SECURITY AND PERFORMANCE VERIFICATION

MAKING IT WORK

Improve your verification of software security and performance by focusing on **eight major principles for success**.

WHAT'S IN THIS REPORT?

This report summarizes the results of a research on Security and Performance Verification conducted in partnership with the Experimental Software Engineering Group at COPPE/UFRJ in Brazil and the SINTEF DIGITAL in Norway.

The empirical study identified and characterize the working practices of Brazilian Organizations on security and performance verification. As a result, it was possible to highlight eight major factors influence the success of the security and performance verification practices.

Experimental Software Engineering
http://lens-ese.cos.ufrj.br

SINTEF
https://www.sintef.no

COPPE UFRJ
http://www.coppe.ufrj.br/

Introduction

The popularization and extensive use of software systems bring benefits to modern life. However, their full availability increases specific concerns regarding some critical software quality dimensions, including security and performance. Software development organizations usually include quality assurance activities throughout the software life-cycle to evaluate software quality, preventing failures after software releases. Software verification, including testing and reviewing, encompasses a set of actions to analyze the software looking for defects.

Security is relevant owing to critical and sensitive information manipulated and stored by the software systems while executing their functionalities. For instance, software systems are responsible for the manipulation of personal data, strategic information of organizations, and control of financial transactions. This information usually requires high confidence and different levels of classification, resulting in a growing interest in accessing it to obtain improper benefits.

Performance is relevant owing to the limitations of computational resources. Long response time can make users migrate to rival software systems, a delayed financial transaction can result in financial losses, and excessive power consumption can make the use of software systems unfeasible (if hosted on battery-based devices) or can increase the energy costs of those software systems running in large data centers.

Security and performance verification are activities that seek for defects regarding these specific quality perspectives. Individual or combined verification practices and techniques promote particular benefits and pose multiple challenges to the verification of security and performance.

Despite the existence of some security and performance verification techniques, software systems still present several defects related to these quality properties. Performance issues account for a significant fault category in specific domains (e.g., telecommunications) and news reporting software systems attacks are increasingly frequent. It may be due to (1) the inefficiency of security and performance verification practices, (2) the fact that software organizations do not adopt proper verification practices, or (3) the lack of evidence-based verification practices owing to an apparent disconnection between academy and industry in this context. Furthermore, automated attack scripts, the abundance of attack information, and global interconnection make it easier to attack software systems nowadays.

This report presents the results of an empirical study carried out in the context of Brazilian software organizations. The study aimed to understand how such organizations software performs security and performance verification activities. Thus, it identified the real and relevant issues faced by software organizations, allowing the software engineering researchers to answer them; providing useful feedback to the state of the practice.

THIS REPORT HIGHLIGHTS EIGHT ESSENTIAL FACTORS FOR SUCCESSFUL SECURITY AND PERFORMANCE VERIFICATION.

1. PROMOTE ORGANIZATIONAL AWARENESS OF THE IMPORTANCE OF SECURITY AND PERFORMANCE

\+ 2. BUILD A CROSS-FUNCTIONAL TEAM

\+ 3. MAKE SURE YOU HAVE CLEAR REQUIREMENTS

\+ 4. SELECT SUITABLE SUPPORT TOOLS

\+ 5. CONFIGURE AN ADEQUATE VERIFICATION ENVIRONMENT

\+ 6. USE A SYSTEMATIC VERIFICATION METHODOLOGY

\+ 7. PLAN SECURITY AND PERFORMANCE VERIFICATION ACTIVITIES

\+ 8. ENCOURAGE REUSE PRACTICES

=

Successful Security and Performance Verification

1 ORGANIZATIONAL AWARENESS

Security and performance verification should not be the responsibility of a separate organization department. The global organizational perception of the importance of the security and performance of software systems affects verification activities. Thus, security and performance verification activities require the support of every stakeholder.

The high-level manager should financially support security and performance activities. For instance, they should authorize the purchase of tools and including security and performance costs in the project budget planning.

Actions to improve

- ✓ Promoting training
- ✓ Informing the customer about the real state of software security and performance
- ✓ Keeping programmers well-informed about security and performance

The development team should consider security and performance verification a competitive advantage, understanding that if the verification team reports a failure, this is not against the project. Furthermore, the development team has in-depth knowledge of domain and software architecture. Thus, they can aid the decision-making regarding what should be verified, the prioritization of verification scenarios, and the identification of the verification scenarios dependencies.

The customer should understand that security and performance verification is not a waste of resources. They should be informed about the real status of security and performance of the system they are operating. Moreover, customers should understand that security and performance verification activities do not ensure a fully secure system or a system with no performance issues.

2 CROSS-FUNCTIONAL TEAM

Security and performance verification should not be performed in isolation by only one separate team. These activities require interaction between different roles as well as various skills.

Security and performance verification experts are responsible for providing knowledge related to security and performance, such as information security policies, security and performance development standards, digital certification, and cryptography. Furthermore, the security and performance verification team have the required technical knowledge to perform the fingerprinting step, identifying the technologies for developing the software (database, web server and programming language), and usage of security and performance tools.

Actions to improve

- ✓ Stimulating interaction between members of different teams
- ✓ Building a team having multiple skills
- ✓ Disseminating the view that verification team is not the enemy but allied

Infrastructure team supports the experts during security and performance verification activities. For instance, it may be necessary to allow a specific IP to access the server, make some changes in the kernel of the operating system, or restart the server after a catastrophic failure.

Database team is required in the security and performance verification activities because to repair or restore a database after a test battery.

Legislation expert supports the understanding of the legal impact of the lack o security and performance and assists in assessing legal risks.

3 CLEAR REQUIREMENTS

The requirements are the oracle for security and performance verification. Therefore, the lack of requirements prevents the team from a judge if the verification results are correct. Moreover, inaccurate requirements overload other groups (e.g., analysts, architects, and developers) because the verification team must continuously contact them. There are a set of issues brought by the lack of suitable requirements.

Actions to improve

- ✓ Involving the verification team in requirements phase
- ✓ Stimulating verification team to assess the testability of requirements
- ✓ Using techniques to handle security and performance requirements

The **lack of clear security and performance requirements**

- Forces the verification to be performed to identify the current state of the security and performance of the software instead of assessing whether the software meets its requirements but to evaluate the capacity of the system.
- Can be dangerous because the verification team determines the requirements by their own experience, which may not reflect customer expectations. Furthermore, the verification team could have some difficulties in deciding security and performance requirements.
- Cause **unnecessary stakeholders' interactions**, overloading other teams (e.g., analysts, architects, and developers) because the verification team must continuously contact them.

4 SUPPORT TOOLS

The use of suitable support tools is essential in security and performance verification activities because it can decrease the effort of manual activities. Free tools are advisable because the acquisition process is faster, as it involves the technical team only. In the case of adopting proprietary tools, it is necessary to ask the managers for permission, and the price may hinder or impede the buying process.

Actions to improve

- ✓ Allowing technical team to suggest and adopt support tools
- ✓ Supporting the use of free tools
- ✓ Using tools consistent with the verification team knowledge

It is crucial **to consider the team capability** when choosing a tool because the verification team may not have the necessary knowledge to use the advanced features of the tools.

Be careful about the **excessive number of false positives** produced by the tools. In this case, the results can be ignored because it takes substantial effort to analyze each of the reported failures.

The tools' report should not be directly delivered to the customer or the developers. These **report results should be analyzed and processed by the verification team**. Thus, a consistent report can be delivered to the target audience.

It is also essential to **make explicit in the reports the verification activities that did not reveal incidents**. It is psychologically positive for the client or developer to know that the system operates correctly in several respects.

5 VERIFICATION ENVIRONMENT

A suitable environment is essential for security and performance verification. It should encompass the configuration of the infrastructure responsible for system operation (e.g., application server and database parameters) and the configuration of the system itself (e.g., the data stored on the database while verification activities start).

The **insulation of the environment** should be a priority because other activities (e.g., users acceptance testing) may influence the verification results.

The use of **homologation environments is not recommended**. The performance tests may jeopardize the user acceptance activities because the simulation of a large number of users operating the system causes hardware overloaded.

The use of the **production environment is not suitable** because it is difficult to predict the behavior of the system's real users. Thus, the verification results could be misleading if the system is in operation.

The local network (or virtual private network) **influence the performance testing**. If the machine used for performance tests uses the default organization network, the requests and responses may be delayed owing to an overload of the network nodes (e.g., routers and sweets) that route them to the server on which the system is running.

Testing environment should be as much as possible similar to the production environment. In some cases, the hardware used in the production environment is more powerful than the equipment used in verification activities, and this may result in false results regarding system performance.

Actions to improve

- ✓ Using virtualization technologies to simulate execution environment
- ✓ Using virtualization technologies to set up tests agents
- ✓ Scheduling the verification activities if it is not possible instantiate a specific verification environment so that verification should never be performed in parallel with any other activity
- ✓ Keeping verification team well-informed about used technologies
- ✓ Performing each test cases more than one once and at different times to mitigate external influences

6 VERIFICATION METHODOLOGY

If an adequate methodology is available, the security and performance verification practices can be systematically performed.

Actions to improve

- ✓ Using a proposed methodology and adapting it to the context of the organization

The methodology should be **adaptable to the technologies** used in the development of the software system. For example, it is useless to perform web vulnerability analysis in embedded systems or database verification if the system does not store any data.

The methodology should **allow the increasing adoption of the proposed practices** so that the teams can have time to adapt to the new working practices.

The **methodology should allow evolution** because system security and performance should also evolve with time. Regarding security verification, evolution is mandatory, as new invasion techniques are continuously created.

The **asset identification and risk analysis are essential elements** of a sound methodology. Further, a methodology should make clear that the security and performance verification should occur after the correction of the defects identified by functional verification. Otherwise, the security and performance verification may identify functional failures, contrary to its real purpose

7 SECURITY AND PERFORMANCE VERIFICATION PLANNING

Planning is vital to define what should be done, to know the available resources (e.g., human resources, time, tools) and then establish the way the activities should be performed. However, usually, the security and performance verification is not well planned, leading to the need to reprioritize the verification activities, and consequently to the reduction of their coverage.

Actions to improve

- ✓ Prepare a plan to support the activities and anticipated risks, effort, and costs regarding the execution of security and verification activities

Managers have the **perception that security and performance verification activities require undue additional effort and cost**. Then they neglect these activities, excluding them from verification planning.

The stakeholders (e.g., managers and customers) have the **perception that verification activities can change the delivery time or the cost** of a software system. However, they do not consider the benefits of these activities.

8 REUSE

The reuse of knowledge and artifacts, such as functional test cases, brings a set of benefits to security and performance verification. Besides, it also brings benefits to functional verification activities.

Reuse the knowledge of previous systems and the functional test cases **increases the agility** and **reduce the cost** of the security and performance verification because it is possible to appropriate from past decisions (e.g., choose of techniques and criteria) already and the test cases should not be built from scratch.

The reuse of functional test cases **increases the verification coverage** and **improve the failure detection rate** as they represent real usage scenarios.

Reusing functional test cases as security or performance test cases also brings benefits for the functional verification itself: it **increases in the quality of functional testing** because the effort saved in non-functional testing can be used improving functional testing; it **increases in the diffusion of functional testing** owing to the increased importance of them in the development process.

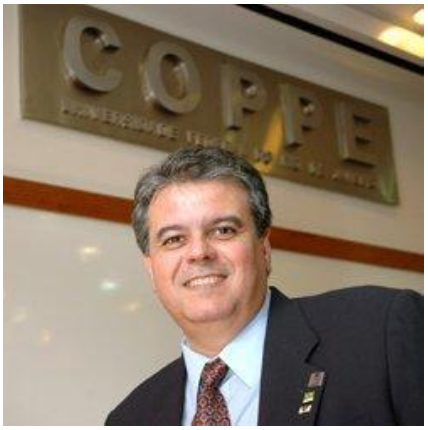
Actions to improve

- ✓ Reusing functional test cases as they represent real usage scenarios
- ✓ Reusing test cases from similar systems adapting parameters
- ✓ Reusing the knowledge acquired from other similar systems as a basis for the definition of the requirements
- ✓ Knowing common defects (e.g., vulnerabilities) and to use pre-defined test cases to identify the failures caused by theses defects

Authors

Victor Vidigal Ribeiro obtained a master degree, and he is a D.Sc. candidate in Systems Engineering and Computer Science at COPPE/Federal University of Rio de Janeiro. His research interest includes the application of experimental methods on the study of software systems verification (test and inspection), mainly focusing on non-functional requirements of contemporary systems. Further information available at http://www.cos.ufrj.br/~vidigal.

Daniela Soares Cruzes is a senior research scientist at SINTEF. Previously, she was an adjunct associate professor at the Norwegian University of Science and Technology (NTNU). She worked as a researcher fellow at the University of Maryland and Fraunhofer Center for Experimental Software Engineering-Maryland. Dr. Daniela Cruzes is the project manager of the SoS-Agile (Science of Security for Agile software Development) project funded by the Research Council of Norway. Her interests are agile and DevOps software development, software security, global software engineering, software testing processes, empirical research methods, theory development, and synthesis of software engineering studies.



Guilherme Horta Travassos received a D.Sc. degree in systems engineering and computer science from COPPE/UFRJ, with a post-doc in experimental software engineering at UMCP/USA. He is currently a full professor at COPPE/UFRJ and a CNPq (Brazilian Research Council) researcher. He is the Head of the Systems Engineering and Computer Science Program and leads the Experimental Software Engineering Group at COPPE/UFRJ. Member of the ISERN, SBC, and ACM. Apart from that, he takes part in the editorial board of Elsevier - IST (associate editor), World Scientific-IJSEKE, SBC - JSERD, and e-Informatica. Further information at http://www.cos.ufrj.br/~ght.