

# Security Architecture of the Distributed System – Layered Approach

Miloš M. Cvetanović, Zaharije R. Radivojević

**Abstract**—Architecture of a security subsystem is a very important aspect of any computing infrastructure, so, it is not unusual practice to dedicate special attention to problem addressed in this paper. Modern attack techniques require new theoretical model for their elimination. Layered approach incorporated with metadata promises easy extensibility, flexibility and above all reliable security.

**Index Terms**—Architecture, Database systems, Distributed systems, Hierarchical structure, Layer, Layered approach, Metadata, Multidatabase, Security, Security architecture.

## I. INTRODUCTION

SECURITY is a very important aspect of any computing system, and has become a serious problem since companies and their infrastructure have become distributed in very heterogeneous fashion. As software on which infrastructure relies has become ever more complex, interdependent and interconnected, company's reputation has in turn become more vulnerable [6]. Companies that are faced with a choice between adding features and resolving security issues need to choose security. In any other case they will be faced with possibility of compromising sensitive data, loss of reputation and customer/collaborator confidence in other word faced with possibility of tremendous financial loss. So, it is necessary to build new systems incorporating security as integral part of their design. Instead, many companies and organizations try to avoid the definition of a security architecture and jump directly to ad-hoc testing against the security of their computing and networking infrastructures.

What we need is a security model that should be easy for developers to understand and build into their applications. Then again, it must be build upon the appropriate trust model [1], [7]. Layered approach is a design philosophy for protecting assets using layers of defensive protections. Each layer is extensible and therefore capable to respond on

increasing of asymmetric threat and attack tool sophistication. In order to meet rapidly changing business and technical requirements flexibility of this model must not be forgotten. But we have to be aware that ultimate goal of this approach is security. It will be accomplished through mutually reinforcing, complementary security controls and processes. So, what we have here is an architectural strategy implementing "defense in depth" motto.

Although much of what will be discussed here will be considered by some as intuitively obvious, it is these principles that are often forgotten and lead to misunderstandings. Even experienced researchers and practitioners should review this document.

## II. PRECONDITION

Before we get deeper into the consideration of mentioned problems, the precondition must be fulfilled. What we need is a switch in a way of thinking and distinction from traditional deep-rooted stereotypes. The most evident switch has to be made in comprehension of central to global networked environment. Central control has to be delegated to the widely dispersed nodes with limited visibility of the entire system. While projecting entire system one should not assume it as fixed bounded entity, on the contrary one should have in mind continuously evolving structure and therefore end points and perimeters should have relaxed boundary. This kind of structure requires that currently used fortified and insular approaches to security system design have to be abandoned, and replaced with one which will include more interdependency between nodes. There should be no absolutely trustable node only limited confidence is allowed. In such environment distinction between insiders and outsiders disappears. This will enable uniform defense mechanism whether the attack origins from inside or outside of the specific node.

Another important switch in the way of thinking includes importance of the fact that most events occur stochastically and asynchronously. This is opposite to the current architectures which base their security systems on the assumptions of predictability and patterns of the malicious attacks.

The next suggested change in thinking could cause polemics about it. It is about collective guilt and responsibility that should be shared. This is in a contrast with inherent practice of

Manuscript received May 9, 2003.

M. M. Cvetanović is with the Department of Computer Science and Computer Engineering, Faculty of Electrical and Electronic Engineering, University of Belgrade, Belgrade, 11000, Serbian and Montenegro (phone: +381-63-88-77-66-8; e-mail: cmilos@etf.bg.ac.yu).

Z. R. Radivojević is with the Department of Computer Science and Computer Engineering, Faculty of Electrical and Electronic Engineering, University of Belgrade, Belgrade, 11000, Serbian and Montenegro (phone: +381-11-3218-392; e-mail: zaki@etf.bg.ac.yu).



Figure 1. The internal security design of each node. This figure differs from the standard one because each level provides unique interface based on metadata. Dark levels represent metadata levels, which will be used as access points for probes.

single point of known responsibility which is held by theory of classical management. Principle of shared responsibility has its argument in the fact that it causes multiple responsibility of each participant and in the same time it will increase fault tolerance of the entire system. Then again, it will make management more difficult and complex. Security of the system should be considered as an essential part of the business infrastructure rather than just an overhead [2]. So, in any way it should not be a question which will be dealt only by technical experts, it should be accepted as a risk management perspective which will require involvement of the whole organization. It is not important to insure complete secure of each node, but to provide survival of mission. In other words survivability over security.

### III. OVERVIEW OF THE EXISTING ARCHITECTURES

Purpose of this section is to emphasize the lacks of the current available security systems. Majority of existing systems are burdened with the uniform protection of all the parts whether they are mission critical or not. This may cause overhead of the resource utilization in the cases when only important assets need protection. Another important disadvantage of legacy security systems is their basic paradigm that says “all or nothing”. But instead of that, the “show must go on” paradigm must be used. Also, static structure of the security systems implies the impossibility to accommodate to a new and sophisticated attack tools. Dynamic prediction of the malicious attacks should be incorporated as basic functionality, but with the existing architecture only post-attack analysis is possible.

Most of the present systems have some kind of hierarchical structure but not the structure in which multiple independent supervisors can act in order to protect the important resource.

Issues that can not be neglected are inability of the existing systems to adopt themselves to everyday needs in the

company’s workflow. That means that these systems are inappropriate for the modern concepts of total management of entire system. From all the issues mentioned above it could be concluded that new methodology in system architecture is an essential need.

### IV. LAYERED APPROACH

Glance on existing distributed systems reveals their variety of node dispersion across wide geographic regions. Nodes typically belong to the large-scale private networks and are not necessarily connected through Internet. What we are dealing with is a complex information system with the characteristics that are general and common for other infrastructure systems. In this kind of system, security must be considered at all stages of design, which not only satisfy their functional specifications but also satisfy security requirements.

The main idea is the decomposition of a system into hierarchical layers of abstraction, where the higher levels monitor lower levels by using some of their services and data produced in a regular operation mode. The lowest level (node level or functional level) consists of the interconnected nodes whose security architecture is based upon hierarchical multilevel design. The internal security design of each node is depicted on the Figure 1. This figure differs from the standard one because each level provides unique interface based on metadata [4]. This additional complexity has practical consequence that will allow upper levels (see Figure 2) to make probes at any level inside of the particular node. Upper layers (control layers) are designated for real-time monitoring and management of the lower layers. All layers operate independently and in parallel, but not necessarily on different machines. Although, executing on the same machines can be significant resource drain, running them exclusively on the separate nodes is beneficial for efficiency reasons. Because of the scale and the complexity of the presented system, each node in the upper layer is responsible for managing a set of the

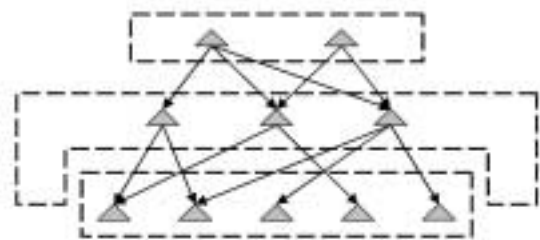


Figure 2. Security Architecture of the Distributed System – Layered Approach. Upper layers (control layers) are designated for real-time monitoring and management of the lower layers. The lowest layer is functional layer. Arrows are directed from observers toward subjects of monitoring. Note that nodes on different layers are not necessarily on separated machines.

nodes in the layer beneath it. These sets in lower layers are not disjunctive, in other words any node could be subordinated to the more than one node from the upper layer. This kind of structure implies that two basic principles of security are accomplished (diffusion and confusion). Mechanism by which upper layer could gather data from the node in the lower layer is by set of monitoring and actuating programs (probes) [3]. With the collected raw data, upper layer node can execute analysis programs. If analysis report any disparity from the normal results, such event could be handled generally in two ways. Locally, with some additional data requested from the node on the same or lower layer (not necessarily involved in the situation). Globally, by sending error signal to the upper layer nodes, and sending warning to the nodes on the same layer. As we have already mentioned metadata is used in intra-layer communication of each node. That provides probes with ability to accommodate and to work with heterogeneous nodes. In some cases probes are assigned to the control function, and it could lead to the collision with some other control probes. This could be solved with some kind of priority mechanism, which would preserve unique command chain of the architecture, but this is not topic of this paper.

The architecture, as described above, gives rise to a series of security concerns. These security concerns can be grouped into two loosely-defined categories. The first category is node protection, which includes both protecting of the node integrity and protecting the appended probes. Each node, as stated before, has its own multilevel internal structure (Figure 1), and is also defended by the one (or more) nodes from the upper layer. The second category is inter-node communication protection. This communication usually occurs over network channels, whose defending mechanisms are part of traditional security theory.

Described architecture poses one very important characteristic that is desirable not only for administrators but also for users and developers. It is transparency of whole security system which is achieved by multiple abstraction levels interconnected with metadata exchange [5]. Presented security model fits into a hierarchy of standard transparencies as shown in Figure 3. It is not always easy to delineate clearly the levels of transparency, but such a figure serves an important instructional purpose even if it is not fully correct [11].

Beside all that was already mentioned in the previous sections proposed architecture have three more implicit advantages. The first one arises from metadata usage, and it enables dynamic reconfiguration of the defense architecture (increased flexibility)[8], [9]. The second one is presented through system capability to survive node failure (increased fault tolerance). And, the third one presents the opportunity to use data mining techniques in order to advance prevention measures (increased attack prediction) [10].

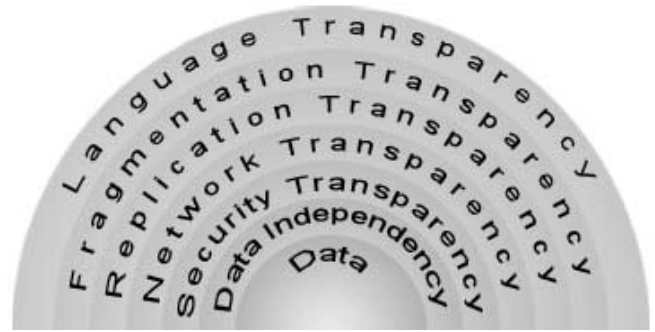


Figure 3. Layers of Transparency. Presented security model fits into a hierarchy of standard transparencies between Data Independence and Network Transparency. All layers which encompass security layer are not burden with security details.

## V. VERIFICATION

Real-life example that will be presented in this section verifies justification of the assumptions which were basis for the described architecture. System on which this theory was tested is distributed betting system (YuBet Project). The legacy system was built with the nodes that have great level of autonomy. What we got as a result is that the topology of the company's system can be described as a loosely-coupled distributed multidatabase system (with the central server at the company headquarter). Independent branches exchange data via semi-permanent connection.

Before the design process started the threats and attack scenarios had been considered. Three categories of intruders relevant to this subject were identified. Network intruders are the category that do not have direct access to the host and are trying to interpose between communicating nodes by sending false messages (which could imitate non existing tickets and bookmakers). The second category was malicious users. They had control over some node resources and programs so they had possibility for creating unreported tickets. And the third one, the most dangerous, is category of privileged users. They could change total history of dataflow and in such a way to manipulate with time, tickets and other important assets.

The proposed architecture, from the previous section, was fully implemented with three layers. Those three layers could be identified as a functional layer on the bottom and two control layers over it. The highest layer was deployed on the central server. Further discussion would be equivalent to the one from the previous section, and therefore it will be omitted here. There two implementation details that deserve to be mentioned because they solve the problems of probe integrity and communication protection. The first one solves the probe integrity problem, through a random set of one-way hash functions driven by a random seed. The second one resolves communication issues by utilization of self-destructing-self-creating public key cryptography.

## VI. CONCLUSION

In this paper we have presented a new methodology for architecture of the entire security system. It could be also used like an upgrade on the existing systems. Sustainability and robustness of this approach is appropriate for the requirements of the modern dynamic structured systems. Layered approach incorporated with metadata open numerous possibilities for self-modifying structure that would be adequate answer to the modern trends of asymmetric attacks. Most of these are yet to be explored, but some of the techniques like data mining could be already used for increasing real-time prevention measures.

## REFERENCES

- [1] D. Andert, R. Wakefield, and J. Weise, "Trust Modeling for Security Architecture Development," Sun Microsystems, Inc. 4150 Network Circle Santa Clara, CA 95045 U.S.A. 650 960-1300, December 2002. Available: <http://www.sun.com/blueprints>
- [2] *Concepts and Trends in Information Survivability*, Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213-3890, December 2001. Available: [http://www.cert.org/nav/index\\_purple.html](http://www.cert.org/nav/index_purple.html)
- [3] C. Wang, "A Security Architecture for Survivability Mechanisms," Ph.D. Thesis, Faculty of the School of Engineering and Applied Science, University of Virginia, Virginia, USA, October 2000.
- [4] G. M. Brunette, "Layered Security Architecture: Seatbelts and Airbags," Sun Microsystems, Inc. 4150 Network Circle Santa Clara, CA 95045 U.S.A. 650 960-1300, September 2002.
- [5] M. T. Ozsu, P. Valduriez, "Principles of Distributed Database Systems," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 07632, USA, 1991.
- [6] R. Oppliger, "What is a security architecture and why would you need one?," Computer Security Institute (CSI) Computer Security Alert, May 2001. Available: <http://www.esecurity.ch>
- [7] *Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria*, NCSC TECHNICAL REPORT – 005 Volume 1/5 Library No. S-243,039, National Security Agency, Fort George G. Meade, MD 20755-6000, USA, May 1996.
- [8] R. Hoque, "XML for Real Programmers," Morgan Kaufmann, 340 Pine Street, San Francisco, CA 94104-3205, USA, 2000.
- [9] E. Neuhold, I. Vujovic, V. Milutinovic, F. Patricelli, "Semantic Web," IOS PRESS, Nieuwe Hemweg 6B, 1013 BG Amsterdam, Netherlands, 2003.
- [10] J. Han, M. Kamber, "Data Mining: Concepts and Techniques," Morgan Kaufmann, 340 Pine Street, San Francisco, CA 94104-3205, USA, 2001.
- [11] V. Milutinovic, F. Patricelli, "Advances in E-Education on the Internet," IOS PRESS, Nieuwe Hemweg 6B, 1013 BG Amsterdam, Netherlands, 2003.