ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(36), DOI: 10.17485/ijst/2015/v8i36/90549, December 2015* ISSN (Online) : 0974-5645

Root Causes of Information Systems Vulnerabilities

A. V. Revnivykh and A. M. Fedotov

Novosibirsk State University of Economics and Management, Novosibirsk State University, Novosibirsk – 630090, Russian Federation; alexchr@mail.ru, fedotov@nsu.ru

Abstract

Background/Objectives: The article analyzes the information security from the standpoint of causes triggering the vulnerabilities of information technologies and systems. **Methods/Statistical analysis:** Information technologies are based on three interdependent components, namely, hardware, software and human resource (Figure 1). The susceptibility of final technologies to a number of threats challenging the information security takes roots in each of the abovementioned aspects both taken separately and in their complex combination. **Findings:** Data processing centers constitute a way to centralize the resources of the organizations' information infrastructure. Implementing such centers increases the system reliability and information availability in the whole and reduces the loading of the data transmission network at the organization. Meanwhile, data processing centers are an expensive option, and not every company can afford it. In addition to it, efficient foundation and functioning of such centers require highly qualified personnel. Modern information systems suffer from security imperfections. The main cause of their vulnerabilities roots in their complexity connected to the fact that information systems consist of a number of interrelated components which are designed and produced separately by different working teams. With the development of the civilization, the complexity increases steadily, therefore there is a burning need in working out measures. **Applications/Improvements:** May be used as guidance in order to improve the quality of testing the components of information systems and their compatibility.

Keywords: Human Factor, Information Systems, Security, Vulnerabilities, Vulnerability Risks

1. Introduction

Information technologies are based on three interdependent components, namely, hardware, software and human resource (Figure 1). The susceptibility of final technologies to a number of threats challenging the information security takes roots in each of the abovementioned aspects both taken separately and in their complex combination[1].

The main factor generating security imperfections in information technologiesis their complexity which tends to increase steadily with the progress of the civilization[2].

It is true that nowadaysone may state with confidence that in this world there is no person who would have a completeunderstanding of how asmartphone or a laptop functions[3]. This statement can be grounded by the fact that hardware and software are designed by different working teams. Thus, the products developed by these teams interact with definite interfaces; however, programmers have no possibility to thoroughly analyzethe working principles of a processor or a power unit. But this is just the tip of the iceberg! Electronic equipment functioning is conditioned by a specific infrastructure. For example, to provide an adequate work of a smartphone we need electricity (which, in its turn, is produced by an entireinfrastructure deserving a particular consideration), operable cellular network, etc.

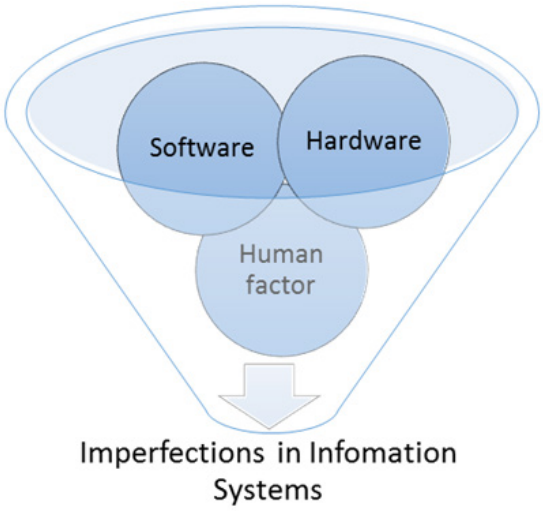

Figure 1. Imperfection of information systems.

* *Author for correspondence*

Therefore, it is the complexity of modern information systems (which tend to become more and more complicated) that will be the keynote of the chapters below.

2. Aspects of Information Security

The information security being a wide and multifaceted concept, its further studying requires defining itsthree specific aspects (criteria), namely, accessibility, relevance/integrityand confidentiality of information (Table 1).

Table 1. The main aspects of information security

Availability	Relevance/Integrity	Confidentiality
Availability of data Availability of services	Relevance and consistency of information Protection from unauthorized modification and deletion	Protection from unauthorized data reading

(i) Availability: Information availability means the guarantee of ready access to data and appropriate services at any time provided by the schedule of the system; the access to information should be divided into three stages:

- Possibility to send data request to the information system; it depends on the operabilityof the system interface involved in receiving the request, as well as on the efficiency and adequacy of communications bandwidth;
- Generating the response within the request-response time which doesn't exceed the timeout value; it depends on the systemefficiency and itsprocessing other requests or performing other operations;
- Possibility to deliver the response of the information system within the request-response time which doesn't exceed the timeout value; it depends on the operabilityof the system interface involved in sending responses to requests, as well as on the efficiency and adequacy of communications bandwidth.

Hence, the possibility toreceive data or a service requested depends on the efficiency and adequacy of communications bandwidthbetween the user and the interface of the information system, as well as on the efficiency and usage of theinformation system itself.

Technical causes of problems in the communication channel between the user and the system interface vary a lot, beginning with ordinary malfunctions and software failures up to successful denial-of-service attacks (PING-flooding, SYN-flooding, DDOS). Repulsing such denial-of-service attacks still remains difficult due to the specific features of IPv4, Internet program transport protocol common for local networksand theInternet.

Risks of the information system malfunction depend on the reliability of the combination of hardware and software components of the system together with the adequacy of the operator controlling their work. Availability problems arise from neglecting the standards on the stage of designing, producing or running the system.

It is also worth mentioning the risk of dimensions connected to the expansion and complication of the information networks. Very big networks generate phenomena quite difficult to unambiguously explain with an adequate specified reason. Apart from this, the denial of service may take roots in an ineffective information infrastructure, when the system architecture doesnot meet the requirements or requests any more.

(ii) Relevance/Integrity: Integrity should be considered as relevance and consistency of information together with its protection from destruction and unauthorized alteration or removal.

Risks of the information integrity violations are grounded by the following factors:

- Probability of a denial of service of the information system hardware and software, as data relevance and consistency may be disturbed as a result of failures;
- Degree of thinking out thealgorithms and reliability of authentication of the system users having the right to edit data stored there;
- Probability ofthe presence of non-documented possibilities in the software;
- Neglecting the standards on the stage of designing, producing or running the system;
- Imperfections in the organizational structure of the information system. For example, frequent reset of the whole system or its separate parts may causeboth additional costs and the integrity violations of the data stored and processed there.
- Human factor. For example, probability of social engineering towards people having access to editing data stored in the system. Insiders' threats.

(iii) Confidentiality: Confidentiality comprises the immunity of information to unauthorized readout.

Risks of the information confidentiality violations are grounded by the following factors:

- degree of thinking out the algorithms and reliability of authentication of the system users having the right to edit data stored there;
- probability of the presence of non-documented possibilities in the software;
- neglecting the standards on the stage of designing, producing or running the system;
- Imperfections in the organizational structure of the information system. For example, frequent reset of the whole system or its separate parts may cause both additional costs and also the integrity violations of the data stored and processed there;
- Human factor. For example, probability of social engineering towards people having access to editing data stored in the system. Insiders' threats[1].

3. Information System Components and their Influence on Information Security

As it was mentioned earlier, the security of final information systems is influenced, on the one hand, by the specificities of all their components taken separately, and, on the other hand, bythe way of combining these components into complexes. Let's analyse in detail all of the main components[4,5].

3.1 Causes of Hardware Imperfections

The equipment is fundamental for any information system. It is hardware resources that are used to launch system and applied programs.

Since 1950-s information systems have begun to develop intensively.It should be noted that information technologies and hardware in particular tended to advance in an accelerated way and still follow it, weighting coefficients of the reasonshaving been changed.

In 1950-1960-scomputing machines and data-transmission networks lying in the base of information technologies constituted a priority direction of research and production, mainly due to their implementation in the military-industrial complexes of the leading world powers. As modernized and destructive weapons represented the basis of the defensive capacity and influence coefficientof a big country, the technologies were implemented quite fast. Scientists and designers wereurgedby the military to do their best tocarry out intensive computations, to store and process steadily increasingdata arrays as fast as possible[6].

Since 1970-sthe weighting influence coefficientconcerning the hardware development trends began to shift to the commercial direction, as information technologies gained wider peaceful applications. In1980-scomputersappearnotonlyinbigc ompanies, butarealsousedat home, and a decade later portable electronic equipment begins to gain its pace. An enhanced competition in the market of electronics pushes designers and producers to minimizing the time required for the new units to come out in the market, since every hardware generation becomes out of date very quickly.

Thus, moral expiry of hardware takes place long before the physical one, when at the average first sights of malfunction can be detected. Hardware producers participating in the market competition have to constantly balance between the hardware reliability and its price for the final consumer. So, taking into consideration the shortening hardware lifespanconditioned by its fast moral expiry, producers are tempted to make their hardware less durable, saving materials, technologies etc.

It is worth noting that the task to minimize the time, while developing new generations of hardware, is set together with a constant and quite significant complication of technologies used in this hardware. One of the conditions required for modern hardware design and production is comprehensive hardware testing at various production stages. First samples of new hardware are subject to a detailed testing. The commercial hardware development time (including tests) being limited, it seems impossible to reveal all the defects in such a sophisticated equipment. The point is that a wider set of functions of modern hardware and various conditions for its applications don't allow checking all possible modesand situations.

The mankind does not have technologies to produce identical hardware components. Inanycase, two hardware units, even if they are produced in line, one just after another (their serial numbers being adjacent), will differ from each other. This is the reason why units being produced are subject to testing (selective or comprehensive).

The hardware reliability also depends on the production type. The production is usually divides into piece production, small-scale production and commercial production. Small-scale production is considered to be of the lowest quality, since the production control is not very thorough and the implementation of automatic testing lines is not justified from the economic standpoint.

3.2 Causes of Software Imperfections

Software can be divided into two main types: system and applied. The system software includes operating systems and drivers, while the applied software impliesall other programmes (Figure 2).

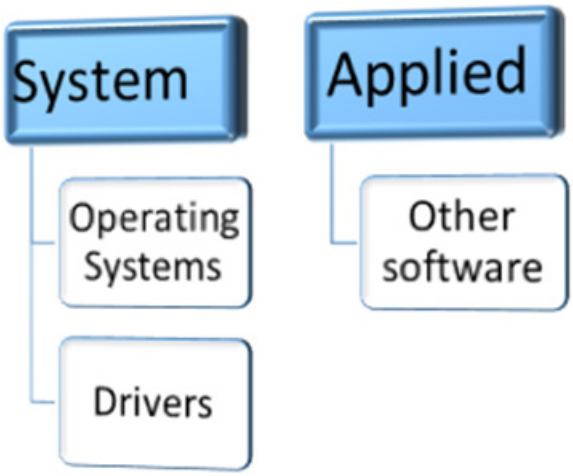

Figure 2. Types of Software.

In the case of software, as in the case of hardware, the issue of complexity and limited production time still remains very relevant.

Modern programmers, be they either system or applied ones, do not have a clear, detailed understanding of how their programs function, due to a number of reasons.

The software being multifunctional, it requires modular design by a working team, where each member only knows well the module he/she designed. Moreover, modules interact with special interfaces which are not ideal either.

The working tools of modern programmers are programming languages of a higher level, with their compiler and interpreter routines written, in their turn, in the programming languages developed earlier. An applied programmer designing his/her products in a language of a higher level cannot be aware of all the aspects concerning the algorithm of functioning of the program he/she created, as it is performed by the system only after numerous translations which finally transform into an algorithm written in the form of machine codes.

Besides, early generations of programming languages (and translators from these languages) had various features which are able to influence everything made in daughtertranslators. For example, we know the specific trait of the C language realization, which allows the interference into functioning of the program stack, when the data array transmitted into the stack and the memory automatically assigned to it do not correlate in size. A result of using programs with such features is a potential possibility to transfer control to a code area with an arbitrary addresswithin the address space of the process. This feature may be a precondition of the successful realization of attacks aiming at the buffer overflow and constituting one of the most dangerous modern technologies of unauthorized trespassing into protected systems[7-9].

A great influence on the software reliability is exerted by unpredictability of hardware and software system configuration, for which the software designers adopt their product. The composition, purpose and working conditions are unique for each of the information systems, therefore it is impossible to predict with confidence how the software product will behave in any various configurations, both hardware and software. This makes the software comprehensive testing rather problematic.

We should draw a particular attention to the software disseminated according to the open model. Some users are full of illusions about its quality, safety and protection from various tabs, assuming that it can be easily checked through source codes. However, nowadays this assumption is not valid, as source codes are enormous and can contain millions of code lines in different programming languages. Checking this data array and correlate all variants of the algorithm fork in the program is far from being realized in the manual mode[10,11].

To check software for tabs and vulnerabilities special heuristic software-hardware complexes can be used. However, it does not have sense to hope that there are no vulnerabilities if, as a result of the check, this complex did not reveal problems. The point is that it may only discover vulnerabilities programmed in it in advance, and it is mostly engaged in searching for well-known signatures[12-16].

Let's also point at the repeatability of operating systems used as host-platforms. There exists an insignificant

number of operating systems, thus, having learnt the features of the majority of them, the offender may define the type of the operating system used in the object under consideration and take advantage of its vulnerabilities which can be considered as non-documented possibilities.

3.3 Human Factor

The life cycle of any information system consists of a succession of several stages, beginning with the idea of creating an appropriate information system (as a rule, to ease an aspect of potential users' lives; with economic reasons, etc.), through the stages of its design, testing, usage and ending with its utilization. Please note that none of these stages can do without the human. When using information systems worked out by the human and for the human, one cannot avoid the risks connected to the violations of reliability and security[17].

The subject dealing with the types of temperament (choleric, sanguine, phlegmatic and melancholic), main channels of perceiving information (visual, audial, kinaesthetic and digital) and psychological types (compulsive, schizoid, hysteroid and depressive) cannotbeexhausted. Efficiency, attention and motivation of the human depend on a great number of various parameters. Meanwhile, modern information systems represent an inevitable result of work carried out by a lot of people, with the fruits assembled in one final product[1].

Human factor also involves well-known facts of developing various malicious software able to disturb the right hardware and software functioning. Some malicious software may appear as a result of programmers' mistakes; however, the most frequent cause of it is a malicious intent[18-20].

In the early 21 century the motivation of computer trespassers changed completely. Earlier, malicious software was developed mainly for kidding, hooliganism or showing off. Nowadays the main cause of external threats is a real opportunity to make profit. Earlier, writing a malicious code was amateur activity of a private nature, and now it is designed commercially. Apparently, modern creators of computer viruses don't waste their time any more, creating programs "for fun".

A malicious code can be used for a fraud (for example, blocking the user's interface of the operating system with a window with the coordinates and the sum to pay to the fraudster in order to unblock the interface), information terrorism (for example, sending letters containing the information about planned acts of terrorism), fishing (gaining access to the user's confidential databy means of a fraud) and even for sending "spam".

Algorithms of programming modules to the same information system are worked out by several programmers (sometimes, even by several programmers' teams), so it seems naïve to assume that these modules would ideally correlate with each other in the final product. This is especially appropriate to the situations when the system being designed was either not provided with standard documentation, or these standards were not respected.

In some information systems, various causes generate opportunities of unauthorized access to the system, passing by the resources of users' authentication. It often takes place due to the tabs made in earlier versions of the system in order to modernize it further. Afterwards, these tabs are not used, but they remain in the project code. Plus, sometimes program authors deliberately leave a "back door"for them in the system, and offenders learn about the way to take advantage of it.

Apart from this, the above mentioned imperfect correlation of various system modules can provide with the opportunities of unauthorized access as well.

Every day, users of modern technologies have to deal with a great number of manifoldinformation systems, facing the problem of correlating them. Meanwhile, there are no training courses where people would learn how to use all the information systems. And the last thing worrying the users is security: "If only it worked!".

Creators are making extra efforts, designing new versions of their information systems (or their parts), for example, every half a year. Once users got adapted to the system, its new version comes out and requires learning anew; in this version all the mistakes to which users got adapted are corrected and new ones are added users should put their efforts to get used to.

It is the human factor that makes the hardware and software components come out to the market in a hurry, without an appropriate testing. Every producer tries to determine the life cycle of his products and those of his rivals, separate in time presentations of his new products and those of his rivals. These fine marketing tricks do not improve the quality of the final product, designers and producers being under pressure of managers and marketing specialists

4. Conclusion

Nowadays, various technical and administrative measures are taken to minimize the risks of information security violations. Ingeneral, there is a trend to set up data processing centers and contract some functions of the organization's information systems for outsourcing.

In many cases, outsourcing to an organization which is professionally engaged in information systems support, seems quite attractive, as it allows solving the problems connected to the staff, purchasing some expensive equipment, work reliability. However, outsourcing has some disadvantages, such as an inevitable, to some extent, violations of confidentiality concerning data transmitted via the network and stored through the resources of the contracting organization. We should also mention the disputable nature of the long-term economic effectiveness of such infrastructure.

Data processing centers constitute a way to centralize the resources of the organizations' information infrastructure. Implementing such centers increases the system reliability and information availability in the whole and reduces the loading of the data transmission network at the organization.

Meanwhile, data processing centers are an expensive option, and not every company can afford it. In addition to it, efficient foundation and functioning of such centers require highly qualified personnel.

All above mentioned facts are to prove again that modern information systems suffer from security imperfections. The main cause of their vulnerabilities roots in their complexity connected to the fact that information systems consist of a number of interrelated components which are designed and produced separately by different working teams. With the development of the civilization, the complexity increases steadily, therefore there is a burning need in working out measures to improve the quality of testing the components of information systems and their compatibility.

5. References

1. Mazov NA, Revnivykh AV, Fedotov AM. Analysis of information security risks. Vestnik NGU. 2011; 9(2):80–9.
2. Brinkley DL, Schell RR. What is there to worry about? An Introduction to the Computer Security Problem. In: Information Security: An Integrated Collection of Essays. 1995. p. 11–39.
3. Revnivykh AV, Fedotov AM. Monitoring of information infrastructure of an organization. Vestnik NGU. 2013; 11(4):84–91.
4. Mukhanova AA, Revnivyh AV, Fedotov AM. Classification of threats and vulnerabilities of information security in corporate systems. Vestnik NSU. 2013.
5. Hogan CB. Protection imperfect: Security of some computing environments. ACM SIGOPS Operating Systems Rev. 1988; 22(3):7–27.
6. Department of defence trusted computer system evaluation criteria. Available from: http://csrc.nist.gov/publications/history/dod85.pdf
7. National Vulnerability Database. Available from: http://nvd.nist.gov/
8. MITRE Corp, Common Vulnerabilities and Exposures. Available from: http://www.cve.mitre.org/
9. Security focus. Available from: http://www.securityfocus.com
10. Witten B, Landwehr C, Caloyannides M. Does Open Source Improve System Security? IEEE Software. 2001; 18(5):57–61.
11. Lawton G. Open Source Security: Opportunity or Oxymoron? Computer. 2002 Mar; 35(3):18–21.
12. Wagner D, Foster JS, Brewer EA, Aiken A. A first step towards automated detection of buffer overun vulnerabilities. In: Network and Distributed System Security Symposium. San Diego: CA. 2000 Feb. p. 3–17.
13. Viega J, Bloch JT, Kohno Y, Mcgraw G. Its 4: a static vulnerability scanner for C and C++ code. In: Computer Security Applications. ACSAC '2000. 16Th Annual Conference. 2000. p. 257–67.
14. Ball T, Bounimova E, Cook B, Levin V, Lichtenberg J, Mcgarvey C, Ondrusek B, Rajamani SK, Ustuner A. Thorough static analysis of device drivers. SIGOPS Oper Syst Rev. 2006; 40(4): 73–85.
15. Evans D, Larochelle D. Improving security using extensible lightweight static analysis. IEEE Software. 2002; 19(1):42–51.
16. Xie Y, Chou A, Engler DR. Archer: Using symbolic, path-sensitive analysis to detect memory access errors. In: ESEC / SIGSOFT FSE. 2003. p. 327–36.
17. Islam S, Dong W. Human factors in software security risk management. Proceedings of the first international workshop on Leadership and Management in Software Architecture; Leipzig, Germany; ACM: 2008.
18. Aycock J. Computer Viruses and Malware. Germany: Springer; 2006.
19. Filiol E. Computer viruses: from theory to applications. Germany: Springer; 2005.
20. Kim K, Lee S, Yun Y, Choi J, Mun H. Security evaluation metric of windows-based Information security Products. Indian Journal of Science and Technology. 2015 Apr; 8(S8):54–62.