S. K. Pandey, K. Mustafa / International Journal of Engineering Research and Applications (IJERA)
ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 1, Jan-Feb 2012, pp.1174-1178

Access Control and Rights related Risk Assessment

S. K. Pandey[1], K. Mustafa [2]
[1]Department of Information Technology
Board of Studies, The Institute of Chartered Accountants of India (Set up by an Act of Parliament), Noida- 201 309
[2]Department of Computer Science
Jamia Millia Islamia (Central University), New Delhi-110 025

ABSTRACT

Deployed software, now-a-days, are continuously under attack. Attackers have been exploiting vulnerabilities for decades and seem to be on rise everyday. Firewalls, intrusion detection and antivirus systems cannot simply solve this problem to the desirable extent. The major reason may be the in-built vulnerabilities, which are not curable at these stages. Only a concerted effort, by the software development community for building more secure software can foil attacks and allow users to feel protected from the exploitation. It is observed that each phase of the SDLC should include the appropriate security assurance mechanisms and countermeasures. From requirements through design and implementation, to testing and deployment, security measures must be embedded throughout the SDLC phases. 'Access Control and Rights' is one of the measure protective mechanisms, which is broadly accepted. Appropriate level of access control may well enforce security features and hence assure security. In this paper, various attributes of 'Access Control and Rights' are identified and then a rank/weight is assigned to each one, followed by the risk assessment to integrate steps for security assurance from the early in the development lifecycle.

Keywords: Software Security, Security Assurance, Access Control and Rights Policy, Attributes of Access Controls and Rights, Risk Assessment.

1- INTRODUCTION

Software security is not only a desirable but also an essential feature of software, to function correctly even under malicious attack. Most of the critical infrastructure is fairly complex, interconnected, and interdependent systems. A single programming or design flaw in today's complex software system may disturb the entire system. In 1990, failure due to a single line of buggy code in AT & T's 4ESS switch caused systems drop roughly 50% of long distance over a period of nine hours and $60 million loss [1][2]. Another incident of computer security reported to the CERT coordination center, in recent years due to a single class of programming flaws buffer overruns [3]. Software security is the foremost concern for modern information enterprises. Designing highly dependable security systems to ensure secure access for distributed software and information has been recorded as one urgent problem. Software security is about designing software to be secure, making sure that software remains secure, and guiding software developers, architects and users about how to build and maintain secure software.

Requirements are considered as the foundation stone on which the entire software is built. In earlier days, the requirements phase was not taken seriously, which caused many big software problems. These problems' nature and quality both continue to grow exponentially with the growth in software complexity and its versatility. The failure and success of any software depends upon the quality of requirements. It is observed that about 71% of the software is not completed due to poor requirements [4] [5] [6] [7]. Studies indicate that more than 60% failure rate for software projects in the US, with poor requirements as one of the top five reasons. Studies also show a high percentage of project schedules overruns, with 80% due to creeping requirements [8]. The importance of the requirements engineering has been well recognized and now many reversed researches are underway on *'ways to incorporate security right from the beginning'* and from the requirements phase itself.

The requirements phase is one of the foremost opportunities for the product team to consider 'how any security features can be integrated into a development process, identify key security objectives and otherwise maximize software security' [9]. In continuation to this process, the team needs to consider 'how the security features and assurance measures will integrate with other software likely to be used with it'. The requirement team's overall perspective of security goals, challenges, and plans need to be incorporated in the SRS that is produced during the requirement's phase.

Security policies are the most primitive to securing a system, organization or other entity. Different security policies can be implemented at the software level [10]. Mostly, these are traceable in the literature and reported practices, to one or more of the policies given in the Figure 1, as follows:

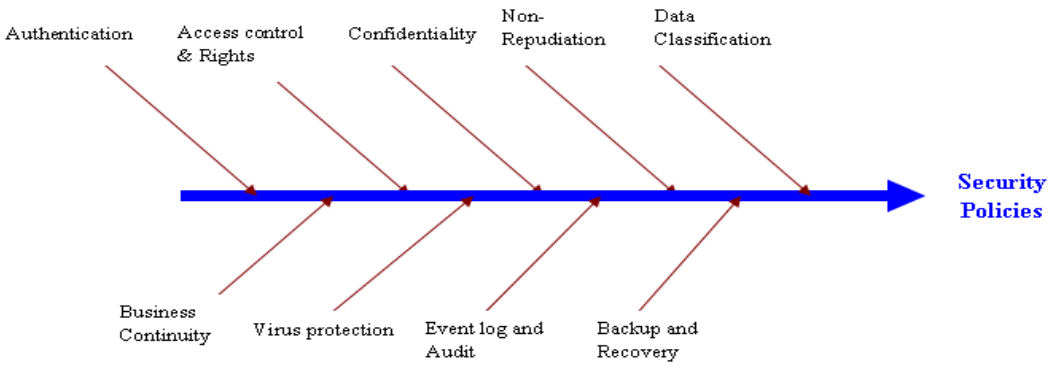

Figure 1. Security Policies

In this paper, we concentrate on 'Access Control and Rights' policy and risk assessment procedure. The purpose of this policy is to establish a standard for access control and rights of the users to the IT systems. The risk assessment activity is performed on the basis of various attributes identified for this policy. The remainder of this paper is organized as follows: Section 2 describes the 'Access Control and Rights' Policy. The attributes for 'Access Control and Rights' are discussed in Section 3 while a ranking/weight is proposed in Section 4. 'Risk Assessment' is discussed in Section 5, whereas 'Experimental Validation and Results' is given in Section 6. 'Conclusions and Future Work' are given in Section 7.

2- 'ACCESS CONTROL AND RIGHTS' POLICY

It refers to the ability to permit or deny the use of a particular resource by a particular entity [10]. Access control mechanisms can be used in managing the following but not limited to:

- Physical resources such as a movie theater, to which only ticket holders should be admitted;
- Logical resources such as a bank account, with a limited number of people authorized to make a withdrawal; and
- Digital resources for example, a private text document on a computer, which only certain users should be able to read.

In order to safeguard software systems, procedures are developed and implemented for protecting them from unauthorized modification, disclosure or destruction to ensure that information remains accurate, confidential, and is readily available when required [11]. The administration of user access to the software, applies the principles of least privilege and 'need to know' basis. Logical access level to software and information are restricted to users authorized by the respective Security Administrator. This policy should address the policies and procedures related to 'Access Control and Rights' of users to the organization's information resources. This policy should be adapted to all the users and the information resources including all operating systems, applications, databases, and other computing resources [10].

3- ATTRIBUTES OF 'ACCESS CONTROL AND RIGHTS' POLICY

Taking into account, the need and significance of an access control and rights policy for building secure software, various attributes of this policy are identified. These attributes have been derived from the reported and well-verified practices which is evident from our earlier publication [12]. A pictorial representation of these attributes is depicted as follows:

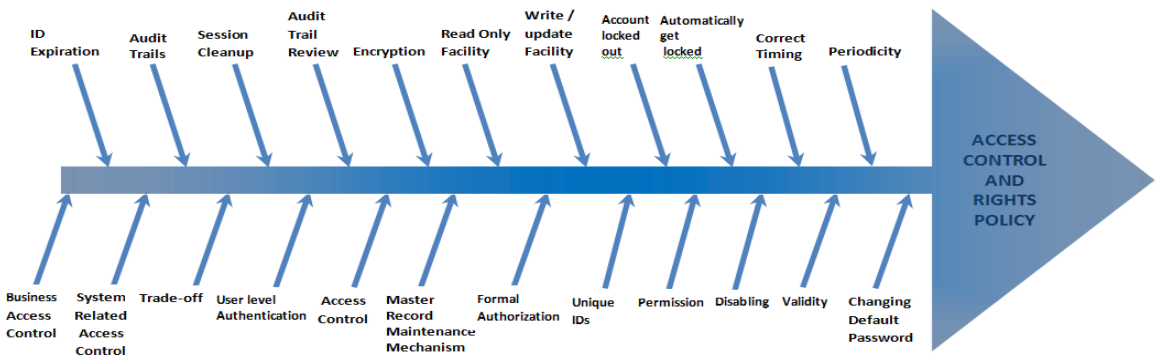

Figure 2. Attributes of Access Control and Rights Policy

4- RANKING / WEIGHT OF THE ATTRIBUTES

After proposing these attributes, we realized that each attribute may have its unique weight for the implementation of this security policy; that means the ranking / weight of all the attributes may not be the same rather it will be different. Therefore, it was decided to take the help and guidance of experts' feedback by designing a feedback form. The feedback was collected on the following issues:

- Analysis of the attributes' quality which include following heads:
  - importance of the attribute,
  - Potential utility for evaluation practice,
  - Completeness/coverage of attributes, and
  - Relevance of all the attributes, and
- In the rightmost column of each attribute, to assign a *weight between 1 and 5 (1 is minimum and 5 is maximum)* to each attribute for the implementation of this security policy.

These attributes along with the review form were sent to the thirty experts from the varied fields' viz. academia, industry, scientific organizations, educational institutions, research bodies, government organizations. Really, it was a daunting task to have the feedback from the experts. After a long exercise, we were able to have duly filled feedback forms from the twenty experts only. After collecting these forms/comments, we compiled this data in two ways. At the first level, based on the comments cited in the review forms, we made some revisions in the attributes and then again a fresh ranking was taken. On the second level, we designed a format in an excel sheet, in which all the data from the experts' comments were filled. Since, we received the feedback from twenty experts only; an average rank value of each attribute was calculated. Based on the average value of each attribute, we finalized the weight of the attributes of 'Access Control and Rights' policy which is displayed in the following table:

Table 1: Attributes' Weight of Access Control and Rights Policy

S. No	Attribute	Attribute's Weight
1.	*Business Access Control*	4.5
2.	*System Related Access Control*	4.4
3.	*Trade-off*	3.95
4.	*User level Authentication*	4.5
5.	*Access Control*	4.75
6.	*Master Record Maintenance Mechanism*	4.75
7.	*Formal Authorization*	4.55
8.	*Unique IDs*	4.15
9.	*Permission*	3.8
10.	*Disabling*	4.35
11.	*Validity*	4.65
12.	*Changing Default Passwords*	4.35
13.	*ID Expiration*	4.2
14.	*Audit trails*	4.4
15.	*Session Cleanup*	4.1
16.	*Audit Trail Review*	4.1
17.	*Encryption*	4.6
18.	*Read Only Facility*	4.25
19.	*Write/update Facility*	4.2
20.	*Account locked out*	4.05
21.	*Automatically get locked*	4.5
22.	*Correct Timing*	4.25
23.	*Periodicity*	4.3

5- RISK ASSESSMENT

After determining the weight of the attributes of the policy, we hereby propose the risk assessment procedure which can be done by using the following formula:

Risk = Policy [Attributes]

Risk for 'Access Control and Rights' Policy = $\sum$ Wi Xi / n where Xi = {1 or 0

and i = 1, 2, 3, ……n

Here, Wi is the weight of the attribute, and Xi is the value based on the satisfaction of this attribute i.e. if a attribute is satisfied, the value will be 1, and if not, its value will be 0.

Based on the above calculated risk value, its tolerance limit may be decided. We propose the following limits, as given:

- **Low Risk:** The implementation of this policy is at low risk if the value of the final risk value is ≥ 3.5.
- **Medium Risk:** The implementation of this policy is at medium risk if the value of the risk lies between 2.5 to 3.5.
- **High Risk:** The implementation of this policy is at high risk if the risk value is ≤ 2.5.

6- EXPERIMENTAL VALIDATION AND RESULTS

The proposed methodology is applied to a real life project from industry (on the request of the company, identity is concealed), and the final result of attributes' assessment is calculated on the bases of total checked, and unchecked points. The results are given in the following table:

Table 2: Validation Data for 'Access Control and Rights' Policy

S. No	Attribute	Attribute's Weight	Checked/Unchecked
1.	*Business Access Control*	4.5	Yes
2.	*System Related Access Control*	4.4	Yes
3.	*Trade-off*	3.95	No
4.	*User level Authentication*	4.5	Yes
5.	*Access Control*	4.75	Yes
6.	*Master Record Maintenance Mechanism*	4.75	Yes
7.	*Formal Authorization*	4.55	Yes
8.	*Unique IDs*	4.15	Yes
9.	*Permission*	3.8	Yes
10.	*Disabling*	4.35	No
11.	*Validity*	4.65	No
12.	*Changing Default Passwords*	4.35	No
13.	*ID Expiration*	4.2	No
14.	*Audit trails*	4.4	No
15.	*Session Cleanup*	4.1	No
16.	*Audit Trail Review*	4.1	No
17.	*Encryption*	4.6	No
18.	*Read Only Facility*	4.25	No
19.	*Write/update Facility*	4.2	No
20.	*Account locked out*	4.05	Yes
21.	*Automatically get locked*	4.5	Yes
22.	*Correct Timing*	4.25	No
23.	*Periodicity*	4.3	Yes

Now, as per formula given the above section,

Risk for 'Access Controls and Rights' Policy (ACRP) = (4.5 × 1) + (4.4 × 1) + (3.95 × 0) + (4.5 × 1) + (4.75 × 1) + (4.75 × 1) + (4.55 × 1) + (4.15 × 1) + (3.8 × 1) + (4.35 × 0) + (4.65 × 0) + (4.35 × 0) + (4.2 × 0) + (4.4 ×0) + (4.1 × 0) + (4.1 × 0) + (4.6 × 0) + (4.25 × 0) + (4.2 × 0) + (4.05 × 1) + (4.5 × 1) + (4.25 × 0) + (4.3 × 1) ⁄23
= (4.5 + 4.4 + 4.5 + 4.75 + 4.75 + 4.55 + 4.15 + 3.8 + 4.05 + 4.5 + 4.3) / 23
= (48.25) / 23
= 2.10

Now, the value of the calculated risk is compared with the threshold values, as specified. It can also be decided by the requirement engineers according to the security needs; the threshold value may vary according to the security requirements of

the software. Here, the value of the final risk is 2.10 which is at the high risk. This value is not tolerable at any cost. Hence, requirement engineers should revise the SRS by strengthening the 'Access Control and Rights' Policy.

6- CONCLUSION AND FUTURE WORK

The attributes of 'Access Control and Rights' policy are identified and a unique weight is hereby proposed for the implementation of the 'Access Control and Rights' policy. A risk assessment formula is also proposed for determining the risk related with this policy. The system will be stronger with respect to this policy implementation if it satisfies all or most of the attributes and will be on the low level of risk. A complete process of 'Access Control and Rights' policy is described for the security assurance of the SRS. Being prescriptive in nature, risk assessment is a concrete step towards implementing security *'right from the beginning'.*

Moreover, these proposals need to be validated in large samples for standardization. Therefore, future work may include the integrated level validation of the proposals along with the standardization for a large sample space. A software tool may also be developed for the automation of this complete process. In future, we are also trying to identify the attributes of other remaining security policies given in the section 1 based on the same pattern. This will help software developers and security experts for building secure software.

REFERENCES

[1] I. Peterson, "Fatal Defect: Chasing Killer Computer Bugs", Vintage Books, New York, pp. 210-216, 1996.
[2] Anup K. Ghosh, "Addressing New Security and Privacy Challenges, IT Pro pp. 10-11, May/June 2002.
[3] C. Cowan & Coleagues, "Stachgard: Automatic Adaptive Detection and Prevention of Buffer-Overflow attack", Proc. 7th usenix Security Symp., Usenix Assoc, San Diego, Calif, 1998.
[4] John Pescatore, "First Take FT-23-5794", Gartner Research, July 2004.
[5] Stephen Bell Wellington: "Poor requirements-definition equals ICT failure", Computer World, Thursday, 9 November, 2006.
[6] "Stop the seeds of project failure", BCS Project Management Article, www.bcs.org, September 2007.
[7] Nari Kannan, CEO and co-founder of Ajira "Agile Outsourcing: Requirements Gathering and Agile Methodologies" http://www.sourcingmag.com/content/ c061002a.asp
[8] An Innovative Approach to managing Software Requirement http://projectmanagement.knowledgestorm.com/shared/write/collateral/WTP/49705_52374_26971_MKS.pdf?ksi=1290251&ksc=1298777634
[9] Steve Lipner, Michael Howard, "The Trustworthy Computing Security Development Lifecycle", Microsoft Corporation, 2006.
[10] Information Security Policies & Procedures (Final v1.0), technical report of National Thermal Power Corporation Ltd., July 2006.
[11] http://www.comptechdoc.org/independent/security/recommendations/secpolgen.html
[12] Mustafa, K., Pandey, S. K., Rehman, S. (2008, September). Security assurance by efficient access control and rights. CSI Communication, 32(6), 29-33.