Measuring Security Requirements for Software Security

Shareeful Islam, Paolo Falcarin
School of Computing, IT and Engineering
University of East London, UK
{shareeful, falcarin}@uel.ac.uk

Abstract-**For the last decade's software security has gained attention by industries, experts and all other communities. Secure software is about mitigating risks from assets to achieve business goals. Security is highly depending on the context where software is deployed. But measuring software security even within a specific context is still not mature. This is because properties and metrics for measuring security are not properly defined and methods are lacking to provide a complete picture for measuring software security. Here we identify security requirements through asset based risk management process to describe soft ware security goal. Then based on the Goal-Question-Metric approach the identified security requirements are evaluated for measuring software security.**

Keywords- ***Security metrics, security requirements, security goal, software security***

I. INTRODUCTION

Software security defines that only authorized parties can access and use software in an authorized way. However, ensuring security is challenging because software becomes more complex day by day. It is continuously reported to be vulnerable to attacks and compromises despite of using latest security techniques and protocols. That is why software is one of the root causes of all common computer security problems. Security needs to be considered and measured from the early stage of development such as the Measures and Measurement for Secure Software Development by Homeland Security Department, the Microsoft Security Development Lifecycle etc. Security measurement determines strengths and weak security properties of the product. For this it needs to be identified what to measure, how to measure and when to measure.

Without a systematically defined approach for security measurement and without good metrics, it is hard to answer, how secure a software product is. How effectively can the product handle security risks? Security is part of software quality that is known to be difficult to define and measure [10]. Reasons behind this are that the nature of security does not allow one to easily measure its worth, that it is difficult to obtain quantitative and qualitative result, and that risks and the threat environment change continuously. Metrics for measuring security is relatively young [5]. Research work considers more on the overall information security management. Information security management and on the other part software security have been handled as separate areas [11]. Properties for assessing secure software and security metrics are not well defined and a widely accepted approach is still missing [4].

The paper contributes to measure goal for software security. Security goals such as confidentiality, integrity, availability, authenticity, monitor, accountability and non reputation are considered as properties to achieve software security objectives [9]. To accomplish the task a complete software security risk management needs to be considered at an early stage of the development. A set of security requirements identified from risk management is used to describe the security goals of the software. These security requirements are the basis for measuring security. To our knowledge, there have been no attempts to use the Goal-Question-Metrics (GQM) approach for defining software security measures. The GQM allows a clear derivation from software security goals to metrics by developing questions that relate to the goals and are answered by metrics. This way, we can define clear and comprehensible measures for a set of established security requirements.

II. APPROACH FOR SECURITY MEASURING

Security metrics characterize, manage, evaluate and improve the security activities. It may also be used for decision support for instance, certification or evaluation of a product, risk control activities from risk management, security testing etc [11]. Metrics can derive from direct, objective, absolute value or from indirect, subjective, relative value. The approach here for measuring security consists of two consecutive steps. Firstly security requirements are identified through risk management to describe software security goals and finally the identified security requirements are measured through the GQM approach. Both objective and subjective measures are considered for the security properties. This section gives a short introduction of the whole process.

A. Security Requirements Identification

Security requirements identification is a step by step approach with considering a complete asset based risk management at the requirements engineering phase. The paper main contribution is on security metrics so here only a short overview is given for this part. The security quality requirements engineering (SQUARE) methodology [2] is partially followed here for identifying security requirements. The whole task starts with identifying the scope of the product. Asset based risk management is then conducted for identifying risks for all critical assets. Critical assets are identified based on costs for production and reproduction, the amount of loss for any destroys etc. Possible threats and vulnerabilities to this critical asset are then determined through threat profiles, attack trees, threat sources etc.

Identification of asset, threat and vulnerability related to these assets are critical elements for risk identification. These risks are then analyzed by the likelihood of occurrence and by estimating their negative impact. Finally, a mitigation plan, protection strategies and action lists are developed to control the risk at an acceptable level. Security goals and policies are then outlined considering the product and organization. Security goals are the organization's motivation and business gain by applicability of the management control principles. Security policy sets out conditions to achieve the security goals. The SQUARE process considers security goals before risk management while here we consider it after risk management. The reason is that risk management identifies critical assets; risk associates for this asset and plans how to control this risk. Security goals and policies can then be outlined from these control strategies. Results of risk mitigations can also be used as a guide to elicit an initial set of security requirements by using any suitable elicitation techniques. Identified security requirements help to determine what need to be protected, from whom it needs to be protected and how long and with how much cost. These initial security requirements are then validated to achieve the security goals of the product. Finally these initial requirements are refined for deriving a final set of security requirements. However, these identified security requirements are not enough for ensuring software security goal. It needs to be measured. In the following, we show how these requirements can be assessed through the GQM approach.

B. Goal-Question-Metric approach

The GQM approach is a mechanism that provides a framework for defining and interpreting metrics [1]. We consider this paradigm for measuring software security goals because it is a basic method in software measurement and provides a clear derivation from goals to measures. GQM involves a set of goals, generating questions that define those goals as completely as possible by quantifying them. Specifying measures answers the questions in order to identify whether the goal can be achieved or not. A set of data are interpreted from answering the generating questionnaires to track the goal. A goal can be defined for an object (product, process, resources), for a variety of reasons and from various point of view. A set of values from various parameters can define the goal. The GQM also defines a template for defining these different attributes of a measurement program. In our case of measuring security requirements, the template looks as follows:

Purpose (why): Evaluation for improve and ensure
Object: Product (software)
Issue (with respect to): Security
Perspective (focus): Security requirement
Viewpoint (who): Stakeholder, user, vendor, developer
Environment (context): organization, people factors
When: Mostly in early phase of development

The top goal is a secure software system. This goal is then split into sub-goals with all identified security requirements. Every sub-goal has its own parameters such as view point, purpose, etc. Each sub-goal is then refined into several questions. The questions characterize the security requirements to achieve security objectives. Finally, each question is refined into metrics, some of them are subjective and some of them are objective. Next section shows how the GQM approach is used to measure a set of security requirements based on a software security goal.

III. METRICS FOR SECURITY REQUIREMENTS

Security requirements are actually a set of conditions that describe properties such as confidentiality, integrity, availability, authenticity, and non-repudiation etc of software security goal. It is a set of requirements which consider organization policies, security goals and security policies [8]. The security requirements identified here are based on the security goal and the organizational environment where software is operating [3]. That is why consideration is also on Information Security Management System Standard ISO/IEC 17799:2005 [7]. Table 1 shows relationship among properties for security goal, INCITS/ISO/IEC 27001-2005 clauses and the security requirements. For instance clauses no A.11 defined access control to ensure authorization for access and process organizational service and information. This clause can ensure security properties confidentiality, authenticity and security requirements identification, authentication and authorization requirements, privacy requirements etc. Clause A.12.1-A.12.4 defined correct processing, control of information. This clause can ensure integrity and privacy requirements and integrity properties of security. Due to space limitations all clauses with security properties and requirements are not elaborate. These identified security properties are considered as sub-goals and follow the top down GQM approach for measuring security. Some parameters are set to define the sub goals. A set of questions from section 3.1 to 3.8 are used to characterize to track the sub goal. Answers of these questions are objective or subjective to quantify the goal for security metrics.

TABLE 1. RELATIONSHIP AMONG SOFTWARE SECURITY GOAL, RELATED ISO/IEC 17799:2005 CLAUSES AND SECURITY REQ.

Software security goal	ISO/IEC 17799:2005 clauses for control	Security req.
Confidentiality Authenticity	Access control(A.11) User access man. (A.11.2) User res.(A.11.3) Application & information access control(A.11.6)	Identification, authentication and authorization req. Privacy requirements
Confidentiality Integrity Authenticity	Security req. of IS (A.12.1) Correct processing in app.(A.12.2)	Identification, authentication and authorization req. Privacy

	Cryptographic controls (A.12.3) Security of system files (A.12.4) Electronic commerce services(A.10.9)	requirements Integrity req. Service availability req.
Monitor Nonrepudiation	Protection against malicious & mobile code (A.10.4) Monitoring.(A.10.10) Reporting IS events weakness (A.13.1) Management of IS incidents & improvements (A.13.2) Compliance with legal req.(A.15.1)	Malicious activities detection req. Security auditing req. Nonrepudiation req.
Availability	Back-up (A.10.5)	Back up and recovery req. Service availability req.

Figure 1 show how the GQM approach is used to measure software security goals through security requirements. Here Goal is abbreviated by G, sub-goals are abbreviated by SG1, SG2, SG3…SGn. Security requirements are abbreviated by SR1,SR2…SRn, total 8 sub-goals (SG1--SG8) defined from 8 identified security requirements(SR1--SR8). Questions for specific security requirements are abbreviate by SR1.Q1, SR1.Q2, if metrics required more than one value for relating to the question then abbreviate by SR1.Q1.M1….SRI.Q1.Mn, other wise only SR1.Q2.M1.

G1= SR1= Identification, authentication and authorization G2=SR2= Privacy, G3=SR3= Integrity G4=SR4= Service availability, G5=SR5= Non repudiation
G6=SR6= Malicious activities detection
G7=SR7= Security auditing, G8=SR8= Back up and recovery

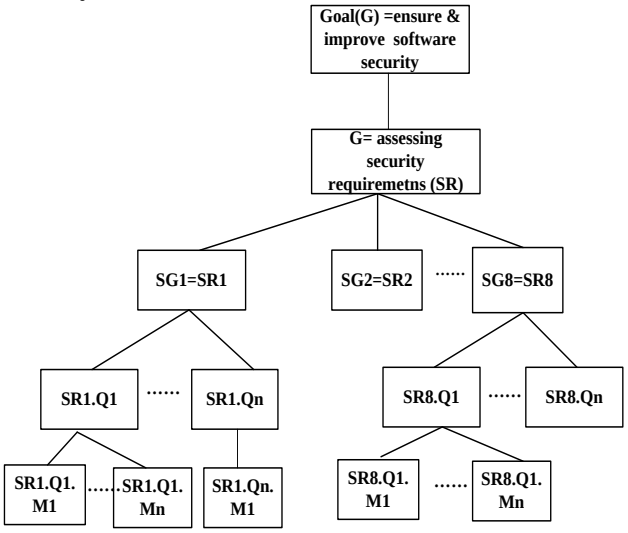

FIGURE1. GQM APPROACH FOR ASSESSING SECURITY REQUIREMENTS

A. Identification, authentication and authorization requirements

Identification and authentication defines user identity and how this identification can be verified before the interaction with the system. Authorization extends this to verify user access and usage privilege to the system resources. Identification and authentication establishes the basis for accountability and is the prerequisite for authorization. Metrics for this combined security requirements consider on the properties for all user identification, authentication and authorization are shown in Table 2.

TABLE 2. METRICS FOR IDENTIFICATION, AUTHENTICATION AND AUTHORIZATION REQUIREMENTS

G1=SR1 Purpose Issue Viewpoint	Ensure Identification, authentication and authorization User, organization
Question (Q)=SR1.Q	**Metrics(M)=SR1.M**
How are the users identified?	Subjective evaluation by how all level user uniquely identified for using the system
How are the users authenticated?	Name, password, token, ID, biometric (finger print, voice)
How is authentication session managed?	Maximum time per authentication, maximum time the session continues without user active interaction.
No of fail attempt to lock user account?	Total failed login attempts for a specific user
How grant user authorization?	Personal or role based or group base with least privilege principle or other means
How many resource and level of use grant for single user?	No of resource access and level of use as access profile for every user
How many layers of authentication / authorization check?	No of authentication/ authorization check per usage

B. Integrity Requirements

Ensure that data, component, communication managed by software are not unauthorized modified, create, or delete. Metrics to measure integrity consider mainly on validation check for input from different source are shown in Table 3. Validation plays a critical role to ensure integrity. But challenge in validation is that so many interfaces due to complexity of the software, so building validation at one interface do not prevent attacks to other interfaces.

TABLE 3. METRICS FOR INTEGRITY REQUIREMENTS

G2=SR2 Purpose Issue Viewpoint	Improve and ensure Integrity Stakeholder, User, organization
Question(Q)=SR2.Q	**Metrics(M)=SR2.M**
How many validation checks for a set of input?	No of validation check (V) per total user input (Tu) or total input from external sources (Tes) for a specific purpose(application or interface). Higher ratio of V/Tu or V/Tes gives strong validation
How many points specific module take validate input?	No of points (Pi) module take input ,no of points module check validation (Pv). Ratio of Pv/Pi =1 ,indicate all input points have at least one validation check
How application program ensure integrity?	subjective, ensure application immune against malfunction
How error checking and exception handle during normal operation?	Subjective, based on error checking and exception handle techniques, features of the programming language, total no of functions call that do not check return value for a specific module
What are the numbers of critical class or module that process data?	Total number of class or module, determine by its location where it store, its responsibility to the critical data modification or offering critical functionality, its value in software as asset, probability to target for attack
How race conditions handle when more than one user intend same time to use shared data?	Subjective evaluation by appropriate techniques to ensure synchronization.

C. Privacy Requirements

Privacy means that sensitive data, communications and application are protected from unauthorized individual or programs from both internal and external access. Metrics for privacy consider how handling, transmitting and storing confidential data, communication etc. are shown in table 4.

TABLE 4. METRICS FOR PRIVACY REQUIREMENTS

G3=SR3 Purpose issue view point relate with	Ensure Privacy User, organization, supplier legal, identification, non repudiation
Question(Q)=SR3.Q	**Metrics(M)=SR3.M**
How secure is data transmission?	Cryptographic algorithm ,algorithm to generate key, key length
How strong is the user password?	Password length, minimum combination with letters, numbers and other characters, restrict common word for password selection, lifetime of password
How password store?	Encrypted, hash code ,plain text
How data classify?	Based on its value, location where it store, value by any security breach etc. classified as sensitive, confidential, public.
How critical data handle, store and access?	Through application or by other means, location with encrypted or plain text for store, appropriate access control mechanism.
How critical module store and access?	Location where it store and access control technique
Does any data require to compliance with any legal requirements?	Yes or no and Subjective evaluation how it make the compliance
How random number consider for cryptographic keys or for other purposes?	Source of random numbers, Seed and entropy size by no of bits

D. Service Availability Requirements

Service availability means that services managed by software must be available as per the requirements. Metrics for these requirements consider the parameters used to make the service available are shown in Table 5.

TABLE 5. METRICS FOR SERVICE AVAILABILITY REQUIREMENTS

G4=SR4 Purpose issue viewpoint relate with	Ensure service availability Stakeholder, user backup and recovery
Question(Q)=SR4.Q	**Metrics(M)=SR4.M**
How much time a specific service is available for normal operation?	Percentage of time (Ta) services by software available for a given period of time (Tt). Ta/Tt =1 implies the service is available for the whole given period.
How fast can a specific service recover?	Difference between recovery time (Tr) and fail time (Tf). Small value of (Tr-Tf) give more impression for availability
Identify the possible point for a service failure?	Calculating total point by parameters such as no of dependency with other internal or external functionality, dependency upon hardware etc.
How critical services interact with each other and maintain trust relationship?	Subjective evaluation by interaction methods and trust relationship among services

E. Non Repudiation Requirements

Non repudiation implies that software shall prevent parties to deny after participate in any interaction. Interaction can be transaction, communication among parties etc. Metrics for this requirement is shown in Table 6.

TABLE 6. METRICS FOR NON REPUDIATION REQUIREMENTS

G5=SR5 Purpose issue view point	Ensure non repudiation Stakeholder, user
Question(Q)=SR5.Q	**Metrics(M)=SR5.M**
What attributes consider for proof of transaction and how it store?	Date, time, authenticate identity of interacting parties, information or data store , relevant ack, store automatically by software as log or normal file
How log file can access?	Appropriate access control technique
How to check all interaction entry in log?	Ratio of no of entry in log (Ne) to no of interaction (Ni). Ne/Ni=1 means all interaction entry in log

F. Malicious Activities Detection Requirements

Software shall detect and record all attempt by user and program to unauthorized access modification etc. and shall also protect itself by attacking from undesirable programs such as virus, worms etc. Metrics may consider the properties of audit record and techniques to detect and protect from malicious code are shown in Table 7.

TABLE 7. METRICS FOR MALICIOUS ACTIVITIES DETECTION REQ.

G6=SR6 Purpose issue view point related with	Detect Malicious activities Organization Identification, authentication & authorization
Question(Q)=SR6.Q	**Metrics(M)=SR6.M**
What are the contains of audit trail record?	Every fail attempt for authentication, authorization modification or for other illegal actions time, date
How long it takes to report after any security violation?	Time difference between report of security violation (Tsvr) after the violation occurred(Tsvo), Small value of Tsvr-Tsvo is preferable
How fail login attempt count?	No of total login attempt (Na)-No of total successful login (Ns) for a specific period of time.
How virus, worms or other malicious code detect, report and stop?	Subjective evaluation by techniques to handle malicious code.

G. Security Auditing Requirements

Audit implies that extent upon which software shall enable to monitor its status and use security mechanism. Metrics consider formal examination and review of actions taken by considering all security requirements to achieve security goal and policy are shown in Table 8.

TABLE 8. METRICS FOR SECURITY AUDITING REQUIREMENTS

G7=SR7 Purpose issue view point related with	improve Security audit stakeholder, organization security requirements, policy, goal
Question(Q)=SR7.Q	**Metrics(M)=SR7.M**
Do all security mechanisms work, update properly and support required security policies and goal?	Yes or no, subjective evaluation if require and time duration to generate report

H. Backup and Recovery Requirements

Software shall ensure proper backup of all critical data and recovery when it requires. Metrics for this requirement can consider backup and recovery properties shown in Table 9.

TABLE 9. METRICS FOR BACKUP AND RECOVERY REQUIREMENTS

G8=SR8 Purpose issue view point	ensure backup and recovery organization availability
Question(Q)=SR8.Q	**Metrics(M)=SR8.M**
How often critical data, log, audit trail backup?	Measure backup frequency per day, per week or per specific time and techniques to do backup
How all back store, authenticate, locate, retain?	Plaintext or encrypted, access control for backup, location where it store, amount of time backup retain
How much time it takes to recovery back up and any legal requirements to compliance?	Amount of time to recover the backup

Here a set of security requirements based on software security goals are measured. However, depending on the context, requirements can be changed. Some metric values are subjective by nature which is difficult and elaborate to measure. Output of the metrics can use as a checklist of artifacts for software security measurement.

I. Evaluation of the Sub-Goals

There are various possibilities to combine sub-goals to goals. We propose one example here. All sub-goals are evaluated through a score to identify whether main goals can be achieved. We consider sub-goals with equal weight of 100 percentages. Every question is assigned to a maximum value 1.Metrics are given (based on full, average, weak

compliance) a value to the question [6]. Accumulate scores from all questions are then averaged to give the total score from all questions. Finally, multiply it by 100 for giving actual score of a sub-goal.

Suppose we are going to evaluate identification, authentication and authorization requirements for an inventory management system to achieve authenticity security goal. Employees working to update stock amount for different items need to identified, authenticate and authorized by the software to update the stock. Value assigned SG1=SR1=100, SR1.Q1= SR1.Q2= SR1.Q3= SR1.Q4= SR1.Q5= SR1.Q6= SR1.Q7=1

Scores from metrics for every question are

SR1.Q1=1 (full compliance by employee id and name)

SR1.Q2=0.5, (average compliance by password)

SR1.Q3=0 (weak compliance, no such time)

SR1.Q4= 1(Full compliance, after 5 fail login attempt)

SR1.Q5= 1(Full compliance, role based)

SR1.Q6=0.5 (average compliance, only level of use defined)

SR1.Q7= 0.5(average compliance, only one layer)

Total score =100 * (1+0.5+0+1+1+0.5+0.5)/7= 64%

So identification, authentication and authorization requirements can achieve 64% from the system.

IV. RELATED WORKS

Security metrics research is still in an early phase. Some works have already been done to introduce security properties, metrics or mentioned how and where in the development life cycle can be measured. Scandariato [4] elicits security properties to quantitatively asses software security in the architecture and design phase of development. Here, metrics are considered to reduce complexity of software design, ensure a multiple, layered based approach and identification of critical modules for software security. The author further associate security metrics to security patterns [5] and patterns to security objectives. Our approach does not explicitly make use of security patterns.

Savola proposes in [11] a common security metric to reduce the gap for business information security management, products, systems and services. Security, trust and dependability metrics for products consider are organized in five different levels. Individual metrics are considered for products in different phases of development. No approach is given how to elicit the metrics or how to interpret them. Khan [6] assesses security properties for third party software components. The approach considers security classes, percentage weights to security objectives and ratings of security functions associated to the security classes. Our approach has similarities with Khan's approach. However, we consider direct derivation from security goals to security requirements rather than using security class.

V. CONCLUSION & FUTURE WORK

Measuring quality is generally a difficult, nevertheless important topic. Even for the hard to grasp quality attribute security, it is important to be able to measure the level of security and to identify the weak points in the security implementation. Only with measures it is possible to make project management decisions on a well founded basis.

We use a set of security requirements derived from software security goal and the accepted ISO/IEC 17799:2005 standard for information security as a baseline to develop such security metrics. It is obvious that there cannot be a single measure for security as it is a multi-faceted concept. We define a set of measures that cover these different facets.

The GQM approach is used for a structured and comprehensible derivation of the metrics. It allows to clearly relating the defined measures back to the original security goals. This provides a well-founded basis for our security metrics. But subjective evaluation of metrics is hard. We need to refinement this value further.

Nevertheless, measuring security needs to be set into the context of general quality measurement. Hence, for future work, we not only plan to conduct case studies for experimenting with the proposed security metrics. We also intent to incorporate these security metrics into broader quality models that can be operationalized.

REFERENCES

[1] V. R. Basili, G. Caldiers, H.D. Rombach., The goal Question Metric Approach.

[2] N.R. mead, E. D. Hough and T.R. Stehney, Security Quality Requirements Engineering (SQUARE) Methodology (CMU/SEI-2005-TR-009).

[3] D. Firesmith. Engineering Security Requirements. Journal of Object Technology, Vol.2, No.1, 53-68, January-February 2003.

[4]R.Scandariato,B.D. Win, and W.Jossen, Towards a Measuring Framework for Security Properties of Software QoP'06, October 30, 2006.

[5] T. Heyman, R. Scandariato, C. Huygens and W. Jossen, Using security pattern to combine security metrics.

[6] K.M. Khan and J. Han, Assessing Security Properties of Software Components: A software Engineering's Perspective, Proceedings of the 2006 Australian Software Engineering Conference (ASWEC'06).

[7] INCITS/ISO/IEC 27001-2005, Information technology - Security techniques - Information security management systems Requirements, developed by incites.

[8] R. Anderson. Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley, 2001.

[9]J. Viega, G. McGraw, Building Security Software. Addison-Wesley, New York, 2001.

[10]B. Kitchenham, S. L. Pfleeger, Software Quality: The Elusive Target. IEEE Software 13(1): 12-21 (1996).

[11] R. Savola, Towards a Security Metrics Taxonomy for the Information and Communication Technology Industry, International Conference on Software Engineering Advances (ICSEA 2007)

[12] F. Deissenboeck, S. Wagner, M. Pizka, S. Teuchert, J.-F. Girard An Activity-Based Quality Model for Maintainability, ICSM '07: Proc. of the 23rd IEEE International Conference on Software Maintenance, 2007.