Methodologies and Tools for Software Vulnerabilities Identification

L. G. C. Barbato [1,2], N. L. VijayKumar [1] and A. Montes [2]

[1] Laboratory for Computing and Applied Mathematics - LAC
Brazilian National Institute for Space Research - INPE
C. Postal 515 – 12245-970 – São José dos Campos - SP
BRAZIL

[2] Information Systems Security Division - DSSI
Renato Archer Research Center – CENPRA
13069-901 – Campinas – SP
BRAZIL

E-mail: lgbarbato@lac.inpe.br, vijay@lac.inpe.br , antonio.montes@cenpra.gov.br

Keywords: secure programming, security tests, software security, software vulnerability

Nowadays, software are very important to help businesses of the companies and, in some cases they don't work without the software. But, in most of cases the software are presenting serious security problems. To help solving this problem, a methodology will be developed to test far software vulnerability starting with the threat analysis and finishing with the development of the final report. This methodology is composed for 6 main activities, such as:

1) Threats Analysis

In this step, threats are identified and analysed.

2) Risks Analysis

In this step, the risks are calculated to help the tests plan development and tests execution.

3) Tests Plan Development

In this step, the test process is planned.

4) Tests Execution

In this step, the tests are executed.

5) Results Analysis

In this step, the results are analyzed to identify software vulnerabilities.

6) Final Report Development

In this step, the final report is developed to present the tests results.

The biggest challenge of this work is develop a simple and easy methodology with great acceptance as well as a tool to implement and automatize all the process.

REFERENCES

Arkin, B.; Stender, S.; McGraw, G. Software Penetration Testing. IEEE Security & Privacy, 2005.

Barbato, L. G. C.; Duarte, L. O. Desenvolvimento Seguro de Software. III CTA - Seminário de Tecnologia, Informação e Conhecimento, 2006a.

——. Software Livre e Programação Segura: Verdades e Desafios. III Ciclo de Palestras sobre Software Livre, 2006b.

Barbato, L. G. C.; Duarte, L. O.; Grégio, A. R. A.; Montes, A. Aspectos Práticos da Codificação Segura. GTS 02.2005 - Grupo de Trabalho em Segurança, 2005a.

——. Codificação Segura: Abordagens Práticas. VII SSI - Simpósio de Segurançaa em Informática, 2005b.

Barbato, L. G. C.; Duarte, L. O.; Montes, A. Programação Segura: Uma Introdução à Auditoria de Códigos. GTS 02.2004 - Grupo de Trabalho em Segurança, 2004.

——. Vulnerabilidades de Software e Formas de Minimizar suas Explorações. GTS 01.2005 - Grupo de Trabalho em Segurança, 2005c.

Barbato, L. G. C.; Montes, A. Segurança de Software: Testes de Caixa Preta. GTS 02.2005 - Grupo de Trabalho em Segurança, 2005.

Blackburn, M.; Busser, R.; Nauman, A.; Chandramouli, R. Model-based Approach to Security Test Automation. Quality Week, 2002.

Cowan, C.; Barringer, M.; Beattie, S.; Kroah-Hartman, G.; Frantzen, M.; Lokier, J. FormatGuard: Atomatic Protection From printf Format String Vulnerabilities. In: Usenix Security Symposium, 10th., 2001, Washington, United States of America. Proceedings... 2001a.

Cowan, C.; Beattie, S.; Wright, C.; Kroah-Hartman, G. RaceGuard: Kernel Protection From Temporary File Race Vulnerabilities. In: USENIX Security Symposium, 10th., 2001, Washington DC, United States of America. Proceedings... 2001b. p. 165 – 172.

Cowan, C.; Pu, C.; Maier, D.; Hinton, H.; Walpole, J.; Bakke, P.; Beattie, S.; Grier, A.; Wagle, P.; Zhang, Q. StackGuard: Adaptive Detection and Prevention of Buffer-Overflow Attacks. In: USENIX Security Symposium, 7th., 1998, San Antonio, Texas, United States of America. Proceedings... 1998. p. 63–78.

Hoglund, G.; McGraw, G. Exploiting Software: How to Break Code, February 2004. ISBN 0-201-78695-8.

Howard, J. D.; Longstaff, T. A. A Common Language for Computer Security Incidents, October 1998.

Howard, M.; LeBlanc, D.; Viega, J. 19 deadly sins of software security, 2005. ISBN 0-07-2266085-8.

McGraw, G. Testing for Security During Development: Why we should scrap penetrate-and-patch. IEEE Aerospace and Electronic Systems Magazine, 1998.

The Common Criteria Project. ISO/IEC 15408, 1999.

Thompson, H. H.; Whittaker, J. A.; Mottay, F. E. Software Security Vulnerability Testing in Hostile Environments. ACM, 2002.