Securing Object Oriented Design: A Complexity Perspective

Suhel Ahmad Khan
Department of Information Technology
Babasaheb Bhimrao Ambedkar University, Lucknow

Dr. Raees Ahmad Khan
Department of Information Technology
Babasaheb Bhimrao Ambedkar University, Lucknow

ABSTRACT
The security breaches are responsible for not only financial loss but affect the trustworthiness and integrity of the software design and development industry. The breaches are directly affecting the security attributes like Privacy, Availability, Confidentiality, Authorization, Authentication, and Integrity. The aim of security at design level is to protect software to any damage done to the security attributes. Complexity is one of the major factor affect the security. One way of security improvement is a proper control of complexity factor related to security attributes.

General Terms
Security, Complexity, Object oriented design

Keywords
Security, Privacy, Availability, Integrity, Decomposition, Coupling Function, Complexity.

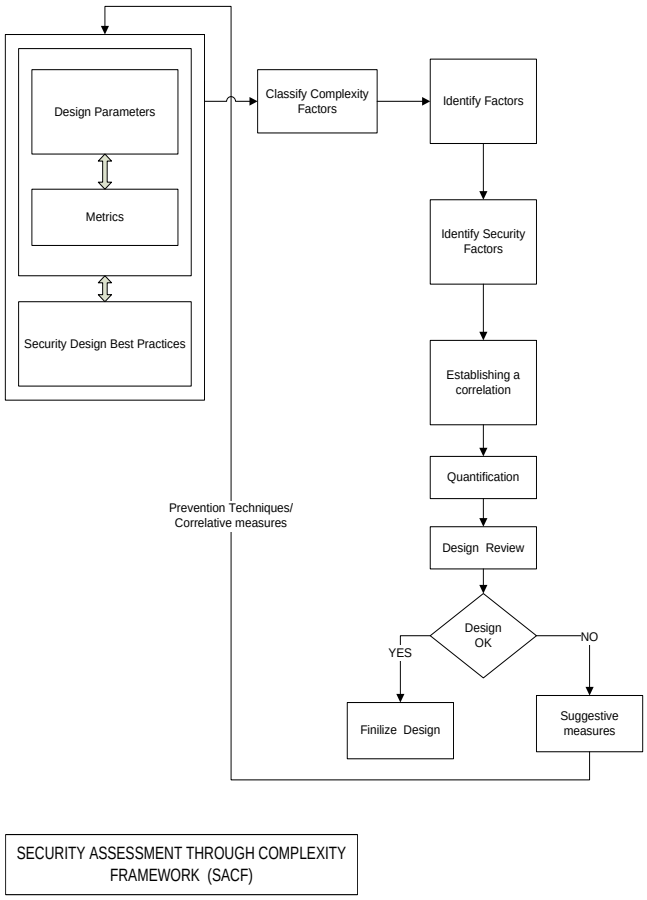
1. INTRODUCTION
The complexity of a system is the enemy of security. More complexity provides more space for attacks of intruders to violate security. But every time it is not bad, complexity is dangerous when it will extent to certain limit. Many complex programs are in real word is successfully facilitating the humanity. Even the space shuttle program or the complex structure of jet fighter planes etc. Everyone in this world is inspired with God's creation. To better understand this complexity, a human circulatory system where the least subdivision 'cells' are responsible to maintain all metabolism of system for proper function. These things are clearly identified by proper boundaries between the given levels. For a human being providing complexity in any design shows his intellectual capability to launch a well defined system which is not an easy task for others to handle it. A human being is accepted to adopt a chunk of information simultaneously is on order of seven, plus, minus two [1]. According to such information it limits the ability to deal with such complexity, but the demand to develop such complex products is increasing. The development of complex products is a combination of interrelated subsystems which forms hierarchy to the higher to lower level of components. The design of complex products is a hierarchal relationship among the components and the leading functions on it.

Security is a serious problem in software development, and may become much worse in the future. Unfortunately there is no simple solution to the software security problem. Defects that occur during software design phase impacts the whole development process. It affects the development time, production cost, quality, reliability and the most important factor for software is security. One of the major reasons for these defects at design level is complexity. McGraw states that 50 % of security issues are arises at design level [3]. Security must be integrated into the software development life cycle from the beginning and continue until the product is in use. The security of software can be enhanced by reducing the frequency of defects by analyzing the whole software development process.

2. SECURITY: COMPLEXITY PERSPECTIVE
Complexity is a major challenge for software developers to design a quality product with maximum security. The most important issue of this study is to prove Complexity as a major factor of security at design level. Any design work is a process of balancing the lay down of requirements. At designing phase the blueprint of product is being prepared. Object oriented designing concept is the best way to tackle any real word entities. A famous medium of designing which includes the decomposition of objects, explain the behavior of objects and static and dynamic models under design [2]**.** The worthful software development can be achieved only when we minimize the production cost and time with respect to quality and security. There is need to develop a framework or metrics to reduce the software complexity to enhance security which increases due to cost and difficulties software complexity. The complexity of software is associated with number of object oriented design parameters and relationship of objects to each others. The degree of software complexity is also mitigated by the extents using decomposition of design into individual components.

Complexity is a factor which affects mostly the security of the system. As complexity of software design increases, the level of difficulty also increases for different counter parts of design like activities, decisions, relationships and actions maps to be performed. The relevant information becomes more difficult to understand and to manage. [4] This creates attack surface for intruders for security breaches. The term complexity is one of the most important factors for software design & development which directly or indirectly involve with software security, quality,

development time, cost, reliability, maintenance and all possible achievements of software users. The defined framework in fig 1 Security Assessment through Complexity Framework (SACF) is an approach to finalize design by using security deigns best practices correlating them with security attributers and complexity factors.



SECURITY ASSESSMENT THROUGH COMPLEXITY FRAMEWORK (SACF)

Fig 1: Security assessment through complexity framework

3. COMPLEXITY IN OBJECT ORIENTED DESIGN

The Object Oriented concept is most efficient and effective approach for designing purpose. This particular language share most of projects and it is well recommended by academicians and industry persons due to its designing simplicity. It improves the clarity and comprehensiveness of the components. The detailed design of software as its first eye sight is usually consists of a set of all behaviors, methods and the environment in which action depth performed through relation**.** The metrics defined by Chidamber & Kermerer also supports our issues that design hierarchy having impacts on complexity by proper adjustment of objects, related methods and relation among them. The following issues to complexity which have positive or negative impact on security at design level are:

- The possibility of reusability may be decreased by the objects which have more application specific through large no of methods.
- Deeper trees constitute greater design complexity, since more methods and classes are invoked.
- If a class has large number of children, it may require more testing of methods in that class.
- The higher the inter-object coupling, the more rigorous the testing needs to be.
- The larger number of methods that can be invoked from an object, the greater the complexity of the object.
- Low cohesion increase complexity. [5]

On the basis of the above the constraint & metrics which can be identified at design level through deign hierarchy are discussed in Design Parameters & Metrics Table:

Table 1: Design Parameters & Metrics Table:

DESIGN PARAMETERS	METRICS	DISCRIPTION
No. of classes	NC	All related class must be identified at this level
No. of attributes	NA	All class attributes which fulfill the design objective
No. of methods	NM	All related methods which fulfill the design objective
No. of parameters	NP	No. of parameters used by a method of a class
No. of dependency	ND	Dependency of classes
No. of associations	NASO	Associations of class with complex relations
No. of aggregations	NAGR	A specific association
No. of compositions	NCOP	Specific aggregation
No. of generalizations	NGR	Subclass inherited all non primitive char. of parent class
No. of bindings	NBD	Total interconnections of classes and its elements

4. SECURITY AND DESIGN COMPLEXITY

Brooks also suggests that 'The complexity of software is an essential property, not an accidental one' [2]**.**No pertinent solution is available to remove complexity, but reduction or minimization is the key term to manage complexity. Main concern of study is to identify those factors and other inherent properties which have

impact on complexity and some of them are: the complexity of problem domain, the difficulty of managing the development process, the flexibility possible through software, and the problems of characterizing the behavior of discrete systems at design time. There is lot of specific design practices available which are best known guideline for security design principles. As Saltzer and Schroeder describe in 1974 and again in more descriptive manner Bishop [6] has explained these design principle for secure design issues. This novel work presents a key principle for secure designing. The security design principles are built upon an idea of simplicity, separation and restriction. [6]

At design time, a specific way to handle complexity is to prevent the contradictory inputs. For large real complex operations like ATC, aircraft egg, where ambiguous inputs cause trauma, safer preservation of inputs can be achieved by providing maximum strength of protection. At that level security can be enhanced by controlling complexity through maximum strength of protection, because in reality most of development resources are used for the preservation.

Software provides maximum flexibility to users and designers to express anything at its higher level of abstraction which depends up to the discretion of observer. At viewer's perspective what he wants to see, those necessary features can be viewed by such kind of abstraction. The economy of mechanism supports abstraction that design should be simple and the component and its interfaces should be clearly defined for unnecessary annoyance.

For a precise design, there must be a possibility of improvement, but these things only possible when the designer also understand the real behavior of component and its supporting system. The separation of concern also suggests that one component of system having some minimal impact to another. E.g. Like the high technology missile system the Patrick missile destroys the ballistic missile by following the trajectory of that missile by knowing the proper motion behavior. Considering the nature of complexity, it concludes that there are following five attributes common to all complex system. [2]

- Hierarchical structure
- Relative primitives
- Separation of concerns
- Common pattern
- Stable intermediate forms

It is well known concept that at design level a complex system is the combination of its components and the hierarchical relationship between those components [7].All systems are interrelated to each other up to the lowest level of the component. A hierarchal structure is also decomposed into parts, which belongs to a factor of complexity that decides the arrangement of the components as their threshold limit. Through decomposition the unity and integrity of design also be maintained.

The separation of concerns among the various parts of a system having interconnected or intra connected provide a clear significance that the linkage is generally stronger in the case of intra connected rather than interconnected components, because their linkage are on the basis of internal structure of the components and interaction among components. The principle of least common mechanism also describes the avoidance of duplicity of unnecessary mechanism which is helpful to reduce complexity of design by minimizing the coupling of components**.** [8] The principle of least privilege states that only provide those compulsory request which is needed for process. Minimum privilege between services and request should be provided to avoid low performance ratio and more error prone situation. Total supporting services of the components of design is another big issue for security concern because complexity of design enhanced by all related supporting services of components and their relationship**.** [8]

The economy of expression is a best way to implement a complex system. The reusability process of component can be involved to understand the common patterns of complex system. An example is helpful to clarify that the biochemistry of cell is that it's a lowest unit of system and having is in same pattern either in plant or animal. [2] These factors are related to the concept of design principles. The principle of least privilege, the principle of separation of privilege and principle of economy of mechanism are those principles which affects complexity as various manners. Through analysis of these design principles the factor of complexity are defined. The least privilege said that each component should be allocated a sufficient privilege to accomplish its specified functions.

A comparison through the factors which derives complexity[2], and the issues related to software design principles produces a result as a complexity factor which having a great impact on the multiple attributes of security such as Availability, Integrity, Confidentiality, Authentication, authorization, and Privacy.[9,10] . Fig 2 shows a correlation of security attributes with identified complexity factors. Security is a big challenge for software design and this become much worse in the future. All industry persons or academician are helpless because no simple solution is available for software security problems. Defects, vulnerabilities and design flaws are unintentional and complexity is one of the factors which influenced such security breaches. Their prevention is necessary for reducing those factors which have impact on security. The common issues of complexity having weight on software security directly or indirectly need to be identified. The identified factors of complexity are as follows:

- Coupling function
- Total supporting services
- Minimum privilege between service and requests
- Maximum strength of protection
- Maximum depth of hierarchy
- Behavior of component
- Higher level of abstraction
- Decomposition

5. CORRELATING COMPLEXITY FACTORS WITH SECURITY ATTRIBUTES

Fig 2: Correlation diagram

Complexity is a noticeable matter for secure design. The deep analysis of every factor of security and complexity shows a correlation between them as follows:

5.1. Privacy:

Privacy is related to the disclosure of information of individuals. It limits the access of user's personal information accordingly to the privilege provided between services and requests. In modern age where mostly works done on computer and internet like banking, reservation etc. and our all activities are traceable through IP addresses and browsers history. Everyone is paying a price as privacy to work on computer to solve his daily routine task as fewer efforts. At design level there is chance to loose privacy policy through decomposition process. [11]

5.2. Availability:

Availability readiness for correct services, which can also be defined as the degree to which a system or component is operational when required for use. Availability is also related to maximum strength of protection because the information can be traced by intruders to violate security. To provide services in a very high availability medium virtual servers takes responsibility to serve and having all information of client keys. This offers a chance of leakage of information through random no of virtual servers. Therefore security of services highly dependent on the proper behavior of all servers and these servers need enough protection in form of hardware or software. [12]

5.3. Confidentiality:

Confidentiality refers unauthorized disclosure of information. It also limits the access of information in right direction and prevention of disclosure of information to unauthorized users. The confidentiality works as a security policy that insure no one can access the date or information outside of this system which insures that a protection technique is also implemented here. Confidentiality is a broader concept of privacy which limits access to individual's personal information which requires a trusted binding mechanism of design and its total supporting services and related components. It insures that there is no chance of leakage of information**.** [13]

5.4. Integrity:

Integrity is the concept of credibility of information resources. It allows the possible authorized changes to insure that during alteration of information that appropriateness of design or information must be insured at origin level or source level. In integrity process the whole information is also validated which insure that changes having not any affect of the integrity. Researchers also identified the decomposition, abstraction, behavior of components and maximum depth of hierarchy and coupling having affect on integrity of design. Hierarchy is the primary tool to manage complexity. [2, 5, 9, 13]

5.5 Authentication:

The authenticity is always bound by the behavior of the component. A trusted binding mechanism is always used to bind data for the execution environment by conform identification process for security concern. That will also checks that the data authenticity will not change before completion of the process. Authentication process obtains identification information from user and validates this information against authorized process. If this information is valid, it will be considered as authentic entity and having access to given resources. This requires a maximum strength of protection for authentic process, because the weakness can arise when this identification credentials are stolen accidentally revealed or forgotten. This can be avoided through more securing process like biometric security or digital certificates. [13, 14]

5.6. Authorization:

In authorization process where decision has been taken for access to a protected resources works on trust and appropriateness for access these resources. These trust and appropriateness can be maintained through providing minimum privilege between services and request by providing maximum strength of protection. Authorization process is like that you are showing your identity card on ticket checking counter at the time of boarding. It is the process that person or process or entity is identified that he have right to use resources. [13, 15]

6. CRITICAL FINDINGS

Software engineering has been long seen complexity as the major challenge to good design. Object oriented technology ware developed to enable the design of very large and complex software. Security design principles are best theoretical

information for creating secure application. These guidelines presents a right way too thinkers and researchers that how these rules works on complexity factors which having impacts on coupling , abstraction, decomposition and hierarchy etc. The complexity factors have been identified and having great effects on security attributes are as follows:

- Coupling function
- Total supporting services
- Minimum privilege between service and requests
- Maximum strength of protection
- Maximum depth of hierarchy
- Behavior of component
- Higher level of abstraction
- Decomposition

It's hard to ignore those factors which impacts on security at design level. At design level the complexity of whole design is increased. Hierarchical decomposition of design breaks whole design in modules having a proper functioning with less interdependency. So it distributes complexity of design from more to fewer for better understanding and proper functioning. The privacy of design may be hurt by decomposition of dependency associations of objects. Maximum depth of hierarchy indicates more inherited classes and methods from the root to leaves, which influenced by the properties of its ancestors and total possible impact depends upon decedents. To avoid more cumbersome situation in design for the inter object coupling of non inherited methods should be minimum. More coupling between objects and used methods can violate integrity of design.

7. CONCLUSION

Security problems arise because of the lack of inherent security measures. An effort in respect of early and accurate security estimation needs to be undertaken for worthwhile software development. An approach make a novel contribution for secure design and security assessment by proper mapping of security attributes to complexity factors through SVCF. It appears inevitable to have a potentially effective approach for an early, on time and accurate estimation of security during development life cycle.

8. ACKNOLODGEMENT

This work is sponsored by University Grant Commission (UGC), New Delhi, India, under F.No.34-107\2008 (SR).

9. REFERENCES

1. Miller, G. March 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on Our capacity for Processing Information. The Psychological Review vol. 63(2), p 86.
2. Greedy Booch ,Object Oriented Analysis and Design with Application,.. 3rd edition, Addison Wesley ISBN-0-201-89551-X
3. Davis N., Humphrey w. Redwine S.T.Jr. Zibulski, G.Magraw,Processes, May-June 2004, for producing secure software Security & Privacy, IEEE Volume 2,Issue 3, Page(s):18-25
4. Stephen T. Albin ,The Art of Software Architecture: Design Methods and Techniques ,John Wiley &Sons © 2003 ISBN: 0471228869
5. Shyam R. Chidember, Chris F. Kemerer, OOPSLA'91 Towards A Metric Suite for Object Oriented Design, ACM, pp.197-211
6. Per Hakon Meland, Jostine Jenesen, ARES.2008,Secure Software Design in Practice,.28,0-7695-3102-4/08 © IEEE
7. Dr.R.A.Khan, Suhel Ahmad khan, A Roadmap for Security, June 2010 , International Journal of Computer Science & Emerging Technologies (IJCSET) 5 Volume 1 Issue 1, June 2010,(pp:5-8)
8. Terry V. Benzal, Cynthiya E. Irvine ,Paul c. Clark, Design Principles for Security, Secure code Technical Report ISI-TR-605,
9. Wang, C. and W.A., Wulf, 1997, A framework for security measurement.. Proceedings of National Information Systems Security Conference, Oct. 7-10, Baltimore, pp: 522-533.
10. Bharat B. Madan, Trivedi, Dependable systems and networks DNS'02, Modelling and Quantification of Security Attributes of Software Systems, , 0-7695-1597-5/02© 2002 IEEE
11. http://www.askwebhosting.com/article/129/Computer_privacy.html,
12. Li Gong , June, 1993, Increasing Availability and Security of an Authentication Services, , IEEE Journal, vol.11,No 5, ,pp 657-662.
13. G,H. Walton,Thomas A. Longstaff, r.C. Linder,97, Computational Evaluation of Software Security Attributes, 8-0-7695-3450-3/09 © IEEE.
14. http://msdn.microsoft.com/en-us/library/eeyk640h.aspx
15. Clifford J. Berg, , High-Assurance Design: Architecting Secure and Reliable Enterprise Application Addison Wesley Professional,2005, ISBN: 0-321-37577-7