International Journal of Computer Science and Engineering (IJCSE)
Vol.1, Issue 1 Aug 2012 65-73

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

RISK ASSESSMENT THROUGH EFFICIENT AUTHENTICATION

[1]S. K. PANDEY & [2]K. MUSTAFA

[1]Department of Information Technology Board of Studies, The Institute of Chartered Accountants of, Noida- 201 309, India

[2]Department of Computer Science Jamia Millia Islamia (Central University), New Delhi-110 025, India

ABSTRACT

Deployed software, now-a-days, are continuously under attack. Attackers have been exploiting vulnerabilities for decades and seem to be increasing their attacks. Firewalls, intrusion detection and antivirus systems cannot simply solve this problem to the desirable extent. Only a concerted effort, by the software development community for building more secure software can foil attackers and allow users to feel protected from exploitation. It is observed that each phase of the SDLC should include the appropriate security assurance mechanism and countermeasures. From requirements through design and implementation to testing and deployment, security measures must be embedded throughout the SDLC phases. Authentication is one of the measure protection mechanisms, which is broadly accepted. Appropriate level of authentication may be well enforce security features and hence ensure security. In this paper, various attributes of 'Authentication' Policy are identified and then a weightage is assigned to each one, followed by the risk assessment to integrate steps for security assurance from the early in the development lifecycle. This will enable the assessment of appropriateness of authentication in terms of risk and lead to counter/additional measures for security assurance.

KEYWORDS: Authentication Policy, Authentication Attributes, Risk assessment for Authentication, Software Security, Security Assurance,

INTRODUCTION

Security is not only a desirable but now an essential feature of software so that it continues to function correctly under malicious attack. Most of the critical infrastructures all of us take for granted are fairly complex interconnected and interdependent systems. A single programming or design flaws in today's complex software system can disturb an 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 enterprise. Designing highly dependable security systems to ensure secure access to distributed software and information has been recorded as 'one urgent problem'. Software security is about designing software to be secure, making sure that software is 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 beginning'. The requirements phase is one of the foremost opportunities for the product team to consider how any feature including security 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 requirements 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 [13].

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:

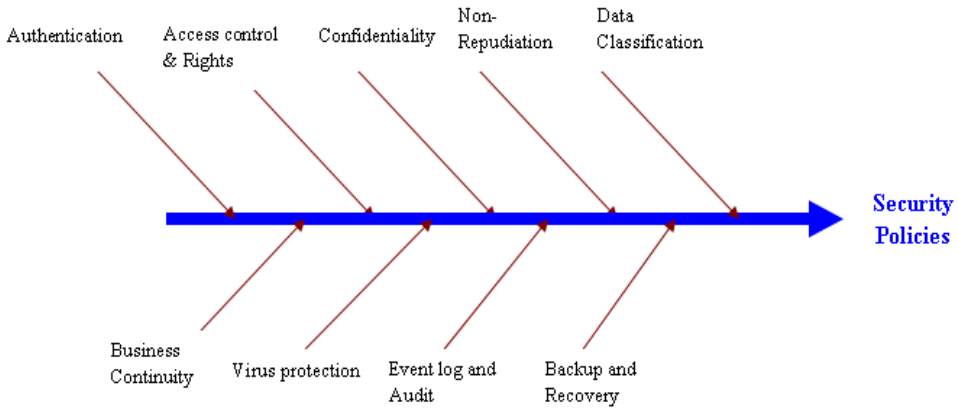

Figure 1: Security Policies

In this paper, we concentrate on 'Authentication' policy and risk assessment procedure. The purpose of this policy is to establish a standard for authentication 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 'Authentication' Policy. The attributes for 'Authentication' are discussed in Section 3, while a weightage 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.

AUTHENTICATION POLICY

Authentication permits the system to verify one's identification credentials. Authenticating yourself to a system tells it the information you have established to prove that you are who you say you are [12]. In order to prevent software from various business and environmental hazards, systems and procedures are being developed and implemented for authentication of users so that only authorized users given access to the application. Strong authentication process should be adapted for all critical applications and databases. It is the act of establishing or confirming something (or someone) as authentic, that is, that claims made by or about the thing are true [10]. Authenticating an object may mean confirming its provenance, whereas authenticating a person often consists of verifying their identity.

Authentication depends upon one or more authentication factors. In order to safeguard software, from various business and environmental threats, systems and procedures are developed and implemented for authentication of users so that only authorized users are given access to the application. The access controls can be well implemented through authentication, which should have approved solution. Strong authentication should be used for all critical applications & databases. Every organization has business data spread across multiple servers and location. These servers' process and data worth millions of rupees hence authentication of users has to be strictly controlled as per standard procedure. This policy should address Policies and Procedures related to the authentication of users to the organization's information resources. This policy should be applied to all the users and all the information resources including all operating systems, applications, databases, and all other computing resources [10].

ATTRIBUTES OF 'AUTHENTICATION' POLICY

Taking into account, the need and significance of an authentication 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 [11]. A pictorial representation of these attributes is depicted as follows:

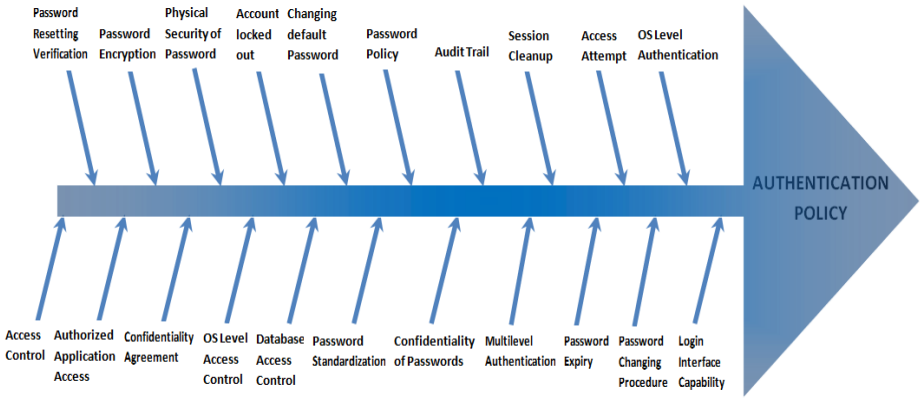

Figure 2: Attributes of Authentication Policy

WEIGHTAGE OF THE ATTRIBUTES

After proposing these attributes, it was felt by a team of experts that each attribute may have its unique weightage for the implementation of this security policy; that means the weightage of all the attributes may not be the same rather it will be different. We tried to explore the feasible ways to assign the weightages. Unfortunately, we could not find any such work in which such weightages have been assigned. Therefore, it was decided to take the help and guidance of experts' feedback on the relevant issues by designing a feedback form. The feedback was collected on the following issues:

- Checklists' relevance to the purpose;
- Analysis of the checklists' quality which include following heads:
  - Importance of the attribute;
  - Potential utility for evaluation practice;
  - Completeness/coverage of attributes; and
  - Relevance of all the attributes.
- In the rightmost column of each checklist, to assign a *weightage between 1 to 5 (1 is minimum and 5 is maximum)* using Likert's scale, to each attribute for the implementation of the each of the security policies.

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. It was completed by personal interactions with the experts by having 2-3 meetings in which the complete framework was discussed in detail. After a lengthy 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 weightage 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 weightage of each attribute was calculated. Based on the average value for each attribute, we finalized the weightage of the attributes, which is displayed in the following tables:

Table 1: Attributes' Weightage of Authentication Policy

S. No.	Attribute	Attribute's Weightage
1	Access Control	4.75
2	Authorized Application Access	4.4
3	Confidentiality Agreement	3.95
4	OS Level Access Control	3.9
5	Database Access Control	4.6
6	Password Standardization	4.35
7	Confidentiality of Passwords	4.4
8	Multilevel Authentication	4.1
9	Password Expiry	3.8
10	Password Changing Procedure	3.95
11	Login Interface Capability	4.15
12	Password Resetting Verification	3.95
13	Password Encryption	4.4
14	Physical Security of Password	4
15	Accounts locked out	3.95
16	Changing default Password	3.8
17	Password Policy	4.3
18	Audit Trail	4.35
19	Session Cleanup	4.15
20	Access Attempt	4.15
21	OS Level Authentication	3.85

RISK ASSESSMENT

After determining the weightage of the attributes of this security policy, we propose the process of risk assessment, which can be performed by using a mathematical formula. The formulation is done by using the concept of averages, which is a suitable statistical tool that may be used in these conditions. Here, risk assessment can be done by using the following formula:

Risk [Attributes]

$$\text{Risk} = \sum W_i X_i / n$$

where X_i = { 1 or 0

and i = 1, 2, 3, ……n

Here, W_i is the weightage of the attribute, and X_i is the value of the compliance of the attribute i.e. if a attribute is compliance, 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 risk 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.

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 basis of total compliance and non-compliance attributes. The results are given in the following table:

Table 2: Validation Data for Authentication Policy

S. No.	Attribute	Attribute's Weightage (W)	Compliance Status (X)	Weighted Compliance Factor (WCF)
1	Access Control	4.75	1	4.75
2	Authorized Application Access	4.4	1	4.4
3	Confidentiality Agreement	3.95	0	0

4	OS Level Access Control	3.9	0	0
5	Database Access Control	4.6	1	4.6
6	Password Standardization	4.35	1	4.35
7	Confidentiality of Passwords	4.4	1	4.4
8	Multilevel Authentication	4.1	0	0
9	Password Expiry	3.8	0	0
10	Password Changing Procedure	3.95	0	0
11	Login Interface Capability	4.15	1	4.15
12	Password Resetting Verification	3.95	1	3.95
13	Password Encryption	4.4	0	0
14	Physical Security of Password	4	0	0
15	Accounts locked out	3.95	1	3.95
16	Changing default Password	3.8	0	0
17	Password Policy	4.3	1	4.3
18	Audit Trail	4.35	0	0
19	Session Cleanup	4.15	0	0
20	Access Attempt	4.15	1	4.15
21	OS Level Authentication	3.85	1	3.85
				$\sum$ WCF = 46.85
				Risk = (46.85) / 21 = 2.23

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 2.23,

which is at the high risk. This value is not tolerable at any cost. Hence, requirement engineers should revise the SRS by strengthening the authentication attributes.

CONCLUSIONS AND FUTURE WORK

The attributes of Authentication policy are identified and a unique weightage is proposed for the implementation of the Authentication 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 Authentication 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 I, based on the same pattern. This will help software developers and security experts for building secure software.

REFERENCES

I. Peterson, "Fatal Defect: Chasing Killer Computer Bugs", Vintage Books, New York, 1996, pp. 210-216

1. Anup K. Ghosh, "Addressing New Security and Privacy Challenges, IT Pro pp. 10-11, May/June 2002.
2. C. Cowan & Coleagues, "Stachgard: Automatic Adaptive Detection and Prevention of Buffer-Overflow attack", Proc. 7th usenix Security Symp., Usenix Assoc, San Diego, Calif, 1998.
3. John Pescatore, "First Take FT-23-5794", Gartner Research, July 2004.
4. Stephen Bell Wellington: "Poor requirements-definition equals ICT failure", Computer World, Thursday, 9 November, 2006.
5. "Stop the seeds of project failure", BCS Project Management Article, www.bcs.org, September 2007.
6. Nari Kannan, CEO and co-founder of Ajira "Agile Outsourcing: Requirements Gathering and Agile Methodologies" http://www.sourcingmag.com/Content/c061002a.asp
7. An Innovative Approach to managing Software Requirement, http://projectmanagement.knowledes-torm.com/shared/write/collateral/WTP/49705_52374_26971_MKS.pdf?ksi= 290251&ksc 1298777634

8. Steve Lipner, Michael Howard, “The Trustworthy Computing Security Development Lifecycle”, Microsoft Corporation, 2006.

9. Information Security Policies & Procedures (Final v 1.0), Technical Report of National Thermal Power Corporation Ltd., July 2006.

10. S. K. Pandey & K. Mustafa: Security Assurance: An Authentication Initiative by Checklist, International Journal of Advanced Research in Computer Science, Vol. 01, No. 02, July-Aug 2010, pp. 110-113.

11. Mark Merkow & Jim Breithaupt, “Information Security: Principles and Practices”, Pearson Education, First Impression, 2007, pp. 234.

12. Nhlabatsi Armstrong, Security Requirements Engineering for evolving software systems: a survey, International Journal of Secure Software Engineering, Vol. 01, No. 01, Jan-March 2010, pp. 54-73.