▼ Introduction to the CERT Software Application Security Minitrack

Robert C. Seacord
CERT/Software Engineering Institute
rcs@cert.org

Jason Rafail
CERT/Software Engineering Institute
jrafail@cert.org

Dan Plakosh
Software Engineering Institute
plakosh@sei.cmu.edu

Today's dependency on networked software systems has been matched by an increase in the number of attacks against governments, corporations, educational institutions, and individuals. These attacks result in the loss and compromise of sensitive data, system damage, lost productivity, and financial loss. To address this growing threat, the introduction of software vulnerabilities during development and ongoing maintenance must be significantly reduced, if not eliminated [1].

It is no secret that common, everyday software defects cause the majority of software vulnerabilities. A 2004 analysis of the National Vulnerability Database (NVD) showed that 64% of the vulnerabilities are due to programming errors and 51% of those due to classic errors like buffer overflows, cross-site-scripting, injection flaws [2].

Software vulnerabilities have financial consequences to both the users of the software and software vendors. A study based on real vulnerability announcements in 1999-2004 revealed: an average drop of the concerned vendor's stock price of 0.6% after each vulnerability announcement—not to mention the damage to the vendor's reputation [3].

The CERT Software application minitrack focuses on the research and automation techniques required to develop secure software systems that do not compromise other system properties such as performance or reliability. Current security engineering methods are demonstrably inadequate as software vulnerabilities are currently being discovered at the rate of over 4,000 per year as shown by Figure 1. These vulnerabilities are caused by software designs and implementations that do not adequately protect systems and by development practices that do not focus sufficiently on eliminating implementation defects that result in security flaws. An opportunity exists for systematic improvement that can lead to secure software applications and implementations.

Figure 1 Increasing vulnerabilities.

Software security, perhaps more any other area of computer science, could benefit from more interaction between researchers and software developers. A great deal of academic research fails to address the real world concerns of software developers, while the commercial software industry has failed to act on promising research results. The primary goal of this minitrack is to encourage an interchange of ideas between these communities.

References

[1] Seacord, Robert C. Secure Coding in C and C++. Boston, MA: Addison-Wesley, 2005.

[2] Heffley, J. Meunier, P. Can source code auditing software identify common vulnerabilities and be used to evaluate software security? Published in Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004.

[3] Telang, Rahul Wattal, Sunil. An Empirical Analysis of the Impact of Software Vulnerability Announcements on Firm Stock Price.. IEEE Transactions on Software Engineering, pp. 544-557. Aug. 2007.