I. Safonov, V. Safonov

SECURITY ENGINEERING AND PATCH MANAGEMENT FOR SAFETY OF INFORMATION TECHNOLOGIES

Automation of safety systems plays a crucial role among the prime factors influencing the provision of safety to high-risk objects [7]. If the "intellectual kernel" of research in the ***structure problems*** of safety is the Logical-Probabilistic approach [4], so it will also be correct for the Structured-Algorithmic approach [5, 6] to fit the ***behavior problems***. Everybody knew the most important Hippocratic advice to doctors: "First, do not harm". We suppose this advice is not less important to engineers: "First, do not damage". It is why security engineers should to trace how their design, development and management decisions impact on the safety of their technologies.

One of the most wanted technologies for contemporary informatization is a Patch Management. One of the most dangerous characteristics of a Patch Management is a contingency of its influence on data processing. Nevertheless a Patch Management must be considered as forced rather than desired approach for everybody connected to the Internet – the common hope and the common trouble. Eugene Kaspersky forecasts: "We risks to lose the Internet". He calls to the world professional community for development of the ***network protection technologies*** and ***information protection law***.

We agree with him but we must make the one addition.

The criminality is genetically caused ethics of small part of human society (mutants!) and it is almost independent from social conditions. The ***law*** is a set of artificially created rules convenient for interest ***intersection*** of basic social groups and strongly dependent from social system. But the ***ethics*** is a set of almost naturally created rules convenient for interest ***union*** of almost all social groups and therefore with a small dependence from social system. As the previous statement indicates, the criminal ethics (Remember Dale Carnegie?) more close to common ethics than criminal law to common (but country specialized) law. It is why we can not get by with only a ***law watchdogs***, we need also an ***ethics watchdogs***. And in view of bigger commonness of ethics comparatively with law, we can create the ***automated tools for common ethics support*** and give them more independence from their creators. Therefore, we propose to create the ***automated ethics watchdog for the Internet*** as a world community agent in the World Wide Web in the hope of preventing non normative ethical and supporting normative ethical behavior in the Internet.

Because we have not this tool by now and can not separate the sheep from the goats, let us return to our rams. The patch management must have a common methodological and theoretical basis. We know from the long-standing experience that the separation of concerns needed for the aspect-oriented (design for

Safety, design for Reliability, design for Security, etc.) hardware and software design and development may be effectively implemented in software engineering practice only on base of the Structured-Algorithmic approach. It is why we use ourselves and propose others the Structured-Algorithmic approach and its frameworks as the methodological basis for Security Engineering (as part of Trust Engineering) and Patch Management (as part of Risk Management).

Our research of Patch Management publications and our own experience in security and safety engineering give us ground and provide reason enough to recommend the ***insertion programming*** [3] as the theoretical basis for a Patch Management. The terms "insert" and "insertion" were borrowed from the genetic analysis where the ***insertional mutagenesis of genes*** was investigated traditionally. "In a complete plasmid clone, there are two types of DNA – the "vector" sequences and the "***insert***". The vector sequences are those regions ***necessary for useful cloning***. In contrast, however, the ***insert*** is the piece of DNA ***in which you are really interested***." [8].

Insertion programming is based on the theory of interaction of agents and environments. Every agent inserted into an environment is controlled by this environment but changes behavior the environment and, by doing so, its future reactions on insertion of other agents. Non-sanctioned insertion of malicious agents or sanctioned (but mistaken) insertion of agents changing a behavior of environment in undesirable way may be detected by the reaction an environment on special external agents testing retention and diagnosing violations of the environment behavior. And so, every intruder as well as every a patch violating behavior of the environment can be detected.

In more common case, insertion programming provides different elementary strategies for the environment tracing: 1) interactive trace generation, 2) search for the states having a given property, and 3) search for the traces having a given property. The elementary strategies may be used for creation of more sophisticated strategies. It is significant that insertion programs can be optimized on the algorithmic language which is more high-level than contemporary programming languages such as Java, C++ and C#. It allows us, for example, to reach the optimal trade-off between safety and security.

In conclusion, let us note that Security Engineering is one of directions of Trust Engineering [5], when the goal function is A) to maximize the security level in conditions that all requirements to other aspect levels are met, or B) to reach the extreme levels of aspects in condition that all requirements to the security level are met. The definition differentiates from commonly accepted definitions, which includes reliability problems in security problems, but we strongly believe that a security and reliability are very different by nature and have different causes (a security presumes malicious premeditation, but a reliability – unpremeditated mistakes, errors, defects, faults, malfunctions, or failures). But the reliability and safety problems do not occupy even five percents of his fundamental book. In fact, reliability problem and Reliability Engineering are at least not less important and not less complex than security problem and Security Engineering, but they have own theoretical basis and stores of techniques.

"How good is all this new security technology? Unfortunately, the honest answer is "nowhere near as good as it should be". New systems are often rapidly broken, and the same elementary mistakes are repeated in one application after another" [1]. These incompleteness and imperfection of Security Engineering pose a patch problem and Patch Management.

The first condition for successful decision of the Security Engineering problems, and between them the Patch Management problem, is the organization of research and development of all of these problems in context of problem of Business Process Engineering and Management Optimization, and simultaneously in three directions: 1) Business Process Engineering – Forecast-Driven Optimization, 2) Business Process Management – Event-Driven Optimization, and 3) Business Life Cycle Reengineering – Market-Driven Optimization.

The second condition is to interrupt a standardization process on the product level which is more useful for bad gays than for good gays, and start this process on the design and development levels.

The third condition is to create the theory and methodology of software engineering and reengineering oriented to manufacture the software self-protected from natural pathology and artificial influence by a purposeful design for functional correctness and evolution stability of the software. And we hope that the insertion programming may be the kernel of these attempts.

References

1. Anderson R. Security Engineering: A Guide to Building Dependable Distributed Systems. – New York, NY: John Wiley & Sons, 2001.

2. Letichevskii A.A., Kapitonova Yu.V., Volkov V.A., Vyshemirskii V.V., Insertion Programming. – Kibernetika i Sistemnyi Analiz, No. 1, 2003.

3. Ryabinin I.A. Reliability and Safety of Structural-Complex Systems. – Saint-Petersburg: Polytechnica, 2000.

4. Safonov I.V. Reliability Design for Management Algorithms. – Vladivostok: IAPU, 1982.

5. Safonov I. and Safonov V. Forecasting and Planning of Corporate Business Activity and Data Processing for Optimal Trust Engineering and Risk Management. – Proc. of Int. Sc. School "Modeling and Analysis of Safety and Risk in Complex Systems". – Saint-Petersburg: IPME, 2004.

6. Topolsky N.G. and Bludchy N.P. The Foundations of Integral Safety for High-Risk Objects. – Moscow: Fire-Safety Institute, 1998.

7. http://www.ebi.ac.uk/2can/glossary/index.php?letter=l.

8. http://www.gazeta.ru/techzone/2004/07/29_n_140735.shtml.