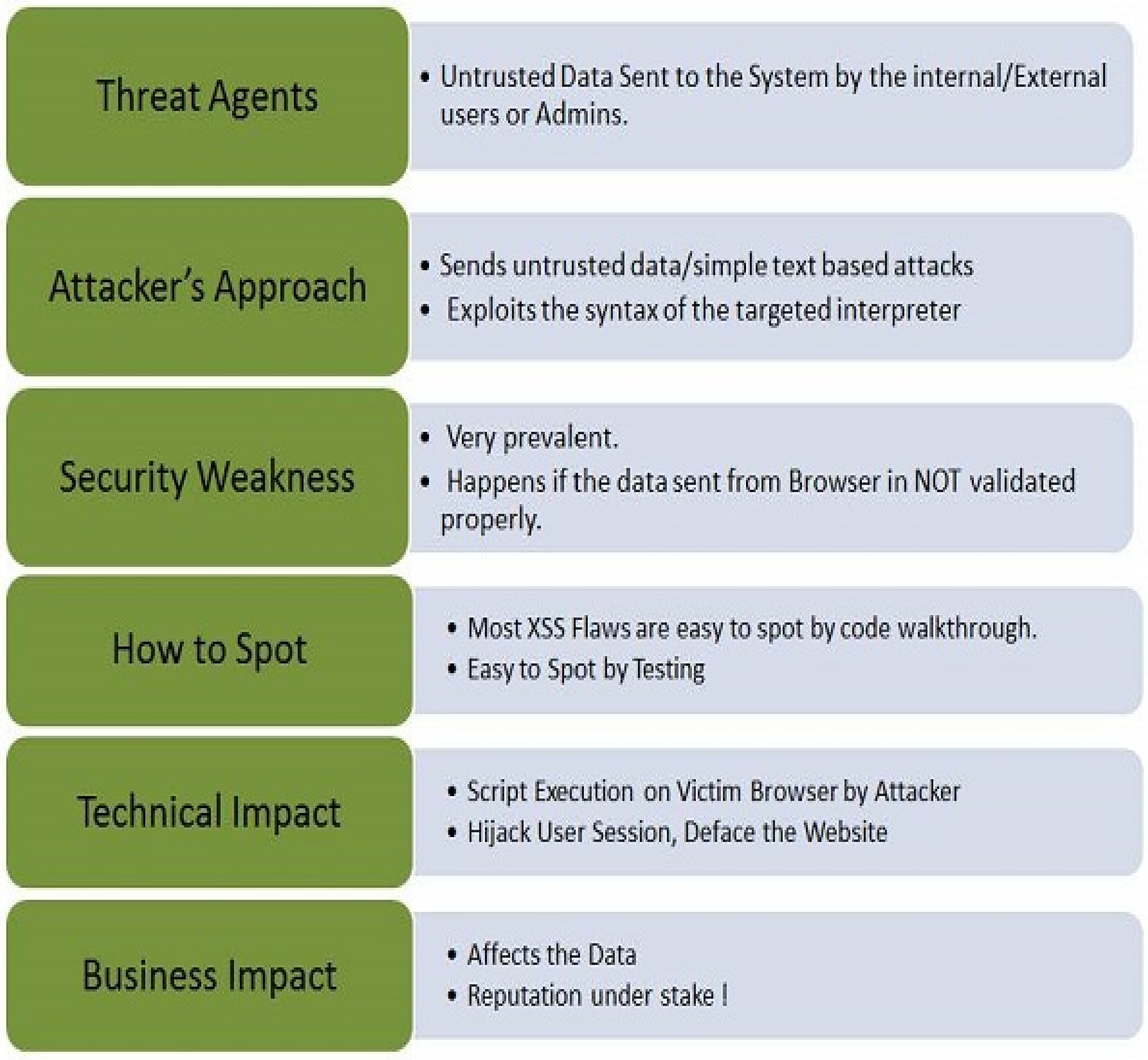
Threat Agents	• anyone who can load content into your users' browsers, and thus force them to submit a request to your website
Attacker's Approach	• Attacker creates forged HTTP requests and tricks a victim into submitting them via image tags, XSS, or numerous other techniques.
Security Weakness	• CSRF takes advantage the fact that most web apps allow attackers to predict all the details of a particular action.
How to Spot	• Detection of CSRF flaws is fairly easy via penetration testing or code analysis.
Technical Impact	• Attackers can trick victims into performing any state changing operation the victim is authorized to perform
Business Impact	• The impact to your reputation • Imagine not being sure if users intended to take these actions.

Preventive Mechanisms

- CSRF can be avoided by creating a unique token in a hidden field which would be sent in the body of the HTTP request rather than in an URL, which is more prone to exposure.
- Forcing the user to re-authenticate or proving that they are users in order to protect CSRF. For example, CAPTCHA.

Cross-Site Scripting

- Cross-site Scripting (XSS) happens whenever an application takes untrusted data and sends it to the client (browser) without validation. This allows attackers to execute malicious scripts in the victim's browser which can result in user sessions hijack, defacing web sites or redirect the user to malicious sites.
- Let us understand Threat Agents, Attack Vectors, Security Weakness, Technical Impact and Business Impacts of this flaw with the help of simple diagram.



Threat Agents	• Untrusted Data Sent to the System by the internal/External users or Admins.
Attacker's Approach	• Sends untrusted data/simple text based attacks • Exploits the syntax of the targeted interpreter
Security Weakness	• Very prevalent. • Happens if the data sent from Browser in NOT validated properly.
How to Spot	• Most XSS Flaws are easy to spot by code walkthrough. • Easy to Spot by Testing
Technical Impact	• Script Execution on Victim Browser by Attacker • Hijack User Session, Deface the Website
Business Impact	• Affects the Data • Reputation under stake !

Types of XSS

- **Stored XSS** − Stored XSS also known as persistent XSS occurs when user input is stored on the target server such as database/message forum/comment field etc. Then the victim is able to retrieve the stored data from the web application.
- **Reflected XSS** − Reflected XSS also known as non-persistent XSS occurs when user input is immediately returned by a web application in an error message/search result or the input provided by the user as part of the request and without permanently storing the user provided data.
- **DOM Based XSS** − DOM Based XSS is a form of XSS when the source of the data is in the DOM, the sink is also in the DOM, and the data flow never leaves the browser.

Preventive Mechanisms

- Developers have to ensure that they escape all untrusted data based on the HTML context such as body, attribute, JavaScript, CSS, or URL that the data is placed into.
- For those applications that need special characters as input, there should be robust validation mechanisms in place before accepting them as valid inputs.

HTTP Rendering

- Google recently announced that the majority of web pages they crawl are now being rendered, and they released a new feature in Webmaster Tools to preview a rendered version of your site.
- The primary function of a web browser is to request resources from the web and display them inside of a browser window. Typically, a browser will request HTML, CSS, JavaScript and image content from a server and interpret them based on web standards and specifications. They follow standards because it allows websites to behave the same way across all browsers, and creates less work and fewer headaches for web developers.
- There Are 7 Key Components of Any Web Browser:

  1. **Layout Engine –** this takes input from the browser (URL bar, search box, mouse clicks and key presses) and passes them to the rendering engine

  2. **Rendering Engine** – this takes HTML code and interprets it into what you see visually. For instance, a **tag would be interpreted by the rendering engine as a set of instructions to make the text inside of the element bold**

  3. **User Interface** – This is the visual presentation of controls in the browser, for instance the back and forward buttons, bookmarks, and all chrome that appears around the main browser window where web pages display.

  4. **JavaScript Engine** – This engine takes JavaScript code, parses it, executes it, and returns the results.

5. **Network Layer** – This is a function of the browser that happens behind the scenes and handles network functions such as encryption, http and ftp requests, and all network settings such as timeouts and the handling of HTTP status codes

6. **Storage** – Browser's must store some data which can include cached files, cookies, and recently browsers have updated to be able to store data and objects created with JavaScript

7. **Operating System Interface** – The browser must interact with the operating system to draw out several elements of the page like drop down boxes and the chrome of a window (close, maximize, and minimize buttons.

HTTP Rendering

The rendering engine has a very important job as it displays what you see on your screen. It communicates with the networking layer of the browser to grab HTML code and other items passed from a remote server. Then it follows these steps:

1. Parsing HTML and creating the DOM Tree – HTML is a hierarchal structure that begins with a <html> tag, usually contains a <head> and <body> tag, and elements can be nested within elements. These HTML elements are parsed and turned into a "DOM tree" by the rendering engine. It is a tree like structure made out of the HTML, where each tag is a branch starting at the root element.

2. Render Tree Construction – CSS attributes (style attributes) are also parsed and combined with the DOM tree to create a "render tree". This is a tree of visual elements such as height/width and color ordered in the hierarchy in which they are to be displayed in the browser.

3. Layout Process – Once the render tree is constructed, the rendering engine recursively goes through the HTML elements in the tree and figure out where they should be placed on the screen. This starts at the top left in position 0,0 and elements and attributes are mapped to coordinates on the screen.

4. Painting – Each node (branch) of the render tree is drawn out on the screen by communicating with the Operating System Interface which contains designs and styles for how UI elements should look.

- **NOTE**:Server-side **rendering** (SSR) is a method of providing pre-generated HTML as a response to an **HTTP** request. For example, the **content** of this website is pre-built from source files. These files are converted into HTML pages and uploaded into an **HTTP** hosting server.

Findings Vulnerabilities

- To find security vulnerabilities on the business' network, it is necessary to have an accurate inventory of the assets on the network, as well as the operating systems (OSs) and software these assets run. Having this inventory list helps the organization identify security vulnerabilities from obsolete software and known program bugs in specific OS types and software.

- Without this inventory, an organization might assume that their network security is up to date, even though they could have assets with years-old vulnerabilities on them. Also, if a new security protocol is applied to assets on the network to close security gaps, but there are unknown assets on the network, this could lead to uneven protection for the organization.
- For example, say that Servers A, B, and C get updated to require multi-factor authentication, but Server D, which was not on the inventory list, doesn't get the update. Malicious actors could use this less-secure server as an entry point in an attack. Breaches have occurred in this manner before. As noted by *The New York Times* in an article about a major data breach affecting JPMorgan Chase bank, "Most big banks use a double authentication scheme, known as two-factor authentication, which requires a second one-time password to gain access to a protected system. But JPMorgan's security team had apparently neglected to upgrade one of its network servers with the dual password scheme."
- When it comes to finding security vulnerabilities, a thorough network audit is indispensable for success.
- Penetration testing is highly useful for finding security vulnerabilities. However, it isn't the only method companies should use. General steps of a penetration test usually involve:
- Getting a "white hat" hacker to run the pen test at a set date/time.
- Auditing existing systems to check for assets with known vulnerabilities.
- The "hackers" running simulated attacks on the network that attempt to exploit potential weaknesses or uncover new ones.
- The organization running its incident response plan (IRP) to try and contain the "attacks" simulated during penetration testing.
- Another tool for identifying potential issues is the threat intelligence framework. This framework helps your organization:
- Define what it needs to protect.
- Set goals for overall network security.
- Identify primary threat sources.
- Refine cyber security protections.
- Choose appropriate threat intelligence feeds to monitor new and emerging cyber threats and attack strategies.

Secure Development

- Secure development is a practice to ensure that the code and processes that go into developing applications are as secure as possible. Secure development entails the utilization of several processes, including the implementation of a Security Development Lifecycle (SDL) and secure coding itself.
- The Security Development Lifecycle (SDL) is a software development security assurance process consisting of security practices grouped by six phases: training, requirements & design, construction, testing, release, and response.
- Secure development can be incorporated into both a traditional software development lifecycle and the rapid pace agile development.
- One of the important steps in secure development is integrating testing tools and services such as Vera code into the software development lifecycle. These tools allow developers to model an application, scan the code, check the quality and ensure that it meets regulations. Automated secure development testing tools help developers find and fix security issues. Secure development services like Vera code also offer secure development training so that developers can become certified in secure development and gain further education and insight into issues that they may have created.

The most important reasons to adopt SDL practices are:

- **Higher security.** In SDL, continuous monitoring for vulnerabilities results in better application quality and mitigation of business risks.
- **Cost reduction.** In SDL, early attention to flaws significantly reduces the effort required to detect and fix them.
- **Regulatory compliance.** SDL encourages a conscientious attitude toward security-related laws and regulations. Ignoring them may result in fines and penalties, even if no sensitive data is lost.

SDL also provides a variety of side benefits, such as:

- Development teams get continuous training in secure coding practices.
- Security approaches become more consistent across teams.
- Customers trust you more, because they see that special attention is paid to their security.
- Internal security improves when SDL is applied to in-house software tools.

Secure development includes following phases:

- Planning and requirements
- Architecture and design

- Test planning
- Coding
- Testing and results
- Release and maintenance
- Generally speaking, a secure SDLC is set up by adding security-related activities to an existing development process. For example, writing security requirements alongside the collection of functional requirements, or performing an architecture risk analysis during the design phase of the SDLC.
- Many secure SDLC models have been proposed. Here are a couple of them:
- **MS Security Development Lifecycle (MS SDL):** One of the first of its kind, the MS SDL was proposed by Microsoft in association with the phases of a classic SDLC.
- **NIST 800-64:** Provides security considerations within the SDLC. Standards were developed by the National Institute of Standards and Technology to be observed by US federal agencies.