

Security

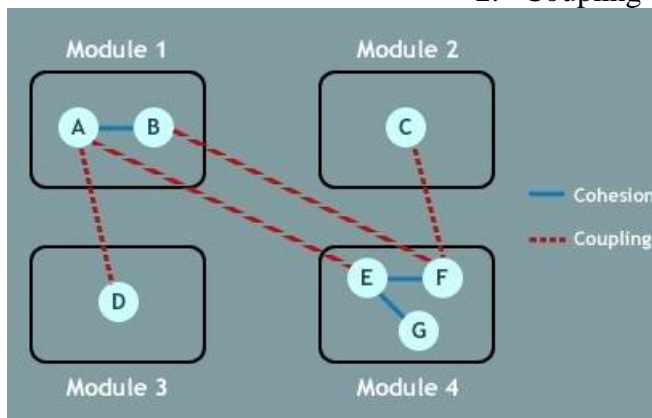
🚧 Software Selection Criteria:

- ✓ Software must be compatible with current and future hardware over the next planning interval
- ✓ Software maintenance and warranties must be of appropriate length and cost
- ✓ Software help desk must be maintained by [vendor, third party, in-house personnel]

🚧 Functional independency:

Functional independence is evaluated using two criteria:

1. Cohesion
2. Coupling



Cohesion	Coupling
Cohesion is the indication of the relationship within module.	Coupling is the indication of the relationships between modules.
Cohesion shows the module's relative functional strength.	Coupling shows the relative independence among the modules.
Cohesion is a degree (quality) to which a component / module focuses on the single thing.	Coupling is a degree to which a component / module is connected to the other modules.
While designing you should strive for high cohesion i.e. a cohesive component/ module focus on a single task (i.e., single-mindedness) with little interaction with other modules of the system.	While designing you should strive for low coupling i.e. dependency between modules should be less.
Cohesion is the kind of natural extension of data hiding for example, class having all members visible with a package having default visibility.	Making private fields, private methods and non public classes provides loose coupling.
Cohesion is Intra – Module Concept.	Coupling is Inter -Module Concept.

Application security vs. software security

- ✓ The terms ‘application security’ and ‘software security’ are often used interchangeably. However, there is in fact a difference between the two.
- ✓ To ensure that a piece of software is secure, security must be built into all phases of the software development life cycle (SDLC). Thus, software security isn’t application security—it’s much bigger.
- ✓ Application security as subset of software security
- ✓ Software security (pre-deployment) activities include:
 - Secure software design
 - Development of secure coding guidelines for developers to follow
 - Development of secure configuration procedures and standards for the deployment phase
 - Secure coding that follows established guidelines
 - Validation of user input and implementation of a suitable encoding strategy
 - User authentication
 - User session management
 - Function level access control
 - Use of strong cryptography to secure data at rest and in transit
 - Validation of third-party components
 - Arrest of any flaws in software design/architecture
- ✓ Application security (post-deployment) activities include:
 - Post deployment security tests
 - Capture of flaws in software environment configuration
 - Malicious code detection (implemented by the developer to create backdoor, time bomb)
 - Patch/upgrade
 - IP filtering
 - Lock down executables
 - Monitoring of programs at runtime to enforce the software use policy

Does software quality equal software security?

- ✓ Software quality and security assurance both concern risk to the organisation, but they do so for different reasons
- ✓ A simple way to think about all this is that quality is binary – the software either works or it doesn’t – where security is not – the software may be secure today but not tomorrow. It is important to have quality code but quality code may not be secure. Then again, secure code must always be quality code.

Computer Security Depends on Two Types of Requirements:

- i. Functional and
- ii. Assurance


Functional requirements describe what a system should do. Assurance requirements describe how functional requirements should be implemented and tested. Both sets of requirements are needed to answer the following questions:

Does the system do the right things (behave as promised)?

Does the system do the right things in the right way?

For checking the computer security verification and validation process are employed. Verification is the process of confirming that one or more predetermined requirements or specifications are met. Validation then determines the correctness or quality of the mechanisms used to meet the needs.

With software, you need both verification and validation answers to gain confidence in products before launching them into a wild, hostile environment such as the Internet. Most of today's commercial off-the-shelf (COTS) software and systems stop at the first step, verification, without bothering to test for obvious security vulnerabilities in the final product. Developers of software generally lack the wherewithal and motivation needed to try to break their own software. More often, developers test that the software meets the specifications in each function that is present but usually do not try to find ways to circumvent the software and make it fail.

 **Do you think security depends on the software? Why and Why not?**

 **What is Secure Sockets Layer (SSL)?**

Secure Sockets Layer (SSL) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser, or a mail server and a mail client (e.g., Outlook).

SSL allows sensitive information such as credit card numbers, social security numbers, and login credentials to be transmitted securely.

More specifically, SSL is a security protocol. Protocols describe how algorithms should be used. In this case, the SSL protocol determines variables of the encryption for both the link and the data being transmitted.

All browsers have the capability to interact with secured web servers using the SSL protocol. However, the browser and the server need what is called an SSL Certificate to be able to establish a secure connection.

The basics of maintaining a website

Owning a website comes with certain responsibilities. You can't just build it and forget it. Well you can, but regular website maintenance is a must if you want your site to be successful.

Regular maintenance can:

Keep secure: monitor for malware, viruses, hackers, and errors

Hackers don't usually announce themselves on the front page of your website. You could be infected and not even know that your site is being used to send spam emails or links to nefarious parts of the web. Setting up a regular monitoring service will ensure that if you do get infected or have site errors, you can fix them fast.

Keep a regular backup schedule

Backing up your site is something that should be done regularly, especially for those who update their site often. Things happen. Do not expect your web host to be keeping a scheduled backup for you. While they may be, it could be old, and not on track with your latest site updates. If the server crashes for some reason, or your site gets hacked, or you make some major mistake, your edits could be gone.

Keep updated: Software

Most websites are built on a content management system, which means it's software that can potentially be exploited. We use WordPress for many reasons, but one is that it's constantly being updated, improved, and made more secure. When WordPress releases a new version, it's a must to update your site. Failing to do this leaves you vulnerable. Plugin updates should be treated in the same way – all software updates are a form of protection.

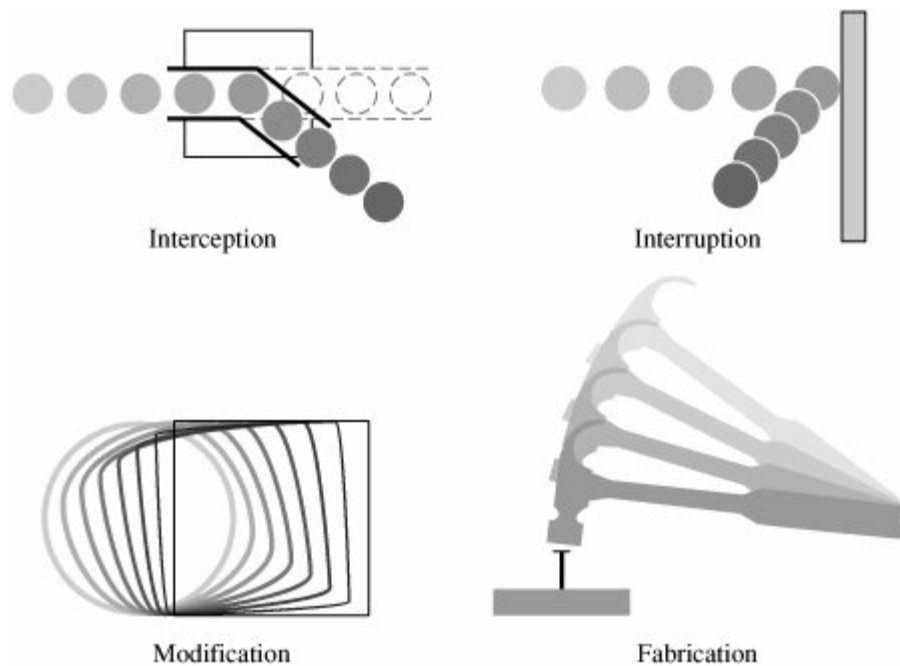
Keep updated: Content

This isn't strictly a maintenance issue, but we feel it's so important to keep your site fresh and updated on the content front that we're including it. A regular blogging or publishing schedule that pushes out relevant content will keep your returning visitors happy and engaged. The search engines will like you a whole lot more too. If there was one piece of SEO advice we had to give it's this: Publish relevant content, and publish it often.

There's a few other tasks that we should mention too:

- **Check for broken links** – nobody likes broken links. Nobody.
- **Check site speed.** A fast site is a good site.
- **Track your site statistics.** We set up Google Analytics on every site we build, but if you don't check you don't know.

Types of threats



- An **interception** means that some unauthorized party has gained access to an asset. The outside party can be a person, a program, or a computing system. Examples of this type of failure are illegal copying of program or data files to obtain data in a network.
- In an **interruption**, an asset of the system becomes lost, unavailable, or unusable. An example is malicious destruction of a hardware device, erasure of a program or data file, or malfunction of an operating system file manager so that it cannot find a particular disk file.
- If an unauthorized party not only accesses but tampers with an asset, the threat is a **modification**. For example, someone might change the values in a database, alter a program so that it performs an additional computation,
- Finally, an unauthorized party might create a **fabrication** of counterfeit objects on a computing system. The intruder may insert false transactions to a network communication system or add records to an existing database. Sometimes these additions can be detected as forgeries, but if skillfully done, they are virtually indistinguishable from the real thing.