Applying the Software Engineering Principles in Building a Secure Software

Huda Abdulaali AL_Amwee

Computer Science department

Al-Mustenseria University

E-mail: huda_ros@yahoo.com

Abstract_ These are exciting times, technology advances, principally in the field of computers have now allowed the creation of for more complex system than before, with new and complex security problems. Because modern systems cut across many areas of human endeavor, security engineering needs to consider the mathematical, physical properties of secure systems to develop it in much more efficient approach. This research concentrate on particular aspect is to build secure software under sound of software engineering, that aim to be efficient secure software in time and resources, with a great balance among productivity, security and cost, formalize the secure system requirements in precision manner, since the secure system requirements are dynamic and always would be changed, and the most important aim is to decrease the cost of secure software maintenance since it built under the principle of software engineering. In the beginning of this research there is a survey on software engineering life cycle, secure systems on the web, security-engineering principles and then display the proposed system with an example to build secure e-mail under the software engineering.

Keywords__ Security systems, security engineering, BPR, secure e-mail.

I. BACKGROUND

This section would show briefly the definitions and basics of two important fields to build secure software they are *software engineering* and *principles of security*

.1. Software Engineering [4]:

Software engineering is the establishment and use of sound engineering principles in order to obtain economically s/w that is reliable and work efficiently on real machine. Software development process contains three generic phases:

A- Definition Phase:

- Interviewing, questioning and living with the users (those needs the software to be developed) to collect all the information represent the basic elements of the software analysis (called problem and needs).

System engineering (system analysis): we would partitions the system into it is basic element and determine the job of each element.

* Feasibility study: is that software feasible as social, economical, and technical.

- Requirement analysis and definition: By no. of analysis methods the analyst converts all the needs and problems submitted by the user to scientific and formal requirement represented by diagrams and formal languages.

B- Development Phase:

- Outline design: set up the big picture, so what analyst and the users can agree on a general way to proceed.
- Detail Design: take more time, as well as more study; for inputs, processing of these inputs and output.
- Implementation: write the code of the software by programming language.
- Testing: check up the developed software by the programmers of that software.
- Quality assurance: make sure the software satisfy on the criteria of the quality.

C- Maintenance Phase:

Contain the following activities: corrective, perceptive, adaptive, and preventive. The implementation of any of these activities depends on the user opinion after operating the final result product.

1. Principles of Security [2]:

The manager or some time called security engineer responsible for security needs some systematic way for define the security requirements and characterizing, the approach is to consider three aspects of the information security:

A- Security Attacks:

Any action that compromises the security of information owned by an organization. *These attacks may be:* ***Interruption:*** An asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability. ***Interception:*** An unauthorized part gains access to an asset. This is an attack on confidentiality. ***Modification:*** An unauthorized party not gains access to but tampers with an asset. This is attack on integrity. ***Fabrication:*** An unauthorized party inserts counter fit objects into the system, this is an attack on authenticity.

B- Security Mechanisms:

A mechanism that is designed to detect, prevent or recover from security attack. Mechanisms according the security requirements are:

Cryptography: The only real defense is to ensure that any private data is encrypted before it is transmitted, so that even if the bits sent could be interpreted, only these with the ability to decrypt the message will actually be able to interpret its content. ***Passwords (Something the user know):*** Include traditional passwords have been used with us since the dawn of the computer. After providing your user ID "user Identification" to computer system, you enter your password.

C- Security Service:

A service that enhances the security of the data processing system and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service. These services are: ***Confidential (Secrecy):*** is concerned with ensuring that data can only be read by those authorized to do so. ***Availability,*** protects system resources from attacks that might render them unusable. *Integrity,* ensure the data and resources cannot be modified by unauthorized persons. ***Authentication:*** is concerned with providing mechanism to allow an entity to prove its identity.

II. SECURITY ENGINEERING [3]

Security engineering is the field of engineering dealing with the security and integrity of the real-world system. Software engineering as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. A major component of discipline has become known as <u>Requirement Engineering.</u> The process that determining exactly what is to be developed by the software engineer and determination of what performance characteristics it must possess. So, software requirements engineering as science and discipline concerned with establishing and documenting software requirement and it consist several separate stages: Elicitation, analysis, specification, verification, and management. Increasingly important requirements that are becoming more common place within the customer communities are the requirements for software and system security defenses and counter measures. The increasing demand and importance of security requirements in systems engineering has created a relatively new engineering for information systems represents a completely new discipline. During the requirement engineering process, the systems and software engineers determine the system user's definition of " Security". This can vary from " a guard at every physical doors" to comprehensive data confidentially, integrity, and availability requirement. In turn, the system engineer uses a knowledgeable security engineer to develop a security architecture that can address the needs of customers and meets the comprehensive system-level requirements. Customers and end users generally are in capable of articulating their security needs as anything more than high level declaration. Building a system to meet a security requirement is often difficult, because the problem being addressed is not static, but dynamic. Requirements such as providing an easy to use interface, on line, help facilities, or real time scheduling are static requirements. For static requirements, the technical solution can be determined when the system is built and delivered and that solution is generally viable for the life of the system. A security requirement is dynamic for several reasons. <u>***First,***</u> the security solution is depend on several operators: *1-The threat against the system. 2-The likelihood of the threat being exercised. 3-The state of technology available for system protection. 4-The state of technology for system stack. 5-The perceived value of the enterprises information assets.* <u>***Second,***</u> a security solution (in most cases) needs to be developed to defend against must likely threats. The security solution itself also a dynamic folder. The threat against an enterprise can change depending upon specific identifiable events. If security solution proposed by engineers is viewed as static, then the engineer must endeavor to establish a protection solution that addresses the max (threat) that can occur.

III. THE PROPOSED SYSTEM

For an organization want to built secure system for it is web because the impacts of security beaches. The most important decision to build this secure system for the web is to be under principles of the software engineering. Building secure web under the principle of security engineering would be done by apply the life cycle of software engineering in developing the secure system with feeding this life cycle with Internal life cycle called ***Security Engineering & process view.*** This Internal life cycle come in place of the definition phase, so the life cycle of secure software engineering would be the following three phases: ***1-Security Engineering Process View Phase (SEPV) . 2-Development Phase. 3-Maintenance Phase.***

1. Security Engineering Process View Phase (SEPV):

Starting with information gained from working closely with customers (current processes, constraints, security policies, desired out comes etc). The security engineer will generally conduct a risk assessment, analysis, a vulnerability analysis, propose an engineered solution, implement the solution test, documents procedures, and train the organization in new procedures as figure 1.

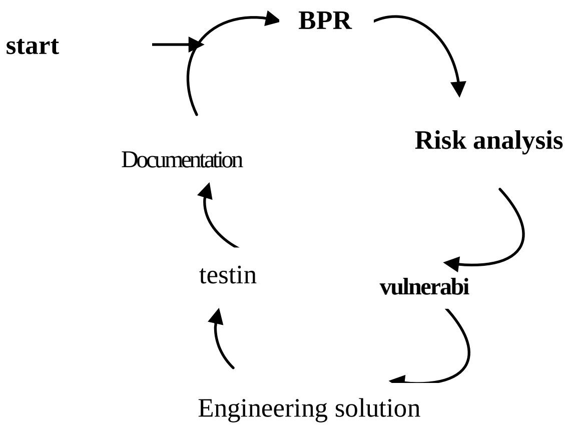


Figure (1) Represent the internal life cycle (SEPV).This phase has the following stages:

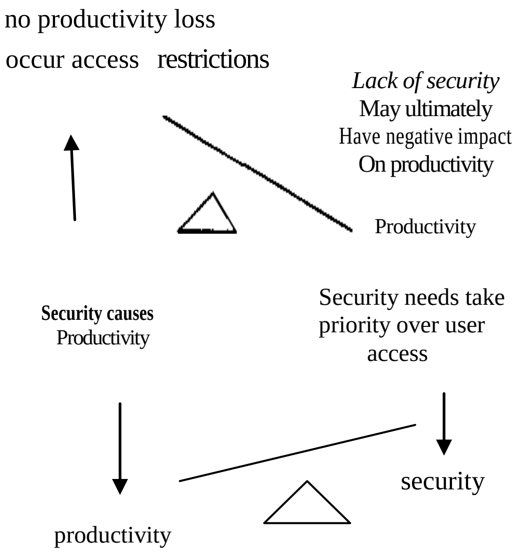
- BPR (Business Process Review).
- Risk and Vulnerability analysis.
- Engineered solution.
- Testing
- Documentation.

A__BPR (Business Process Review):

To develop a common understanding (or perhaps a common mental model) between engineer and customer, some forms of a ***business process review (BPR)*** generally occur. BRP involves the engineer, the end customer, and perhaps other stakeholders who work together to understand the current business process. Sufficiency in information security is achieved when the solutions cost, in operational terms, does not exceed it is value in terms the protection it affords.

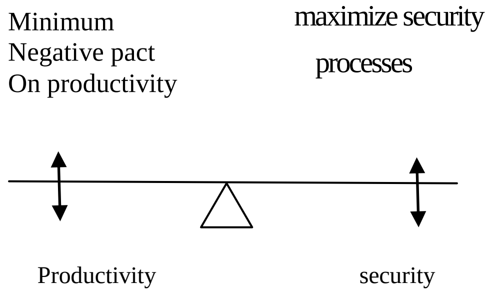
B_Risks and Vulnerability Analysis:

Proper security requirements assessment implies that appropriate security processes and technology have been applied for any given user group's access to / from any potential corporate information resources. Before proceeding blindly with a security policy development project, it is important to property define the scope or limitation of the project to decide the feasibility of the project. One of the key issues addressed during scope definition or feasibility studies is deciding on the balance between security and productivity. This will be illustrated in the following graphics 1- **Lack of security:**



C- optimal balance of security and productivity

- balanced risk and costs
- restrictiveness of security
- policy balanced by people's
- acceptance of those policies



Assets are corporate property of some value that require varying degrees of protection. The most common asset to be protected in an information systems environments is the information or data itself, these data can be classified to: ***unclassified or public, Sensitive,*** *Secret*, ***Top Secret.***

. ***Threats*** are processes or people that pose a potential danger to identified assets. A given asset can be potentially threatened by numerous threats. ***Vulnerabilities'*** are the manner or path by which threats are able to attack assets, also can be thought of as weak links in the overall security architecture and should be identified for every potential threat/asset combination. ***Risk domain*** consist of a unique group of networked system sharing both common business function and common elements of exposure. These common business functions and risk are identified during initial risk analysis or assessment.

D- Engineering Solution:

Here is the heart of the proposed system, the security engineering would decide the protective measures are used to build the security policy for the secure system. The protective measures are designed that effectively block the vulnerability in order to prevent threats from attacking assets. Among the major categories of potential protective measures are: **Authentication, Encryption, Virus protection,** antivirus scanners these scanners divided into two category scanners interpreted in the following points: ***Virus Scanners:*** virus scanner characterize viruses by their "signatures", which are stored in database. Every file on the system is scanned for the presence of these signatures and if one is found, the virus is eradicated if possible. ***Heuristic Scanners:*** have high capabilities of scanners because, It is not limited to specific number of viruses it "known viruses".

E- Testing:

After all the previous stages, the security engineer collect all the information related to built the secure system and specified the requirement according the collected information and explicit user requirement. Then he analysis the risks and vulnerabilities surrounded the web, and according the determined risk domain he would select the suitable protection measure to built the secure system. After all these steps he must built a simplified secure system and try to test it on environment similar to the real environment of the user.

According this test he would decide if this initial secure system is succeed to fulfill the user security requirement or not. If that initiated system is fail then the security engineer would back from the beginning to rebuilt the basics of that system. But if that initiated system succeed then the security engineer would beginning with the following stage.

F- Documentations:

In this stage the security engineer would document all the results of all the stages in the SEPV, this documentation would be described by the structured English scheme supported with Data Flow Diagram (DFD), General DFD and Detailed DFD. After receive these documentations to the users whose want to built that secure system. The users may be satisfied in all the analysis, the proposed engineering solutions (protective measures) and the results of system testing, so here the first phase (SEPV) would be finished and the security engineer would travel to the second phase, ***development phase,.*** Else if the user hasn't satisfied in some or all the result then the engineer would back from the beginning of the first phase, ***SEPV,*** to eliminate the previous errors caused unsatisfied results

.2- Development Phase:

There is no change in this phase in all it is stages to develop any software if was secure software or any other product. Since the secure software also need outline design, detail design, implementation, testing and quality assurance. But there is one most important consideration is to choose secure programming languages to built the secure software such as object oriented languages, like Java, Java script and VB script.

1. Maintenance Phase:

Also, there is no change in this phase, this phase absolutely decided by the system users whose want it. And according their opinions the software configuration would decide the maintenance operation even this software was secure system for a web.

IV. .EXAMPLE

In virtually all distributed environments, electronic mail is the most heavily used network-based application. It is the only distributed application that is widely used across all architectures and vendor platforms. Users expect to be able to, and do, send mail to others who are connected directly or indirectly to the Internet, regardless of host operating system or communications suite. With the explosively growing reliance on electronic mail for every conceivable purpose, there grows a demand for authentication and confidentiality services.

From here we would built secure software to protect top secure e-mail in government web from security penetrations under the principle of software engineering: The life cycle of secure software engineering would be the following three phases:

1-Security Engineering Process View Phase (SEP\0: This phase has the following stages:

A. *BPR (Business Process Review):*

To collect the information related to built secure email, in this research suggest to deal with:

- *Facilitators* They are necessary to keep the design sessions on track and to elicit and document the security requirements.
- *Stakeholders* consist of developers, users and owners. It is important to have a variety of stakeholders (i.e. owners/management and all groups of users should be represented), although for practical purposes the number of participants in the meetings is best kept to 5-6. The reason for involving both owners and users is to ensure that: 1. all contexts in which the system is used are represented, and 2. owners and users become aware of each others' goals and needs.
- *Security Experts* must be involved if neither Facilitators nor

stakeholders have any technical security knowledge. Expert knowledge is best used, however, in the Risk analysis and security design stage.

B. *Risk and Vulnerability analysis:*

The second stage focuses on clarifying the asset model of the system and the security requirements. Dependently on information gathered in the first stage we would build the risk domain, the **asset** is ***e-mail data*** is classified as ***top secret*** data. **Threats** are ***interception, interruption, modification*** or ***fabrication*** may done by ***hackers* or *crackers.* Vulnerabilities** are the path of transmission the e-mail from sender to receiver, or Ip spoofing. *1.3- Engineered solution:*

This is the third stage of an iterative SEPV process of identifying the most cost-effective countermeasures. According the results of the previous two stage the best solution to protect top secret e-mail is Pretty Good Privacy PGP provides a confidentiality and authentication service that can be used for electronic mail . ***The actual operation of PGP, as opposed to the management of keys, consist of five services: authentication, confidentiality, compression, e-mail compatibility, and segmentation (1) . we examine each of these in turn.***

- **Authentication**

Figure 2 illustrates the digital signature service provided by PGP. Illustrated in Figure 2a. the sequence ia as follows:

1-The sender creates a message.
2-SHA-l is used to generate a 160-bit hash code of the message.
3-The hash code is encrypted with RSA using the sender's private key, and the result is prep ended to the message.
4-The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
5- The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.

The combination of SHA-1 and RSA provides an effective digital signature scheme. Because of the strength of RSA, the recipient is assured that only the possessor of the matching private key can generate the signature. Because of the strength of SHA-1, the recipient is assured that no one else could generate a new message that matches the hash code and, hence, the signature of the original message.

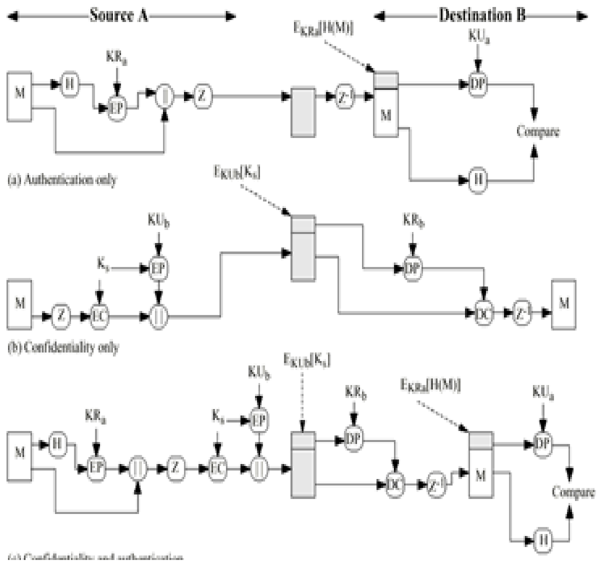
- **Confidentiality**

Another basic service provided by PGP is confidentiality, which is provided by encrypting messages to be transmitted or to be stored locally as files. In both cases, the conventional encryption algorithm CAST-128 may be used. Alternatively, IDEA or TDEA may be used, the 64-bit cipher feedback (CFB) mode is used. As always, one must address the problem of key distribution. In PGP, each conventional key is used only once. That is, a new key is generated as a random 128-bit number for each message. Thus, although this is referred to in the documentation as a session key, it is in reality a one-time key. Because it is to be used only once, the session key is bound to the message and transmitted with it. To protect the key, it is encrypted with the receiver's public key. Figure 2b illustrates the sequence, which can be described as follows:

1-The sender generates a message and a random 128-bit number to be used as a session key for this message only.
2-The message is encrypted, using CAST-128 (or IDEA or TDEA) with the session key.
3-The session key is encrypted with RSA, using the recipient's public key, and is prep ended to the message
4-The receiver uses RSA with its private key to decrypt and recover the session key.
5-The session key is used to decrypt the message.

Confidentiality and Authentication

As figure (2)both services may be used for the same message. In summary, when both services are used, the sender first signs the message with its own private key, then encrypts the message with a session key, and then encrypts the session key with the recipient's public key.



(a) Authentication only

(b) Confidentiality only

(c) Confidentiality and authentication

Figure (2) pretty good privacy

- Compression

As a default, PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space both for e-mail transmission and for file storage. The placement of the compression algorithm, indicated by *Z* for compression and Z^1 for decompression in Figure 2 is critical:

1-The signature is generated before compression for two reasons:

- It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification.
- Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty. The algorithm is not deterministic; various implementations of the algorithm achieve different tradeoffs in running speed versus compression ratio and, as a result, produce different compresses forms.

2-message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult. The best algorithm is ZIP.

- **E-mail Compatibility**

When POP is used, at least part of the block to be transmitted is encrypted. If only the signature service is used, then the message digest is encrypted (with the sender's private key) if the confidentiality service is used, the message plus signature (if present), are encrypted (with a one-time symmetric key). Thus, part or all of the resulting block consists of a stream of arbitrary 8-bit octets. Each group of three octets of binary data is mapped into four ASCII characters. This format also appends a cyclic redundancy check (CRC) to detect transmission errors. See Appendix 5B for a description.

Figure 3 shows the relationship among the four services so far discussed. On transmission, if it is required, a signature is generated using a hash code of the compressed plaintext. Then the plaintext, plus signature if present, is compressed. Next, if confidentiality is required, the block (compressed plaintext or compressed signature plus plaintext) is encrypted and prepended with the public-key-encrypted conventional encryption key. Finally, the entire block is converted to radix-64 format. On reception, the incoming block is first converted back from radix-64 format to binary. Then, if the message is encrypted, the recipient recovers the session key and decrypts the message. The resulting block is then decompressed. If the message is signed, the recipient recovers the transmitted hash code and compares it to its own calculation of the hash code.

- **Segmentation and Reassembly**

E-mail facilities often are restricted to a maximum message Length. For example, some of the facilities accessible through the Internet impose a maximum length of 50,000 octets. Any message longer than that must be broken up into smaller segments, each of which is mailed separately.

To accommodate this restriction, POP automatically subdivides a message that is too large into segments that are small enough to send via e-mail. The segmentation is done after all of the other processing, including the radix-64 conversion. Thus, the session key component and signature component appear only once, at the beginning of the first segment. At the receiving end, POP must strip off all e-mail headers and reassemble the entire original block before performing the steps illustrated in figure (3)

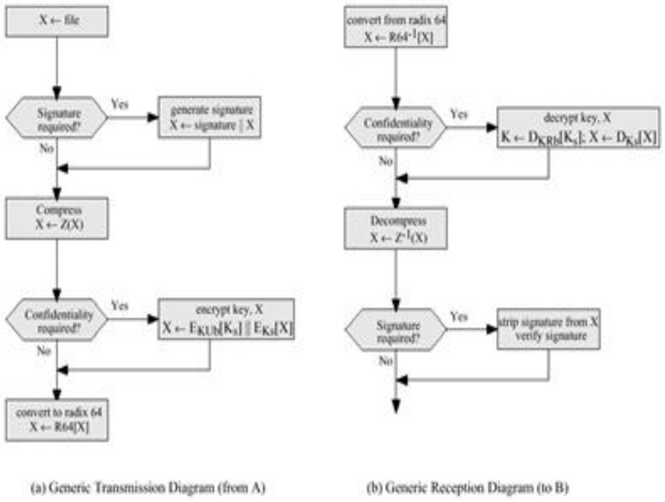

X ← file
Signature required?
Yes
generate signature X ← signature || X
No
Compress X ← Z(X)
Confidentiality required?
Yes
encrypt key, X X ← E_KUb[K_s] || E_Ks[X]
No
convert to radix 64 X ← R64[X]
(a) Generic Transmission Diagram (from A)
convert from radix 64 X ← R64^-1[X]
Confidentiality required?
Yes
decrypt key, X K ← D_KRb[K_s]; X ← D_Ks[X]
No
Decompress X ← Z^-1(X)
Signature required?
Yes
strip signature from X verify signature
No
(b) Generic Reception Diagram (to B)

Figure (3) Transmission and Reception of PGP messages

C. Testing:

After all the previous stages, the security engineer built a simplified secure e-mail and try to test it on environment similar to the real environment, Internet, of the user. Then decide if this initial secure e-mail is succeed to fulfill the user security requirement or not. If it fail then the security engineer would back from the beginning to rebuilt the basics of that system. But if it succeed then the security engineer would beginning with the following stage.

D. Documentations :

In this stage the security engineer would document all the results of all the stages in the SEPV. After receive these documentations to the users whose want to built that secure system. The users may be satisfied in all the analysis, the proposed engineering solutions (protective measures) and the results of system testing, so here the first phase (SEPV) would be finished and the security engineer would travel to the second phase, ***development phase.,.*** Else if the user hasn't satisfied in some or all the result then the engineer would back from the beginning of the first phase, ***SEPV,*** to eliminate the previous errors caused unsatisfied results.

2- Development Phase:

In this stage all the happen is an expanding in details to built the secure e-mail, with consideration of select secure programming language.

3- Maintenance Phase:

This phase absolutely decided by the web administrators whose want it. And according their opinions the software configuration would decide the maintenance operation even this software was secure system for a web.

V. CONCLUSION

The proposed system represent a new approach in information security. Although there is no cortile and perfect secure software because always there are attackers and new methods for attack, so the secure system must not be static, but dynamic and upated, this can be done by building the secure software under principles of software engineering. Security engineering build efficient secure systems in time and resource, security engineer and the users would be in direct attach, so this present a great help to the engineer because if there are any misunderstanding would be discovered in very early stage and this will increase the performance of the of engineer and make all the user security requirements would be real in the developed secure system. The most important conclusion is the decreasing of the maintenance cost of the secure system built be security engineering where it under the principle of software engineering and existing of the SEPV stage.

REFRENCES

[1*] W . Stalling ., Netwok Security Essentials : Application and Standards, Prentice-Hall, 2000.*

[2*] Unix Propeller Head ," Maximum Security : A Hacker's Guide to Protecting Your Internet Site and Networks", Macmillan Computer Publishing, Sams Net, 1997.*

[3] *M. Angela Sasse.Stephen, Ivan Flechais, M. V. Hailes, Bringing Security Home: A process for developing secure and usable systems", 2004.*

[4] T. Berztissl Alfs ,Requirements Engineering, , 2004.