INTRUSION-AVOIDANCE VIA SYSTEM DIVERSITY

Anatoliy GORBENKO, Vyacheslav KHARCHENKO,
Olga TARASYUK, and Alexander ROMANOVSKY

Abstract: The paper discusses a generic intrusion-avoidance architecture allowing the system architects to decrease the risk of intrusions. The architecture employs software diversity at various system levels and dynamically reconfigures the deployment environment to avoid intrusions. This solution reduces the so-called system's days-of-risk which is a period of an increased security risk between the time when a vulnerability is publicly disclosed to the time when a patch is available to fix it. To select the less vulnerable system configuration we propose metrics estimating security risks by accounting a number of not-fixed vulnerabilities and their severity.

Keywords: Security, risk, vulnerability, diversity, intrusion avoidance.

Introduction

In this paper we focus on system security with regards to vulnerabilities of system components that can be exploited to attack the system and cause intrusions. A typical computer system consists of hardware, operating system (OS) and the set of system software components playing the role of a deployment environment for the specific applications. Thus, the dependability of a deployment environment significantly affects the dependability of the application services provided. Typical examples of system software components for web services are operating system (OS), web and application servers (AS and WS), and data base management systems (DBMS). Vulnerabilities of operating system and system software represent threats to dependability and, in particular, to security, that are additional to faults, errors and failures traditionally dealt with by the dependability community.[1] Intrusion tolerance is a general technique, which aims at tolerating system vulnerabilities that have been disclosed and can be exploited by an attacker. This is an active area of research and development with many useful solutions proposed.[2] However, less attention has been given to understanding how to make systems less vulnerable and to avoid intrusions while being configured and integrated out of COTS-components, such as OS, WS, AS, and DBMS. In the paper we propose a general system architecture aiming at decreasing the risk of intrusion and reducing number of *days-of-risk*.[3] This architecture employs diversity of the system software components and uses a dynamical reconfiguration strategy taking into account the security risks of different diverse system configurations. An implementation of the proposed architecture relies on the emerging cloud infrastructure services,[4] known as Infrastructure as a Service (IaaS). IaaS provides a platform virtualization environment and APIs that can enable such dynamic reconfiguration by switching between pre-built images of the diverse deployment environments.

Diversity of Deployment Environment

Design diversity is one of the most efficient methods of providing software fault-tolerance. In regard to multitier architecture of web-services, software diversity can be applied at the level of the operating system, web and application servers, data base management systems and, finally, for application software, both separately and in many various combinations. Platform-independent Java technologies provide the crucial support for applying diversity of different system components. Thanks to JVM, Java applications which meet and J2EE specification can be run on different operating systems under control of various web and applications servers. These components form a flexible deployment environment running the same application software and allowing to be dynamically reconfigured by replacing one component by another one of the same functionality (e.g. GlassFish AS can be replaced with Oracle WebLogic, or IBM WebSphere, etc.). At the same time, the .NET applications can employ only restricted diversity of the deployment environment limited to Microsoft Windows series of operating systems and different versions of Internet Information Server and MS SQL.

Security Risk Assessment

Security is a crucial property of modern computer systems. However, it can hardly be estimated in a probabilistic way similar to system availability or reliability. At the same time, it is indisputable that the level of system security depends on vulnerabilities existing in the system and system components. In this connection we propose weighted metrics estimating security risks of each system component (1) and a system in general (2) by accounting a number of ‘open’ (i.e. unpatched yet) vulnerabilities of each system component and by taking into consideration properties *VP* of an individual vulnerability, in particular, severity *S* and its popularity *P*.

$$VLC_i = \sum_{j=1}^{N_i} S_j \cdot P_j \, , \tag{1}$$

where VLC_i – vulnerability level (security risk) of the i-th system component; N_i – number of open (yet unpatched) vulnerabilities of the i-th component; S_j – severity of the j-th vulnerability; P_j – popularity of the j-th vulnerability.

$$VLS_k = \sum_{i=1}^{M_k} VLC_i \, , \tag{2}$$

where VLS_k – vulnerability level (security risk) of the k-th system configuration; M_k – number of system components (usually, each system configuration uses four basic system components: OS, WS, AS, DBMS); VLC_i – vulnerability level (security risk) of the i-th system component. The metrics proposed can be easily extended by taking into account potential harmful consequences, availability of exploit code, and other vulnerability properties. It is supposed that vulnerability properties have the same scale and are positive (lager value for higher security risk).

If it is not, all the negative vulnerability properties VP_i can be transformed to positive ones using $VP_i^{'} = VP_i^{\max} - VP_i$ and simple additive weighting technique can be applied to normalize them to the range [0…1]. System security risk should be re-estimated dynamically every time when a new vulnerability is discovered in any of system components or when a software patch fixing some of ‘open’ vulnerability is issued by a software vendor and applied by a system owner. Information about vulnerabilities of the different software system components, their amount and criticality can be retrieved by querying existing vulnerability databases (e.g. Common Vulnerabilities and Exposures, CVE and National Vulnerability Database, NVD), publicly available in the Internet. The information about patches and security advisories are released by the product owners.

Intrusion-Avoidance Architecture

The proposed intrusion avoidance approach is based on the idea of running at the different levels of the multi-tier system architecture (OS, WS, AS and DBMS) only those components having the least number of vulnerabilities. The rest diverse components should be hold in a stand-by mode. Every time when a new vulnerability is disclosed or some of existing vulnerabilities is fixed, the system security risks will be re-estimated and the most vulnerable system configuration will be replaced with the diverse one having the lowest security risk. The general intrusion-avoidance architecture is presented in Figure 1.

The architecture employs the IaaS (Infrastructure as a Service) cloud technology[5] providing crucial support for dynamic reconfiguration, storage and maintenance of the images of spare diverse system configurations. The core part of such architecture is a configuration controller. It retrieves information about emerging vulnerabilities of the different software system components and information about patches and security advisories released by the companies (product owners). By analysing such information the configuration controller estimates current security risks, selects the less vulnerable system configuration and activates it. Other functions performed by the controller are patch and settings management of the active and spare diverse configuration.

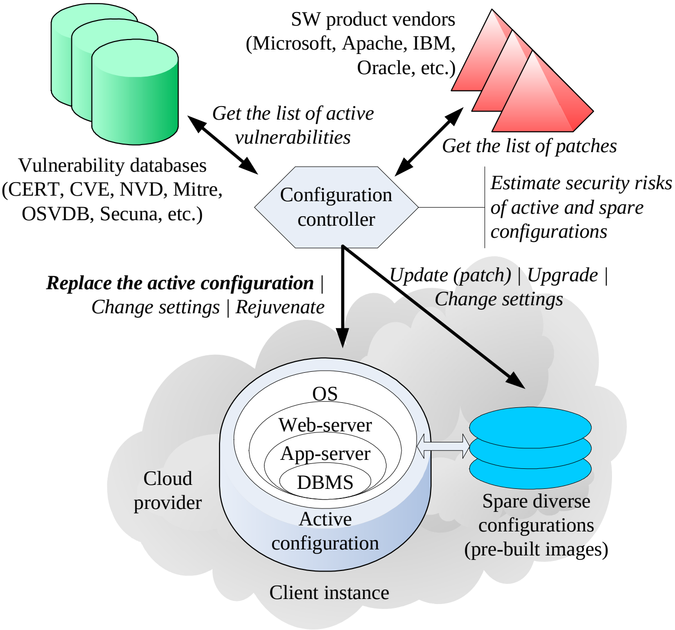

Figure 1. General architecture of the cloud-based intrusion-avoidance deployment environment.

Conclusion

The proposed intrusion-avoidance architecture that makes use of system component diversity can significantly improve the overall security of the computing environment used to deploy web services. Our work is in line with another recent study.[6] The approach proposed to intrusion avoidance is based on dynamical reconfiguration of the system by selecting and using the particular operating system, web and application servers and DBMS that have the minimal number of the residual (yet unpatched) vulnerabilities taking also into account their severity. Such strategy allows us to dynamically control (and to reduce) the number of residual vulnerabilities and their severity by the active and dynamic configuration of the deployment environment. This helps the architects to decrease the risks of malicious attacks and intrusions. The intrusion-avoidance architecture mainly relies on the cross-platform Java technologies and the IaaS cloud services providing the crucial support for diversity of the system components, their dynamic reconfiguration and maintenance of the spare configurations. The existing vulnerability databases like CVE and NVD provide the necessary up-to-date information for the security risk assessment, finding the least vulnerable configuration and reconfiguration decision making.

Notes:

[1] Algirdas Avizzienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing* 1:1 (2004): 11-33; Christian Cachin and Jonathan A. Poritz, "Secure Intrusion Tolerant Replication on the Internet," International Conference on Dependable Systems and Networks (2002), 167–176.

[2] Paulo Veríssimo, Nuno Ferreira Neves, and Miguel Correia, "The Middleware Architecture of MAFTIA: A Blueprint," *3rd IEEE Survivability Workshop*, 2000; Partha Pal, Paul Rubel, Michael Atighetchi, et al., "An Architecture for Adaptive Intrusion-Tolerant Applications," *Special issue of Software: Practice and Experience on Experiences with Auto-adaptive and Reconfigurable Systems* 36:11 (2006): 1331-54; Quyen Nguyen and A. Sood, "Realizing S-Reliability for Services via Recovery-driven Intrusion Tolerance Mechanism," Proceedings of the International Conference on *Dependable Systems and Networks Workshops*, Chicago, IL, 28 June – 1 July 2010, 176-181.

[3] Richard Ford, Herbert H. Thompson, and Fabien Casteran, "Role Comparison Report – Web Server Role" (Melbourne, FL: Security Innovation, March 2005), <www.microsoft.com/hk/windowsserver/compare/ReportsDetails.mspx?recid=31>.

[4] Rajkumar Buyya, James Broberg, and Andrzej Goscinski, eds., *Cloud Computing Principles and Paradigms* (Hoboken, NJ: Wiley, 2011).

[5] Ibid.

[6] Miguel Garcia, Alysson Bessani, Illir Gashi, Nuno Neves, and Rafael Obelheiro, "OS Diversity for Intrusion Tolerance: Myth or Reality?" Paper presented at the Performance and Dependability Symposium, the International Conference on Dependable Systems and Networks, 2011.

ANATOLIY GORBENKO received a PhD degree in Computer Science from National Aerospace University, Kharkiv, Ukraine in 2005. He is an Associate Professor at the Department of Computer Systems and Networks of the National Aerospace University in Kharkiv (Ukraine). His main research interests centre on assessment and ensuring dependability and fault tolerance in computer systems and service-oriented architectures.

VYACHESLAV KHARCHENKO – see the CV on p. 51 of this volume.

OLGA TARASYUK graduated in computer science in 2001 and received the PhD degree from the National Aerospace University, Kharkiv, Ukraine, in 2004. She is an Associate Professor at the Department of Computer Systems and Networks of the National Aerospace University in Kharkiv. Her research interests span many aspects of software development, verification, dependability assessment and prediction.

ALEXANDER ROMANOVSKY is a Professor in Computer Systems Research. His main research interests include system dependability, fault tolerance, software architectures, exception handling, error recovery, system structuring and verification of fault tolerance. Prof. Romanovsky is the Coordinator of the major FP7 Integrated Project on Industrial Deployment of System Engineering Methods Providing High Dependability and Productivity (DEPLOY, 2008-2012, www.deploy-project.eu).