ISSN No. 0976-5697

Volume 1, No. 2, July-August 2010

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Security Assurance: An Authentication Initiative by Checklist

S. K. Pandey*
Department of Information Technology
Board of Studies
The Institute of Chartered Accountants of India, Noida
E Mail: santo.panday@yahoo.co.in

K. Mustafa
Department of Computer Science
Jamia Millia Islamia (Central University)
New Delhi
E Mail: kmfarooki@yahoo.com

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. A checklist is proposed, in this paper, which can enable assessment of appropriateness of authentication and lead to counter/additional measures for security assurance.

Keywords-Software Security, Security Assurance, Authentication Policy, Authentication Checklist

I. INTRODUCTION

Software 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 opportunity 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.

Security policy means what could be securing for 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 following:

- Authentication Policy
- Access Rights and Control Policy
- Confidentiality of Data
- Data Classification Procedures
- Non-repudiation
- Business Continuity Policy
- Virus Protection
- Event Log and Audit Trails
- Backup & Recovery
- Incident Management, Intrusion Detection and Forensic Analysis

In this paper, we concentrate on authentication policy and its implementation procedure. The purpose of this policy is to establish a standard for authentication of users to the IT systems and creation of strong passwords, the protection of those passwords, and the frequency of change. A checklist is proposed for the verification, by structured walkthrough of SRS, of this policy.

The remainder of this paper is organized as follows. Section II describes the Authentication Policy. The Checklist Approach for authentication assessment is discussed in Section III, while a Checklist is proposed in Section IV. Implementation Mechanism is discussed in Section V. Conclusions and Future Works are given in Section VI.

II. AUTHENTICATION POLICY

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].

III. THE CHECKLIST APPROACH

There is no doubt that to ensure better process, atomic checklists are generally used and have been found to be handy and quite fruitful. Further, it becomes evident through the explanation of the researchers that a little work has been reported, hence it is viable to have a checklist for authentication process, which should be atomic in nature and can be easily usable for secure development process *'right from the beginning'*. Taking into account the need and significance of an authentication checklist for building secure software, an integrated and atomic checklist is hereby proposed. Items of the checklist have been derived from the reported and well-verified practices, as evident from the item-wise references.

IV. AN AUTHENTICATION CHECKLIST

The proposed checklist is divided in three main classes: Individual Authentication, Password, and System Related Issues. To have an assured authentication, every individual must be authenticated by various techniques. Strong password also plays a great role in authentication. A proper attention must be paid right from the creation of the password, till their management and backup. On the other hand, system related issues should also be addressed carefully. The access controls can be well implemented through authentication, which should have approved solution and may meet all or most of the following checklist items:

Class	Check point Description	Status (Y/N)
I. Individual Authentication	Is there unique identification or access code (user ID) for each user [11]?	
	Is there any procedure that restricts users to only those parts of application for which they have been properly authorized [12]?	
	Do users sign confidentiality agreement at the time of joining the organization [16]?	
II. Password	Does user IDs on the servers running software created only at the application level and not at the Operating System level [14]?	
	Is there any procedure for users to authenticate themselves for accessing the databases [14]?	
	Are password made as per some prescribed standards? /* At least six characters and combination of alphanumeric characters along with punctuation symbol.*/	
	Are user passwords remaining confidential and not shared, posted or otherwise divulged in any manner [15]?	
	Is there multilevel password authentication system for users [17]?	
	Is there any procedure for the expiry of passwords after a maximum period of 30 calendar days (or number of days defined by company) [13]?	
	Is there any procedure to force the users to change the passwords immediately after the first logon [13]? /* Systems Administrators should provide users with an initial password and configure the system according to the check point. */	
	Is there any capability in the software for the users to change their password on the login interface (after authentication) [17]?	
	Is there any procedure to reset the passwords by the security administrator on the request of the	

	users, only after verification of identity [16]?	
	Is there proper encrypted back up of all the passwords?	
	Are all the privileged user passwords sealed in an envelope and kept in a fire proof safe [14]? /* This is necessary in case the password is forgotten or the related person has left the organization without surrendering the passwords. */	
	Is there any facility that three successive failures must result in a user's account being locked out [18]? /* The users will not be able to login until the account is unlocked and the password reset. The user should submit a formal request to the Systems Engineer to carry out the exercise. */	
	Are defaults passwords shipped with software disabled or changed [10]? /* Vendor Supplied User-IDs/Passwords, encryption keys, and other access codes included with vendor-supplied systems should be promptly changed. */	
	Is there any policy implemented for reporting the password strength or weaknesses [11]? /* The minimum length of password is 12, is considered as a strong password. */	
III. System related issues	Is there any audit trails of successful and unsuccessful log-on attempts [13]?	
	Is there any facility for automatic system reboot or session cleanup following the disconnection of sessions [10]?	
	Is there any capability to limit the number of unsuccessful log-on access attempts [14]?	
	Is there any procedure for users to authenticate themselves to the operating systems for accessing the network resources like file server, print server, proxy server etc [14]?	

V. IMPLEMENTATION MECHANISM

In order to implement the authentication policy, there is a need to clearly identify the respondents for the implementation of this policy. Once respondents are identified, the second activity will be the distribution of checklist to the respondents with certain guidelines. Now respondents will verify the SRS through this checklist and prepare a document based on the implementation reports. Third activity is the collection of these documents from the individual respondent. Compilation of results obtained by these documents will be the fourth activity. Finally, analysis and reporting will be done by the project team. The authentication policy will be treated as strong if the document satisfies all or most of the checklist items. A flow diagram based on the above strategy, may be given as follows:

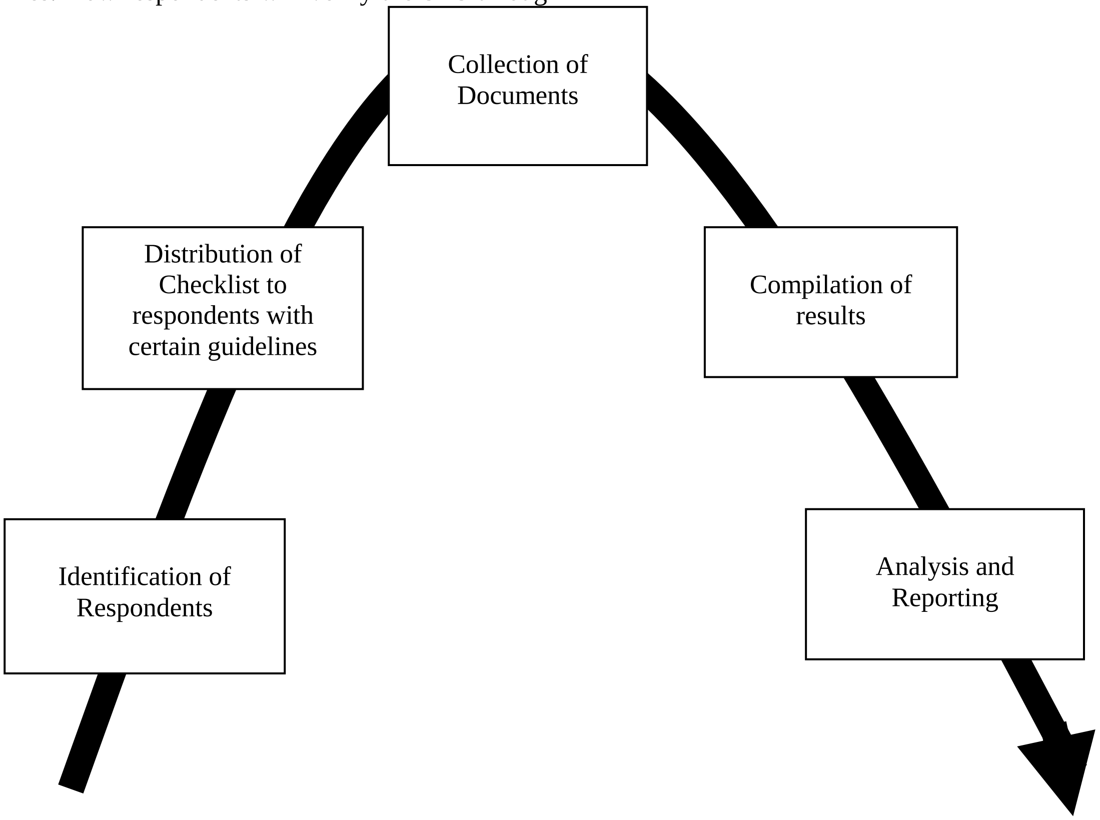
Collection of Documents
Distribution of Checklist to respondents with certain guidelines
Compilation of results
Identification of Respondents
Analysis and Reporting

Figure.1. Implementation Flow Diagram for the Checklist

VI. CONCLUSION AND FUTURE WORKS

A checklist is hereby proposed for the implementation of the authentication policy. The system will be stronger if it satisfies all or most of the checklist items given in the proposed checklist. A complete process of authentication policy is described for the security assurance of the SRS. Being prescriptive in nature, the checklist is highly implementable and it is a concrete step for the implementing security *'right from the beginning'*.

Future work may include the integrated level validation of the proposed checklist along with the standardization for a large sample space. In future, we are also trying to develop some more checklists for the implementation of the security policies for object oriented software, based on the same pattern. This will help software developers and security experts for building secure software.

VII. REFERENCES

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

[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 Requirementhttp://projectmanagement.knowledes-torm.com/shared/write/collateral/WTP/49705_52374_26971_MKS.pdf?ksi= 290251&ksc=1298777634

[9] Steve Lipner, Michael Howard, "The Trustworthy Computing Security Development Lifecycle", Microsoft Corporation, 2006.

[10] Information Security Policies & Procedures (Final v 1.0), technical report of National Thermal Power Corporation Ltd., July 2006.

[11] Marian Ventuneac, Tom Coffey, Ioan Salomie, "A policy-based security Framework for web-enabled applications", 2003, http://portal.acm.org/citation.cfm?

[12] Walter S. Kobus Jr., Identification and Authentication, 2000, http://www.tessllc.com/Physical %20Security %20Policy V4.pdf

[13] Walter S. Kobusjr, CISSP CISM, User Data Protection, 2000, http://www.tessllc.com/user%20data%20protection%20policyv4.pdf

[14] Access control (ISO 17799-27002) Privacy / Data Protection Project, www.privacy.med.miami.edu/glossary/xd_iso_access_control.htm

[15] John P. Quinn, Joseph A. Bailey, David E. Gaulin, "Law Firm Accounting and Financial Management", 2001.

[16] Erol Koç, Marcel Baur, and Germano Caronni: technical report of Sun Systems "PACISSO: P2P Access Control Incorporating Scalability and Self-Organization for Storage Systems SMLI TR- 2007-165, June 2007.

[17] Xunhua Wang, M. Hossain Heydari, Hua Lin, "An Intrusion-TolerantPassword Authentication System", in the proceedings of 19th Annual Computer Security Applications Conference (ACSAC' 03), pp. 110.

[18] Elaine Elvines: "Oxleas NHS Trust IM&T Security Policy Version: 3.2", April 2000.