IRM PRESS
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.irm-press.com
ITB9324

Chapter XI

Toward an Integrative Model of Application-Software Security

Vijay V. Raghavan
Northern Kentucky University, USA

ABSTRACT

Populist approaches to studying information systems security include architectural, infrastructure-related and system-level security. This study focuses on software security implemented and monitored during systems development and implementation stages. Moving away from the past checklist methods of studying software security, this study provides a model that could be used in categorizing checklists into meaningful clusters. Many constructs, such as principle of least privilege, execution monitoring, social engineering and formalism and pragmatism in security implementations, are identified in the model. The identification of useful constructs to study can form the basis of evaluating security in software systems as well as provide guidelines of implementing security in new systems developed.

INTRODUCTION

While academicians and industry practitioners have long recognized the need for securing information systems and computer architectures, there has recently been a heightened awareness of information technology (IT) management on computer-related security issues (Hulme, 2001). IT managers are increasingly worried about possible attacks on computer facilities and software, especially for mission critical software. There are indeed many dimensions to providing a secure computing environment for an organization, including computer viruses, Trojan horses, unauthorized accesses and intrusions and thefts to infrastructure. This complexity and multidimensional nature of establishing computer security require that the problem be tackled at many fronts simultaneously. Research in the area of information systems security has traditionally focused on architectural-, infrastructure- and systems-level security (Oppliger, 1997; Nelson, 1997). Emerging literature on application-level security, while providing useful paradigms, remains isolated and disparate (James, Joshi Walid, Aref & Spafford, 2001; Schneider, 2000; Bakersville 1993; Landerwehr, 1981). The current study focuses on a single, albeit an important, dimension of providing a safe and secure computing environment — application-software security.

THEORETICAL FOUNDATIONS

One of the difficulties in specifying data security requirements for an application is its complexity. Ting (1993) states that the characteristics of application-dependent security policies and requirements have not been clearly understood due to this complexity. Bellovin (2001) affirms that we cannot have "secure computer systems until we can build correct systems" and points out that "we don't know how to accomplish this, and probably never will." Schneider (2000) supports this view by highlighting the need for application-dependent special-purpose security policies. Current notions of architectural and infrastructure security do provide checklist items that could be transformed to an application-security context. A clear understanding as well as synthesizing current paradigms of computer security and transplanting relevant ideas to an application-development context is essential to move toward an integrative model of application-software security. Landwehr (1981) argues that formal models of computer security help designers decide exactly what "secure" means for their particular needs. In addition security regulations written in plain English tend to be "descriptive" rather than being "prescriptive" as in formal models.

Bakersville (1993), in his seminal exposition of application-development related security issues, finds the metaphor of "generations" useful in identifying the evolution of computer security paradigms. His exposition is akin to Nolan's stage hypothesis that is well known to the IS community. Bakersville identifies three major phases of evolution in computer security literature: the early checklist methods, second generation mechanistic-engineering methods and a third generation logical-transfor-