Security Evaluation

Prince Thomas

University of Stuttgart
Institute of Software Technology (ISTE)
70569 Stuttgart, Germany

Abstract. Software security is an idea implemented to secure software against malicious attack and other hacker risks so that the software continues to function correctly under such potential risks. Security is necessary to provide integrity, authentication and availability. The fast growth rate of software and software products makes the software security aspect even more critical. Most organizations these days want their information system to be managed as safely as possible. Security Evaluation is the basic step in achieving this goal for any organization. It is of particularly important because of the rapidly changing environment of the information security system or the operation system. In this survey we are performing a study on Software Security Evaluation techniques. A detailed analysis of Qualitative and Quantitative Security Evaluation approaches are being carried out. The suitability and challenges of different methods of each of this approach is studied. The systems where these techniques are being used are investigated to understand the performance of security evaluation methods.

Keywords: Sotware, Security, Security Evaluation, Security Metrics, Model-Based Metrics, Quantitative Security Evaluation , Qualitative Security Evaluation

1 Introduction

Software security is the idea of engineering software so that it continues to function correctly under malicious attack [10]. The fast-growing software systems and huge amount of data handling makes the software security an important aspect in modern software development. Measuring and assessing software security is a critical concern as it is undesirable to develop risky and insecure software.

Nowadays we are hearing news about the new and new security threats every day. One such case of security threat was the phishing mail incident happened with the Twitter users on the weekend of January 3, 2009, several users on the social network website, Twitter, became victims of a phishing attack. The users were deceived into giving away their passwords when they received an e-mail similar to one that they would receive from Twitter with a link that read, "hey, check out this funny blog about you..". The link redirects to a site misguiding as the real Twitter site. Any personal information entered by the user on the fake site is then captured by the attacker. Generally, most of security incidents are caused by software flaws and bugs called security holes and vulnerabilities. Normally a hacker tries to find and exploit security holes present in the software. He does not create security holes on his own. As a result, ensuring software security has become so critical. Software security has to be evaluated to make sure that software is minimally susceptible to threats. Evaluation of software security is so challenging because of the non-predictability of the threats and attacker behaviors.

In this paper a detailed survey of software security evaluation is done. First a brief description of software security metrics is presented. Then different security evaluation techniques based on qualitative and quantitative security evaluation methods are discussed. Qualitative evaluation methods identify and analyze common vulnerabilities and the probability and damage of risks are evaluated qualitatively. Quantitative evaluation methods are model-based and can be described mathematically. The rest of this paper is organized as follows. Section 3 provides a brief overview of Software Security Metrics, their importance and the classifications . Section 4 describes the qualitative security evaluation. Section 5 the quantitative security evaluation and finally the paper is concluded with section 7.

2 Research Method

The research method I have chosen to perform the survey on security evaluation is literature review. I have attempted to find as many literatures which deal with security evaluation as possible. I have used mainly Google Scholars to search for the relevant papers for my study. The keywords used for the search are "Security Evaluation", "Security Metrics", "Quantitative Security Evaluation" and "Qualitative Security Evaluation".

3 Software Security Metrics

Metrics is a measurement standard which defines what is to be measured, how to be measured and helps the security practitioners to manage the product efficiently. Security metrics is the powerful tool that helps security practitioners to integrate security features into their system. The security metrics are gaining lot of significance recently because with the help of the data obtained from the software security decisions can be taken and which in turn helps the software developers to secure their software product.

3.1 Importance of Software Security Metrics

Security metrics help in decision making regarding security-related attributes of a process, system, or organization. In particular, security metrics can be applied to compare the effectiveness of different security mechanisms, or to indicate the degree to which security requirements of an organization are being met.

In addition, they can also be used to systematically improve the security level of a system, or to predict this security level in a future point in time. All the people involved in the software life cycle from developers to users use the security metrics for different use cases. For example Technical Personnels(Developers) use security metrics to decide which configuration change is the most effective to increase network resilience, Management members for financial investment on security and finally end users for the trustworthiness of a software products available in the market.

The desired properties of a good security metric are granularity, availability, cost effectiveness, localization and validation [12].

3.2 Classification of Security Metrics

The security metrics can be classified as Fig. 1.

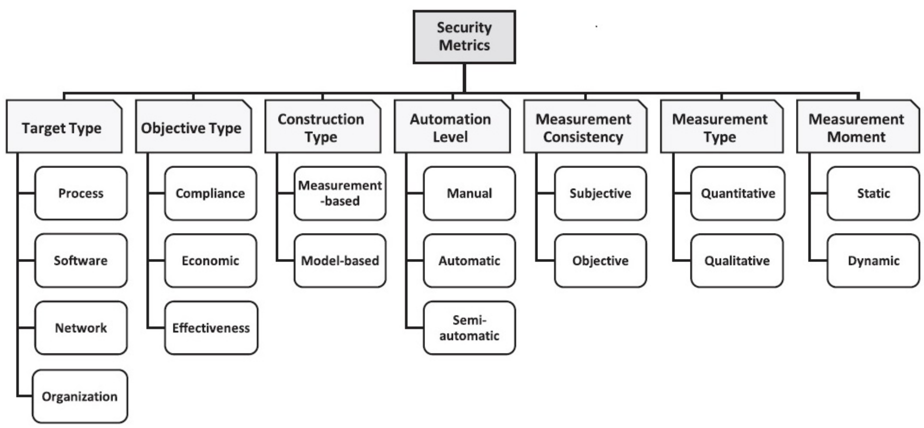


Fig. 1: Classification of Security Metrics [12].

(a) **Target Type:** Security metrics can be categorized according to the target they evaluate. The most common targets assessed (and respective security metrics) are the following:
  (i) **Process:** Process security metrics quantify the security level of a product by assessing its associated development process.
  (ii) **Software:** Software security metrics evaluate software security by assessing source code defects, software (mis)configuration, or other vulnerabilities present in software components.
  (iii) **Network:** Network Security Metrics(NSMs) assess the security of entire networks or parts thereof.
  (iv) **Organization:** Organization security metrics evaluate the physical and personnel security of an organization.

(b) **Objective Type:** Based on its objective type security metric can be classified as:
   (i) **Compliance:** Compliance security metrics measures how well the security requirements of a target is being met based on the security methods and policies.
   (ii) **Economic:** Metrics taking into consideration of the financial aspects of security.
   (iii) **Effectiveness:** It measures how effectively the security measures can perform against security threats or violations.

(c) **Construction Type:** Based on the way security metrics is derived, they can be classified as:
   (i) **Measurement-based:** This security metric is used to quantify the security property that is being measured.
   (ii) **Model-based:** Here the metrics values are derived from the complex mathematical equations used to define the formal mathematical model of the target. Refer Fig. 2 for a simple representation of a model-based security metric. Examples of models used to evaluate security metrics are attack graphs, Markov models, attack trees, Bayesian networks, etc. In the following sections the detailed evaluation of model-based security metrics is described.

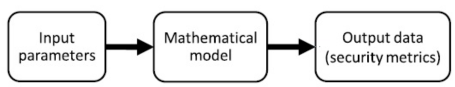


Fig. 2: Evaluation Process of Model-Based Security Metrics [12].

(d) **Automation Level:** Based on the level of automation used for the measurements the security metric can be classified as:
   (i) **Manual:** The collection of metrics values are being carried out manually(by humans).
   (ii) **Automatic:** Metrics values are being collected with the help of computer system without the intervention of humans.
   (iii) **Semi-Automatic:** Measurement is carried out with the help of both humans and computer systems.

(e) **Measurement Consistency:** Corresponding to the consistency of the metric values, security metrics can be classified as:
   (i) **Subjective:** Security metric is subjective to the person performing the metric measurement. Different people evaluating the same security property using same method can produce different results.
   (ii) **Objective:** Same result is being obtained irrespective of the person performing the evaluation.

(f) **Measurement Type:** According to the type of the measurement security metrics can be classified as:

(i) **Quantitative:** Quantitative security metrics are expressed as percentages or cardinal numbers (i.e., numbers that count something, instead of ordinal numbers, which only denote the position occupied by a given object).
(ii) **Qualitative:** Qualitative security metrics are expressed by labels such as high-medium-low values. Ordinal numbers can also be regarded as qualitative values.

(g) **Measurement Moment:** Depending on the instance of time at which the security metrics are applied to assess a given target, they can be classified as either static or dynamic:
(i) **Static, or pre-deployment:** Security metrics are developed to be measured before the assessed target enters operation.
(ii) **Dynamic, or run-time:** Security metrics are those developed to be constantly measured, during the operation of the target being evaluated.

4 Qualitative Evaluation

'Qualitative': means "involving distinctions or involving comparisons based on qualities". Qualitative security evaluation methods identify and analyze common vulnerabilities and the probability and damage of risks are evaluated qualitatively. The below subsections provide an overview about security patterns, criteria for qualitative evaluation and finally describes Vulnerability Assessment and Penetration Testing (VAPT), one of the qualitative security evaluation method.

4.1 Security Patterns for Qualitative Evaluation

In case of qualitative security evaluation different security patterns are evaluated for the security evaluation. A short description of different security patterns and their objectives are described below [6].

(a) The objective of the **Check pointed System pattern** is to organize a system so that its state can be recovered and restored to a known valid state, in case of a component failure.
(b) The objective of the **Standby pattern** is to organize a system so that the service provided by one component can be resumed from a different component.
(c) The objective of the **Comparator-Checked Fault Tolerant System pattern** is to organize a system, so that an independent failure of one component (i.e. a failure of a component that does not affect other components at all) will be detected quickly and so that an independent single-component failure will not cause a system failure.
(d) The objective of the **Replicated System pattern** is to organize a system that allows provision from multiple points of presence and recovery, in the case of failure of one or more components or links.

(e) The objective of the **Error Detection/Correction pattern** is to add redundancy to data (data replication) to facilitate later detection of and recovery of errors.

(f) The objective of the **Protected System pattern** is to organize a system so that all access by clients is mediated by a guard that enforces a security policy.

(g) The objective of the **Policy pattern** is to isolate policy enforcement to a discrete component of an information system and to ensure that policy enforcement activities are performed in the proper sequence.

(h) The objective of the **Authenticator pattern** is to perform authentication of a requesting process, before deciding access to distributed objects.

(i) The objective of the **Subject Descriptor pattern** is to provide access to security-relevant attributes of an entity, on whose behalf operations are to be performed.

(j) The objective of the **Secure Communication pattern** is to ensure that mutual security policy objectives are met, when there is a need for two parties to communicate in the presence of threats.

(k) The objective of the **Security Context pattern** is to provide a container for security attributes and data relating to a particular execution context, process, operation or action.

(l) The objective of the **Security Association pattern** is to define a structure which provides each participant in a Secure Communication with the information it will use to protect messages to be transmitted to the other party.

(m) The objective of the **Secure Proxy pattern** is to define the relationship between the guards of two instances of Protected System, in the case when one instance is entirely contained within the other.

4.2 Qualitative criteria for evaluation of security pattern

Researchers in [6] have demonstrated three set of qualitative criteria for evaluation of security pattern. The first category is based on 10 guiding principles for building secure software by Viega and McGraw (2002) [4]. The 10 principles are :

1. Secure the weakest link.
2. Practice defense in depth.
3. System should fail securely.
4. Follow the principle of least privilege.
5. Compartmentalize system.
6. System should be simple.
7. Promote privacy.
8. Remember that hiding secrets is hard.
9. Reluctant to trust.
10. Use community resources (well-tested solutions).

The second set of criteria focuses on software development problems which can lead to software security holes and are buffer overflows, poor access control mechanisms and race conditions.The last set of criteria can be described as how well a specific security pattern might respond to different categories of attacks as they are described by Howard and LeBlanc (2002) [7]. The model purposed by Howard and LeBlanc is called **STRIDE** model. The different attacks are **S**poofing identity attacks, **T**ampering with data attacks, **R**epudiation attacks, **I**nformation disclosure attacks, **D**enial of Service (DoS) attacks and **E**levation of privilege attacks.

4.3 Vulnerability Assessment and Penetration Testing (VAPT)

Vulnerability assessment is the process of scanning the system or software or a network to find out the weakness and loophole in that. These loopholes can provide back-door for the attacker to attack the victim. A system may have access control vulnerability, Boundary condition vulnerability, Input validation vulnerability, Authentication Vulnerabilities, Configuration Weakness Vulnerabilities, and Exception Handling Vulnerabilities etc. [5].

Penetration testing is the next step after vulnerability assessment. Penetration testing is the process of exploiting the system in an authorized manner to find out the possible exploits in the system. In penetration testing, the tester has the authority to do penetration testing and intently exploit the system and find out possible exploits. The life cycle of VAPT shown in Fig. 3.

The different vulnerability assessment techniques are **static analysis:** analyze the code structure and contents of the system , **manual testing:** tester uses his own knowledge and experience to find out the vulnerabilities, **automated testing:** use automated vulnerability testing tools to find out vulnerabilities in the system and **fuzz testing:** inputs invalid or any random Data into system and then look for crashes and failure.

The different Penetration testing techniques are **Black box testing:** the tester do not have prior knowledge about the network architecture or systems of the testing network, **Grey box testing:** tester has partial knowledge about the system and **White box testing:** tester has complete knowledge about the system.

There are different VAPT tools available in the market for doing VAPT assessment. Few examples are MBSA (vulnerability scanner for windows), AppScan (web vulnerability scanner for windows)and w3af (web vulnerability scanner for cross-platform).

4.4 Case study : Evaluation of web security mechanism using vulnerability and attack Injection (VAIT)

Here evaluation of security mechanisms in the context of web applications is described using the above mentioned qualitative security evaluation. Most common vulnerabilities in web applications was presented in a field study that classified

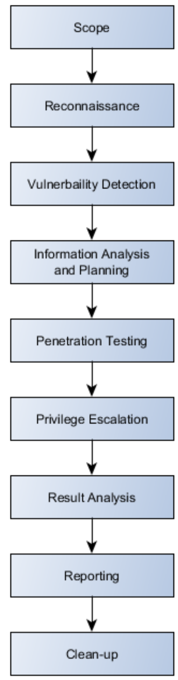
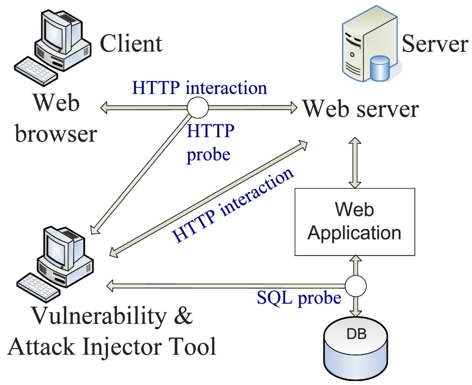


Fig. 3: Vulnerability Assessment and Penetration Testing Life cycle [5].

655 XSS and SQLi security patches of six widely used Linux, Apache, MySQL and PHP (LAMP) web applications [3].

The typical VAIT set up for security evaluation in case of a web application is given in the Fig. 4(a)

The automated attack of a web application is a multistage procedure that includes: preparation stage, vulnerability injection stage, attack load generation stage, and attack stage. The collection of information about the web application pages and their links can be done manually or using a web crawler for the preparation stage. In order to keep the same conditions for all the applications analyzed all the tests were done using the same web crawler (**Acunetix** web vulnerability scanner). Refer the Fig. 4(b) for results of the security evaluation of target web applications. The tool took approximately 11 minutes in the attack stage of the TikiWiki, 12 minutes in the phpBB and 4 minutes in the MyReferences. The vulnerabilities injected represent all the "Missing Function Call Extended" SQLi types that can realistically be injected into the files used in the experiments. On average, the tool injected one vulnerability for every 129 lines of PHP code. Only 20 percent out of all the vulnerabilities injected could not be attacked. From the results its concluded that tool is effective in providing a sufficient number of realistic vulnerabilities in a web application and that these vulnerabilities can be successfully attacked.



(a) VAIT in a typical setup

Web apps.	Files attacked	Code lines	Vuln. injected	Attacks	Attacks successful	Vulnerabilities attacked successfully
TikiWiki	`tiki-editpage.php`	904	3	84	34	3
	`tiki-index.php`	648	1	7	6	1
	`tiki-login.php`	305	3	21	0	0
	Total	1857	7	112	40 (36%)	4 (57%)
phpBB	`search.php`	1405	3	42	42	3
	`login.php`	224	1	21	21	1
	`viewforum.php`	694	1	7	7	1
	`viewtopic.php`	1210	5	84	84	5
	`posting.php`	1106	4	112	112	4
	Total	4639	14	266	266 (100%)	14 (100%)
MyRefs	`edit_paper.php`	310	27	525	61	20
	`edit_authors.php`	169	6	196	46	5
	Total	479	33	721	107 (15%)	25 (76%)
	Grand total	**6975**	**54**	**1099**	**413 (38%)**	**43 (80%)**

(b) Attack Injection Results of the Web Applications Analyzed

Fig. 4: VAIT in a typical setup (a) and Attack Injection Results of the Web Applications Analyzed (b) [3].

5 Quantitative Evaluation

'Quantitative' - means 'that is or may be estimated by quantity'. Quantitative evaluation produces quantitative results which can be easily interpreted mathematically. Due to the similarities between security and dependability, the techniques for model-based techniques applied to the dependability domain has been adapted to evaluation of security. The difference is that dependability metrics measure the effect of natural or accidental failures whereas the security evaluation metrics measure the impact of intentional failures caused by attacks. A detailed discussion on the adaptation of model-based metrics from dependability to security is provided in [11]. Dependability attributes include: **Reliability** - continuity of service , **Safety** - non-occurrence of catastrophic consequences, **Maintainability** - ability to undergo repairs and evolutions, **Availability** - readiness for usage, **Integrity** - data and programs are modified or destroyed only in a specified and authorized manner and **Confidentiality** - sensitive information is not disclosed to unauthorized recipients. Security evaluation is mainly concerned with primarily evaluating the last three attributes. The major quantitative evaluation methods are combinatorial methods adapted from dependability domain and state-based stochastic methods.

5.1 Combinatorial methods

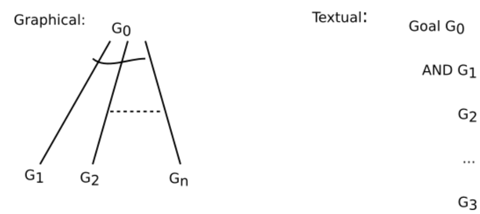
These methods are used to evaluate system dependability measures. Reliability Block diagrams (RBD) and Fault Trees (FTs) are the typical combinatorial methods used in dependability analysis. These methods can be adapted to security domain like the method Attack Trees.

Attack Trees are used to evaluate security of the system. Attack trees are closely related to fault trees, they consider the security breach as a failure and illustrates the group of events that can cause the system failure in a combinatorial way. Attack tree models all possible attacks against a system. Its a formal method which describes security of systems and subsystems based on numerous attacks.

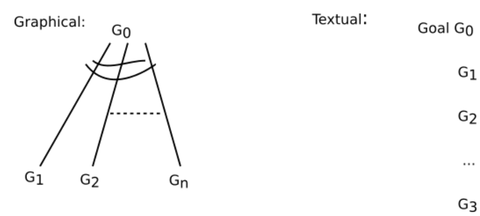
Structure of an Attack Tree: In attack trees the attacks of the systems are represented in the form a tree with root node represents the goal of attack and leaf nodes as the different ways to achieve the goal (atomic attacks). There are two kinds of root nodes AND nodes and OR nodes. An AND node represents an attack goal for which a set of subgoals (represented by leaf nodes) must be achieved in order for the attack to succeed. While an OR node represents an attack goal that can be achieved in several ways, which are represented by the OR node's children. Representation of an 'AND node' and 'OR node' is given in Fig. 5. As shown in figure in case of an AND node goal G_0 can be achieved if the attacker achieves each of G_1 through G_n and in case of an OR node goal G_0S can be achieved if the attacker achieves any one of G_1 through G_n.

Evaluating Attack Trees: The leaf nodes can take Boolean value(e.g., possible versus impossible attacks) or Continuous value (e.g., probability that the attack will succeed/fail). A node's value is a function of its children's values. If the assigned value is Boolean then AND node's value is the Boolean and of all values of its children and OR node's value is the Boolean or of all values of its children. When cost value is considered, the value of the AND node is the sum of the values of its children, and the value of the OR node is the minimum of the values of its children. The attack tree can be used to evaluate different aspects of the system security, depending on the kind of value that is assigned to the leaf nodes.

Attack trees thus help in providing a systematic way to describe the security vulnerabilities and finally for the security assessment for making security decisions. The attack tree for a larger system can be modeled combining attack trees derived from different security features of the system.



(a) And node



(b) OR node

Fig. 5: AND node (a) and OR node (b) [11].

5.2 State-Based Stochastic Methods

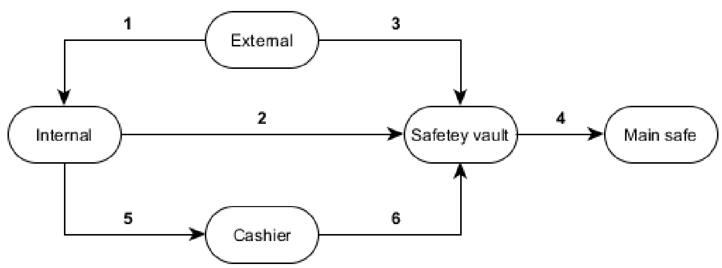
In such models, the system is expressed as a finite state machine. Each state of the system represents the security operational mode of the system. The states are modeled as one or more good states and one or more failure states. Good states are those in which the system is able to deliver the required services, even in the presence of attacks. Whereas, in security failure states, the system has been compromised by attacks in a way that its intended services can no longer be delivered. During the operation of the system continuously alternates between several possible states. A state transition is triggered by the appearance or remediation of a vulnerability i.e. the successful execution of an attack step or the responsive action performed by the security mechanism.

State-based metric models the system using Continuous-Time Markov Chain (CTMC) with discrete state space. CTMC is based on Markov property. As per Markov property, the probability distribution of the next state depends only on the current state and not on the previous states. Also, the time spent in each state takes a continuous set of values (i.e., non-negative real values), and follows an exponential distribution.

The major quantitative models based on stochastic models are described below.

(a) **Time-Based Security Metrics**
Its based on the amount of time to make the system to be compromised by a successful attack. Larger the time value more secure the system is. One of the time based Metrics is based on Privilege Graphs.
Privilege Graph-Based Metrics
To evaluate system security the stated-based Markov model proposed by

Dacier et al. [2] based on the Mean Time to Failure (MTTF) metric is being used. In their model with the help of a privilege graph the vulnerabilities of a system is represented. In this graph nodes represent the access rights (privileges) at hosts and arcs represent the actions that enable the transition from one privilege to another. Each arc is labeled with a value (λ) that represents the success rate of an elementary attack (privilege escalation). Once the privilege graph is constructed, it is turned into a CTMC containing all possible attack paths to target nodes. Refer Fig. 6 for an example of a privilege graph. The Markov model is based on the assumption that the success probability of an elementary attack before time t is represented by an exponential distribution given by: $P(t) = 1 - e^{-\lambda t}$. The mean time of succeeding in an elementary attack is given by $1/\lambda$. The MTTF of the system is computed by aggregating all the mean times necessary for succeeding in elementary attacks that lead to the targets.



(a) 1) Forge an employment card; 2) Forge an authorization; 3) Breaking and entering; 4) Pick the lock; 5) Standing in; 6) Manger delegation.

Fig. 6: Example of a privilege graph.

Even though MTTF provide useful security level information, they do not take into account of partial or interrupted attacks (which can also cause damage to the system).

(b) **Probability-Based Security Metrics**
Probability-based security metrics normally express the likelihood of a threat compromising the system or the probability that the system is secure.

Li et al. [9] presents a renewal stochastic model to estimate the likelihood that an adversary exploits a randomly selected system vulnerability. This likelihood is given by the q metric namely the probability that a randomly picked vertex is compromised when the system enters its steady state. A lower q value indicates more security. To compute q, the network is represented as a vulnerability graph, in which every node represents a vulnerability, and the arc or the edge shows that the exploitation of one vulnerability could lead to the exploitation of the other. At any given instant of time, each node can either be in a secure state or a compromised state. The behavior of the nodes is described by a series of random variables that represent compromise rates and fix rates. Refer Fig. 7 for a probability security model.

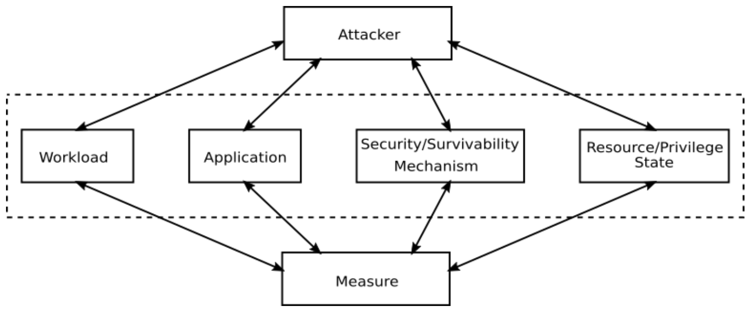


Fig. 7: Probabilistic security model structure [11].

5.3 Attack Graphs

Attack graphs depict ways in which an adversary can exploit vulnerabilities to break into a system. Attack graph can be automatically to generate attack paths to analyze the network vulnerability. It can show users the weak point in the network analysis process for network security risk analysis. Once a potential attack path is found, attack graph tools can generate attack graph or attack trees to help system administrators understand how attacks happen, and then take defensive measures. There are many tools to automatically build attack graphs for a given network. For example open source tools like MulVAL, TVA. Attack Graph Toolkit, NetSPA and the commercial tools like Cauldron, FireMon, Skybox View.

The major quantitative models based on attack graphs are described below.

1. **Path Metrics**
   Its based on the characteristics of the attack paths. One of the path metrics is Shortest Path Metric (SP) proposed by Phillips and Swiler [1]. As per this metric, the network is modeled as a condition-oriented network graph and security level corresponds to the length of the smallest attack path that an adversary can take to reach the desired goal state(compromised state). So longer the shortest path, more secure the network. The advantage of SP metric is simplicity however it does not take into account for the number of different shortest paths in a network. To overcome this researchers have come up with different path metrics like Number of Paths metric (NP), Mean of Path Lengths metric (MPL).

2. **Non-Path Metrics**
   Its based on the number of hosts that can be compromised. Pamula et al. [8] propose the Weakest Adversary metric, which is based on a condition-exploit-oriented attack graph. This metric illustrates the security strength of a network by means of the least amount of effort an attacker needs to compromise a given network asset. The authors consider the minimum effort expended by an attacker as the strength of the set of initial conditions (of an attack graph) that enable the compromise of a network. Therefore, when comparing two networks, the less secure network will be the one with the weaker set of initial conditions.
3. **Probabilistic Metrics**
   Probabilistic AG-based metrics either take probability values as input, or produce probability values as output, or both. The two common Probabilistic metrics are PageRank-Based Metric : This metric measures the probability of goal states occurring (i.e., probability of the network being compromised) and Metric based on Independent Attack Paths : Uses an exploit dependency graph to quantify network security by propagating exploit likelihood scores from initial exploit to final exploit(compromised state).
4. **Bayesian Network Based Metrics**
   A Bayesian Network (BN), also known as belief network, can be defined as a directed acyclic graph (DAG) with nodes representing variables of a system and edges representing causal relationships among these variables. In BN each node is assigned with Conditional Probability Table (CPT). Theses values indicate the conditional probabilities of each vulnerability being exploited.

5.4 Common Vulnerability Scoring System

The Common Vulnerability Scoring System (CVSS) is a frame work used to quantify the severity and risk of a vulnerability to an information asset in a computing environment.It was designed by NIST (National Institute of Standard and Technology) and a team of industry partners. A CVSS score is a decimal number in the range from 1 to 10. The security experts assign set of quality values to predefined attributes of given vulnerability and CVSS score for the desired vulnerability is calculated using the same. For every qualitative value of a given attribute has a predefined corresponding quantitative value. Using these quantitative values final CVSS score is calculated. The more vulnerabilities a software product has, the lower level of trustworthiness this software product has. The more severe vulnerabilities a software product has, the less secure this software will be.CVSS metrics for vulnerabilities are divided into three groups: Base metrics measure the intrinsic and fundamental characteristics of vulnerabilities that do not change over time or in different environments. Temporal metrics measure those attributes of vulnerabilities that change over time but do not change among user environments. Environmental metrics measure those vulnerability characteristics that are relevant and unique to a particular user's environment. Presently several vulnerability databases use CVSS to quantify the severity of reported vulnerabilities.

In the paper, [13] applied CVSS to find out the security metrics for the software products. They had applied the derived security metrics formula for the Internet applications such as Mozilla Firefox 2, Microsoft Internet Explorer 6 and Microsoft Internet Explorer 7 to find out their vulnerabilty scores and finally the security metric. The final security metric scores for applications were Mozilla Firefox 2 = 6.7, Microsoft Internet Explorer 6 = 8.7 and Internet Explorer 7 = 7.9.

6 Evaluation of Qualitative and Quantitative Methods

In this section finally a quick recap of qualitative and quantitative evaluation methods are presented. Refer Table 1 for the glimpse of summary of these methods. The major advantages of qualitative method are it allows for putting in order risks according to priority, allows for determination of areas of greater risk in a short time and without bigger expenditures and analysis is relatively easy and cheap. However it does not allow for determination of probabilities and results using numerical measures and achieved results have general character, approximate etc.

The major advantages of quantitative methods are they allow for definition of consequences of incidents occurrence in quantitative way, what facilitates realization of costs and benefits analysis during selection of protections and they give more accurate image of risk. However quantitative measures depend on the scope and accuracy of defines measurement scale, results of analysis may be not precise and even confusing and these methods are generally more expensive, demanding greater experience and advanced tools.

7 Conclusions

In this paper detailed analysis of security evaluation is presented. Qualitative and Quantitative evaluation methods are discussed. Qualitative evaluation methods are focused on detection and prevention of vulnerabilities while the Quantitative methods provide a characterization of security risks in terms of vulnerabilities present in the software. Such characterization of vulnerabilities can provide us metrics that can be used by the developers and potential users. Results of these evaluation methods are useful in developing guidelines for allocation of resources for security testing, scheduling, and development of security patches. Furthermore, it can be used by the users for assessing risk and estimating needed redundancy in resources and procedures to handle potential breaches. Security evaluation can be useful to know existing threats and potential vulnerabilities of your system, e.g., to avoid them in future systems.

In this survey only a few of the security evaluation methods are discussed. But there are so many security evaluation methods and the major challenge would be choosing the right security evaluation method for the respective software system.

Table 1: Summary of qualitative and quantitative methods

Qualitative Evaluation	Quantitative Evaluation
Putting in order risks according to priority	Definition of consequences of incidents occurrence in quantitative way
Determination of greater risks in short time	Facilitates realization of costs and benefits analysis during selection of protections
Analysis is relatively easy and cheap	More accurate image of risk
Lack of numerical measures for analysis	Depend on the scope and accuracy of defined measurement scale
Does not facilitate realization of costs and benefits analysis during selection of protections	Analysis results may not be precise and even confusing
Results have general character, approximate etc.	More expensive, Requires greater experience and advanced tools

This is often a difficult task. Usually for the security evaluation of software systems (Web applications, Network management) different security evaluation methods are applied, their results are compared and the best among them is chosen. In case of complex systems several methods are combined to produce a better result. This makes the job of security expert challenging. The area of security evaluation is growing rapidly and lately even machine learning concepts are used for the same.

References

[1] L. P. S. Cynthia Phillips. A graph-based system for network-vulnerability analysis. In *Workshop New Security Paradigms (NSPW)*, pages 71–79, Charlottesville, Virginia, USA, 1998. doi: 10.1145/310889.310919.

[2] M. Dacier, Y. Deswarte, and M. Kaâniche. Quantitative assessment of operational security : Models and tools. 1996.

[3] J. Fonseca, M. Vieira, and H. Madeira. Evaluation of web security mechanisms using vulnerability amp; attack injection. *IEEE Transactions on Dependable and Secure Computing*, 11(5):440–453, Sept 2014. ISSN 1545-5971. doi: 10.1109/TDSC.2013.45.

[4] J. V. Gary McGraw. *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison-Wesley Professional, 2002.

[5] J. N. Goel and B. Mehtre. Vulnerability assessment & penetration testing as a cyber defence technology. *Procedia Computer Science*, 57:710 – 715, 2015. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2015.

07.458. URL http://www.sciencedirect.com/science/article/pii/S1877050915019870. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

[6] S. T. Halkidis, A. Chatzigeorgiou, and G. Stephanides. A qualitative analysis of software security patterns. *Computers & Security*, 25(5):379 – 392, 2006. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2006.03.002. URL http://www.sciencedirect.com/science/article/pii/S0167404806000526.

[7] M. Howard and D. E. Leblanc. *Writing Secure Code*. Microsoft Press, Redmond, WA, USA, 2nd edition, 2002. ISBN 0735617228.

[8] P. A. J. Pamula, S. Jajodia and V. Swarup. A weakestadversary security metric for network configuration security analysis. In *2nd ACM Workshop Qual. Protect. (QoP)*, pages 31–38, Alexandria, VA, USA, 2006.

[9] X. Li, P. Parker, and S. Xu. A stochastic model for quantitative security analyses of networked systems. *IEEE Transactions on Dependable and Secure Computing*, 8(1):28–43, Jan 2011. ISSN 1545-5971. doi: 10.1109/TDSC.2008.75.

[10] G. McGraw. Software security. *IEEE Security Privacy*, 2(2):80–83, March 2004. ISSN 1540-7993. doi: 10.1109/MSECP.2004.1281254.

[11] D. M. Nicol, W. H. Sanders, and K. S. Trivedi. Model-based evaluation: from dependability to security. *IEEE Transactions on Dependable and Secure Computing*, 1(1):48–65, Jan 2004. ISSN 1545-5971. doi: 10.1109/TDSC.2004.11.

[12] A. Ramos, M. Lazar, R. H. Filho, and J. J. P. C. Rodrigues. Model-based quantitative network security metrics: A survey. *IEEE Communications Surveys Tutorials*, 19(4):2704–2734, Fourthquarter 2017. ISSN 1553-877X. doi: 10.1109/COMST.2017.2745505.

[13] J. A. Wang, H. Wang, M. Guo, and M. Xia. Security metrics for software systems. In *Proceedings of the 47th Annual Southeast Regional Conference*, ACM-SE 47, pages 47:1–47:6, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-421-8. doi: 10.1145/1566445.1566509. URL http://doi.acm.org/10.1145/1566445.1566509.

Review Comments Prince Thomas

---------------------- REVIEW 1 ---------------------

PAPER: 10

TITLE: Security Evaluation

AUTHORS: Prince Thomas

Overall evaluation: 0 (borderline paper)

----------- Overall evaluation -----------

Summary:

Chapter 2 gives an idea and insights about security metrics and why these are important to use. The next chapter describes security patterns and their evaluation. It also describes methods and tools to test vulnerabilities in software. To wrap it up a case study is presented which show some of the previously described methods. The last chapter then the quantitative evaluation. Different methods for systematic software penetration or attacks are discussed, as well as their advantages and disadvantages.

Positive aspects:

The paper gives a nice overview on methods and tools for software security evaluation.

Its logically structured.

General Issues:

Some things which could be a list are in continuous text which makes it hard to read, because of all the commas, hyphens and colons.

Some sentences tend to get long (3-4 lines).

Prince Thomas Comments: Fixed most of that's possible.

Specific Issues:

Goals/Aims:

The goal is to give a deeper insight into software security evaluation and its aspects. Explaining challenges and reason why doing security evaluation.

Presentation:

Title: Maybe be a little more specific, 'Security Evaluation' of what or in which field?

Prince Thomas Comments: It's for software field, but since this paper is a part of non-functional aspects of software engineering, I didn't mention it specifically..

Abstract: Good

Key Words: Maybe add 'Software' somewhere.

Prince Thomas Comments: Taken care.

Introduction: Good

Conclusion: In the second paragraph: 'There are so many [...]' -> bad phrasing. Due to all the enumerations it is hard to get into a 'reading flow'.

Prince Thomas Comments: Taken care.

Length:

3.5 and 4.5 could be a little bit longer and more detailed, since a lot of things are described in the corresponding chapters (3.1-3.4 and 4.1-4.4). Just to wrap it up.

Prince Thomas Comments: Those subsections are combined to a section to give a better clarity.

Research Methods: Clear

Clarity:

Figure 6 is hard to follow.

Prince Thomas Comments: Changed the representation with enough details.

Result Presentation:

Figure 3 is not a cycle as stated in the description.

Prince Thomas Comments: Taken care.

Figure 5 doesn't give me any information.

Prince Thomas Comments: Gives an idea about 'AND' and 'OR' nodes.

Citations:

Reference style is not consistent

Prince Thomas Comments: Because of different kind of refrences.

Goal/Aim reached:

Each chapter describes methods and concludes them. The conclusion could be a little bit longer (see Length).

Prince Thomas Comments: I feel its sufficient. I will discuss with supervisor and will change accordingly.

Other Issues:

In 3.2 make a list or similar for the 10 principles.

Prince Thomas Comments: Taken care.

Be consistent with enumerations. Currently some of them are continuous text some use bold text with '-' some with ':'.

Page 11, Sentence 1 doesn't make sense to me.

There are errors with spaces before/after colons/brackets/dots/...

Prince Thomas Comments: Taken care.

--------------------- REVIEW 2 --------------------

PAPER: 10

TITLE: Security Evaluation

AUTHORS: Prince Thomas

Overall evaluation: 2 (accept)

----------- Overall evaluation -----------

1. Summary of the paper

The paper talks about why software security is important (continue to function correctly under attack) and why is it important in modern software systems. The author emphasizes on significance of measuring and assessing software security, because secure software is a must. Furthermore, it elaborates on how security metrics aid developers in securing their software product. For example, which configuration change is the most effective to increase parameters relevant for the trustworthiness of a software products. It evaluates web security on the use case, where automated attack of web application is studied. It talks about what are software security metrics and security evaluation techniques. In the 2nd chapter it elaborates more on importance of Security Metrics and how are they are classified. Different quantitative Security evaluation techniques are presented such as Combinatorial methods and State-Based Stochastic Methods, where system is expresses as a finite state machine. Case study is given evaluating security mechanisms in the context of web applications.

2. Positive aspects

The language in the paper is comprehensive and easy to read. The author uses proper constructs and the work is not boring to read. The paper gives motivation why software security is an important aspect in Modern software. The rest of the paper follows and builds upon Abstract and Introduction subsection and gives the suggestions on how to tackle issues presented in these subsections.

3. General issues

Format: On page 5, "Static: Static, or pre-deployment, security metrics are developed to be measured before the assessed target enters operation. Dynamic: Dynamic, or run-time, security metrics are those developed to be constantly measured, during the operation of the target being evaluated." Looks better if you write "Static, or pre-deployment, ...", delete "Static: Static, or pre-deployment ..."

Prince Thomas Comments: Taken care.

Language: Abstract: The 2nd sentence sounds boring, since you used two times for the sentence subject the same word "Security Evaluation". "Security Evaluation basic step in achieving this goal for any organization. Security Evaluation is particularly important because of the rapidly changing environment of the information security system or the operation system."

Prince Thomas Comments: Taken care.

"Metrics is a measurement standard which denes what is to be measured, how to be measured and helps the security practitioners to manage the product efficiently. Security metrics is the powerful tool that helps security practitioners to integrate security features into their system. The security metrics are gaining lot of significance now a days because with the help of the data obtained from them software security decisions can be taken and which in turn helps the software developers to secure their software product."

Prince Thomas Comments: Taken care.

Same as in my first comment, you are using the same verb way too many times. Here in all 3 consecutive sentences you use verb aid".

Prince Thomas Comments: Taken care.

Check other grammar and spelling mistakes.

Prince Thomas Comments: Taken care.

Structure: In the Conclusions section, you should give a future work based on the work in your paper.

Prince Thomas Comments: Future scope of the work is mentioned in abstract manner.

4. Specific issues

Goals/Aims: In the Introduction section, you stated what is the current problem, however, you need to introduce the solution to this issue, to emphasize the relevance of your paper.

Prince Thomas Comments: It seems fine.

Length: In subsection 3.1 you mention all the Security Patterns for Qualitative Security Evaluation, instead you should explain only ones that are relevant to your methodology.

Prince Thomas Comments: All of them are measured to give the author an idea about the valuable security patterns which will help in exploring further.

Clarity: In subsection 3.2, rather than just citing and listing qualitative criteria for evaluation of security pattern, try to elaborate more on this topic. For example, try to extend the part which talks about what is significance of these qualitative criteria, why are they better than the others?

Prince Thomas Comments: Idea was just to give a short overview as it's not really part of the research.

Result presentation: You have figure 4, for your case study, however in the text of subsection 3.4 you should give more detail about this picture. Keep in mind that a reader should be able to find more details about your figure in the text.

On page 7, you wrote: "Penetration testing is the next step after vulnerability assessment.", however in Figure 3, Penetration Testing is the step after information Analysis and Planning.

Prince Thomas Comments: All the 4 steps constitute the part called vulnerability assessment.

Research methods: In subsection 3.4 give more information on how are results produced from the use-case scenario.

Citation: 1) On page 6: "Spyros T. Halkidis, Alexander Chatzigeorgiou and George Stephanides in their paper "A qualitative analysis of software security patterns" [6] had mentioned three set of qualitative criteria for evaluation of security pattern.", when you cite someone else's work you do not need to write all of the author names. Use something like Researchers in [6] have demonstrated that

Prince Thomas Comments: Taken care.

2) In subsection 4.2, you explain the definition of these methods, however, is this really your own idea, or it's a definition from some other book or paper. If not cite the source.

Prince Thomas Comments: Framed with my own words.

5. Other issues

Page 1, twitter should be written with capital t.

On page 2, "now a days" must be written together "nowadays". You also used this verb on page 1, try to use some other word instead.

"Brief description about software", this is wrong it "description about ..." is wrong, should be "description of ....".

"model based" is one word, should be "model-based".

"describes about the qualitative" should be without word "about": "describes the qualitative"

"from them software security" what is "them" trying to tell here? Did you mean to use "the"?

Prince Thomas Comments: All points were relevant and have been taken care.

---------------------- REVIEW 3 ---------------------

PAPER: 10

TITLE: Security Evaluation

AUTHORS: Prince Thomas

Overall evaluation: 2 (accept)

----------- Overall evaluation -----------

1. Summary of the paper:

The paper gives an overview about software security evaluation. It specifies the significance of software security and provides a detailed study on the security metrics used in the research to quantify the metrics for security evaluation. The paper elaborates further on qualitative and quantitative security evaluation approaches in detail.

2. Positive aspects:

The paper gives a very detailed overview of security evaluation. Qualitative and quantitative security evaluation approaches are explained clearly with case studies and examples.

3. General aspects:

1. Figures need to be in vector graphics format.

2. Capitalization of letters needs to be corrected at various places in the paper. (for example, "in Modern software development", "social network Web site", "Several vulnerability databases", "and Race conditions" etc)

Prince Thomas Comments: Taken care (All points).

4. Specific aspects:

1. Please provide a tabular comparison of evaluation in the results for better clarity.

2. "Representation of an AND node OR node is given in Fig.5." needs to be corrected.

3. In section 3.2, 10 guiding principles for building secure software are listed. This can be numbered to make it more readable.

4. In Section 4, please add a sentence on what is detailed in subsection of it to add better clarity, to describe the aspects considered in the evaluation and its relevance.

5. In Section 3, please add a sentence on what is detailed in subsection of it to add better clarity. For example, it was not clear on why the VAPT based security evaluation is performed.

Prince Thomas Comments: Taken care (All points).

5. Other issues:

1. Punctuation issues/Typo: "dynamic:"

2. Typo: "systematic ay" needs to be corrected to systematically.

Prince Thomas Comments: Taken care (All points).