Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 12, Number 3 (2016), pp. 2133–2142

http://www.ripublication.com/gjpam.htm

Main Reasons of Information Systems Vulnerability

Alexander V. Revnivykh
Department of Information Security,
Novosibirsk State University of Economics and Management,
630099, Novosibirsk, Russian Federation.

Anatoliy M. Fedotov
Department of Information Technologies,
Novosibirsk State University,
630090, Novosibirsk, Russian Federation.

Abstract

In today's world, the role of information technology is difficult to overestimate. Due to their global spread, year by year humanity is increasingly relying on various information systems and, as a result, depends on them. But information systems cannot be ideal and the more complex they are, the more flaws and vulnerabilities of different kinds they have.

In this article, the reader is invited to look at information security from the perspective of the reasons of the vulnerability of information technologies and systems.

AMS subject classification:
Keywords: Security, Information Systems, Vulnerability, Vulnerability Risks.

1. Introduction

Information technology is based on three interrelated components: hardware, software, and human factor. The reasons of end technology exposure to a multitude of threats in terms of information security can be looked for in each of the above aspects alone, and in that they represent a complex set [1].

The main reason for the imperfection of information technology from the viewpoint of security is its complexity, which is also increasing continuously with each year of civilization's development [2].

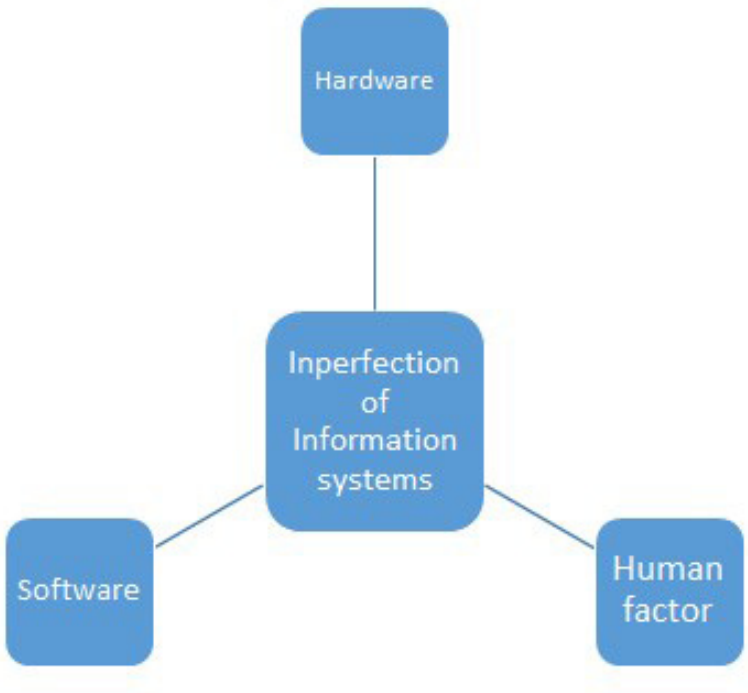

Figure 1: Imperfection of Information Systems.

In fact, at the moment we can say with absolute certainty that there is not a person in the world who would fully understand how a smartphone or laptop functions [3]. To substantiate this statement, it is sufficient to mention that hardware and software are created by different teams of people. Correspondingly, these teams' products interact with each other on certain interfaces, but programmers have no possibility to delve into the principle of a processor or power supply unit operation. But this is only the "tip of the iceberg"! Electronics works only when there is a certain infrastructure. For instance, a full operation of a smartphone requires electricity (which is produced, in its turn, by an entirely different infrastructure, which must be viewed separately), a functioning cellular network and others.

Thus, it is the complexity of modern information systems (which is constantly growing) that will be the central idea of the further narrative in this work.

2. Aspects of information security

Information security is a vast and versatile notion, so in order to further investigate it we must determine the aspects (criteria) which characterize information security. The three aspects are: availability, relevance/integrity and confidentiality of information.

Availability. By availability we understand the possibility of a subject to access data or

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

service on request at any time scheduled by the system operation. Access to information can be divided into several stages:

1. possibility for a subject to send a request for specific data to the information system (depends on the efficiency of the system interface, via which it receives such requests, and also on the serviceability and utilization of the communication channel between the subject and the server);

2. generation of a system response to a request over a time interval not exceeding the timeout (depends on the efficiency of the system, and also on its utilization processing other requests or other work);

3. possibility to deliver a response of the information system to the subject over a time interval not exceeding the timeout (depends on the efficiency of the system interface, via which it sends responses to requests, and also on the serviceability and utilization of the communication channel between the subject and the server).

Thus, availability of data or service on request depends on the efficiency and utilization of the communication channel between the user and the information system interface and on the efficiency and utilization of the information system itself.

Technical reasons for violations of the communication channel between the user and the system interface can be very different - from banal equipment failures and software faults to a successful implementation of denial-of-service attacks (PING-flooding, SYN-flooding, DDOS). At the same time, reflecting denial-of-service attacks is still difficult due to the peculiarities of the most common (in local networks and on the Internet) software network transport protocol IPv4.

The risk of malfunction of the information system comprising the information requested by the user depends on the reliability of sets of hardware and software components that comprise the system, and on the adequacy of the operator controlling their work. Availability violations arise bereason of non-compliance with standards in the system design, production or operation phase.

We should also note the risk associated with the size and complexity of information networks. In very large networks, there are phenomena that are difficult to explain clearly by any adequate specific reason. Moreover, denial of service may be due to an inefficient information infrastructure when the system architecture no longer complies with the requirements/demands.

Relevance/Integrity. By integrity we understand the relevance and consistency of information, its protection from destruction and unauthorized modification or deletion.

The risk of violating the integrity of information is provided by the following factors:

- Possibility of failure of hardware and software of the information system, as a violation of the relevance and consistency of the data can occur as a result of failures during their operation.
- Degree of reasonableness of algorithms and reliability of system authentication of users who have the right to edit the data stored in it.
- Possibility of having undocumented features in the software.
- Non-compliance with standards in the system design, production or operation phase.
- Imperfection of the organizational structure of the IS. For example, the need for frequent reconfiguration of the system or its parts may lead to violation of the confidentiality of stored and processed data in it, as well as additional costs.
- Human factor. For example, probability of social engineering in relation to persons who have access to editing the data stored in the system. Insider threats.

Confidentiality. By confidentiality we understand protection of information from unauthorized read access. The risk of violating the confidentiality of information is provided by the following factors:

- Degree of reasonableness of algorithms and reliability of system authentication of users who have the right to access the data stored in it.
- Possibility of having undocumented features in the software.
- Non-compliance with standards in the system design, production or operation phase.
- Imperfection of the organizational structure of the IS. For example, the need for frequent reconfiguration of the system or its parts may lead to violation of the confidentiality of stored and processed data in it, as well as additional costs.
- Human factor. For example, probability of social engineering in relation to persons who have access to the system. Insider threats [1].

3. Components of information systems and their influence on information security

As mentioned above, end information system security is influenced by both the features of each of its individual components and the way these components combine with each other in complex sets. Let's have a closer look at each of the principal components [4, 5].

3.1. Reasons of hardware imperfection

Hardware is the foundation of any information system. It is on hardware resources that system programs and applications run.

The rapid development of information systems hardware began with the 1950s and has been happening ever since. At the same time information technology (and, in particular, hardware) initially tended to accelerate its development, which is relevant at the moment, although the weighting coefficients of reasons for this have changed.

In the 1950s–1960s computers and data transmission networks, as the basis of information technology, were a priority for research and production mainly due to the need for their application for the military-industrial complex of the world powers. Since perfection and destructive power of weapons were the basis of the defense capability and the impact factor of a large state, it required the rapid development of technologies. Scientists and engineers were rushed by the military – it was necessary to as quickly as possible conduct large amounts of calculations, store and process the ever-increasing volumes of data.

Beginning with the 1970s a weighting coefficient of the impact factor on trends in the development of hardware began to shift in the commercial side, as information technologies were increasingly being used for peaceful purposes. In the 1980s, computers appear not only in large companies, but are also used in homes, and in the next decade portable electronics is beginning to spread. This ever-growing competition in the electronics market forces developers and manufacturers to hurry with the terms of launching new devices, as each generation of hardware devices becomes outdated very quickly.

A situation arises where commercial (moral) obsolescence of hardware occurs considerably earlier than physical, in which on average various kinds of faults begin to appear. In this case, device manufacturers, as participants of a competitive market, are constantly forced to choose a balance between the reliability of the device and its cost to the end user, so, given the declining period of operation of the devices due to their rapid obsolescence, manufacturers are tempted to make the device less durable to save on materials, technologies, etc.

Note that the task of reducing the development time of new generations of devices is set together with a constant and very significant complication of technologies that are used in these devices. One of the necessary conditions for the development and production of hardware in modern conditions has become a comprehensive test of devices, and at different stages of production.

The first samples of new devices are subjected to a detailed testing. However, given a time limit for the development (one of the stages of which is testing) of commercial products, to find all the flaws of sophisticated devices is impossible, as a broad functionality of modern hardware, as well as a variety of possible conditions for its application, do not give hope for the approbation of all possible modes and situations.

It should also be noted that mankind does not have technology to produce exactly the same, undistinguishable hardware components. In any case, the two devices, even manufactured on a conveyor one after another (and having serial numbers that differ by

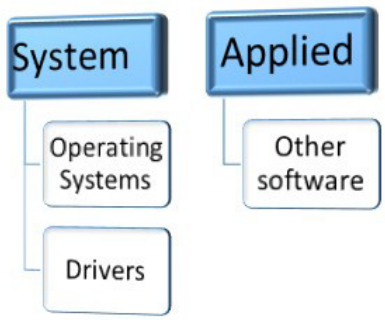

Figure 2: Types of Software.

one), will differ from each other. Therefore it is necessary to expose the products to testing (selective or exhaustive) during manufacturing.

Reliability of the hardware produced depends on the type of production. It is common to distinguish piece, small-scale and full-scale production. The least qualitative is considered small-scale production – control of each product is not particularly thorough, and introduction of automatic lines for product testing is not cost-effective.

3.2. Reasons of software imperfection

Software is divided into two main types – system and application. System software is operation systems and drivers, application software – all other programs.

The problem of complexity and development time limitations applies fully to software as well as hardware.

Modern programmers, both application and system, do not have a detailed understanding of the functioning of all the nuances of their program for a number of reasons.

The versatility of software entails its modular creation by a team of developers, each of whom has information only about his own developed module that communicates with other modules using special interfaces (also imperfect).

A working tool of modern programmers are mostly high-level programming languages, compilers and interpreters for which are written, in turn, using previously developed programming languages. Application programmers, who create their products in a high-level language, cannot know the nuances of the work algorithm of the created program as it is executed by the operation system only after numerous translations, which are ultimately converted into an algorithm, written in the form of machine codes.

In addition, it should be noted that the earlier generations programming languages (and translators from these languages) had different characteristics, which can affect everything created in child translators. For example, we know of a feature of “C” language implementation, allowing for interference in the program stack in case when sizes of the array transmitted to the stack and automatically selected for it in the memory do not match. The result of the programs that use these features – the potential for transfer of control to a code area with an arbitrary address within the address space of the process. This feature may be a prerequisite for the successful implementation of

buffer overflow attacks, which are one of the most dangerous modern technologies of unauthorized access to protected systems [7, 8, 9].

A serious impact on the reliability of software functioning has the unpredictability of hardware and software systems configuration, according to the operation of which developers have to calculate their software. Composition, purpose and operating conditions of each of the information systems are unique, so to predict exactly how the software product will behave in all possible configurations (both software and hardware) is impossible - they are too diverse. Accordingly, comprehensive testing of software is difficult.

A separate topic is software distributed in accordance with the open model. Some users have the illusion about its quality, safety and the absence in it of various "bookmarks" on the basis of the assumption that it is easy to check on source codes. But under current conditions it is not so – the source codes are huge in size and may represent millions of lines of code in several different programming languages. To check this data array and compare all the possible options for branching program algorithm in such cases manually is not possible [10, 11].

To check the software on bookmarks and vulnerabilities, there are special heuristic hardware and software complexes. However, it makes no sense to hope for the absence of vulnerabilities, if a test of such a complex did not reveal any problems – it can only detect pre-programmed vulnerabilities and is mainly engaged in the search for known signatures [12, 13, 14, 15, 16].

Note also the same type of operating systems used as host platforms. In the world there is not a very wide variety of operating systems, therefore having studied the features of most of them, an attacker can determine the type of operating system used in the object of interest and take advantage of its vulnerabilities, which can be regarded as undocumented possibilities.

3.3. Human factor

The life cycle of any information system consists of a sequence of several stages, beginning with the idea of establishing an appropriate information system (usually to simplify some aspects of life of potential users; for economic reasons, etc.), followed by the stages of its creation, testing, operation and final disposal. Pay attention to the fact that each of these stages involves human activity. Inevitable are risks of infringement of the reliability and security in the operation of information systems, conceived, created and operated by and for people [17].

The theme of the differences in human temperament types (choleric, sanguine, phlegmatic and melancholic), the main channels of information perception (visual, audial, kinesthetic and digital) and psychological types (compulsive, schizoid, hysteroid and depressive) is inexhaustible. People's performance, attention, motivation, depend on a huge number of various parameters. At the same time modern information systems are necessarily the result of the work of many people, the fruits of labor of whom have to be linked in a single end product [1].

Human factor also includes a widespread evidence of creating various types of malicious software that could potentially disrupt proper operation of both hardware and software components of information systems. Malicious software can appear as a result of programmers' errors, although the most common reason of its occurrence is malice [18, 19].

In the early 21st century, the motivation of computer hackers completely changed. Earlier malicious programs were developed mainly for fun, vandalism or to show their capabilities to others. Currently, the main reason of external threats - real possibility of financial gain. Earlier writing malicious code was rather amateurish and was of a special character, and now its commercial development is organized. Apparently, modern writers of computer viruses have started to value their time and do not create a program "just for fun".

There can be used a malicious code for fraud (for example, blocking the user interface of the operating system with a window, which contains the coordinates and the amount you need to transfer to the fraudster to unlock the interface), information terrorism (for example, sending letters with information about upcoming terrorist attacks), fishing (gaining access to the user's confidential data by deception) and even sending "spam".

Since the algorithms of software modules to the same information system are developed by several programmers (and even development teams), it would be naive to assume that these modules will be perfectly aligned with each other in the final product. This is particularly evident in cases where the design of the system was not originally supplied with the standard documentation, or it violated the requirements of the relevant standards.

Some information systems for various reasons allow possibilities of unauthorized entry into the system, bypassing the standard means of authenticating users. This is often due to the fact that during the development of early versions of the system bookmarks are made for its further improvement. Then, these bookmarks are not realized but remain in the project code. In addition, we can assume that sometimes developers specifically leave a "back door" for themselves about which attackers often find out.

In addition, a non-ideal combination of different modules of the system (mentioned above) can also provide similar possibilities of unauthorized connection.

Every day users of modern technologies have to deal with lots of different information systems. There is also a need to link different systems together. At the same time there are no training courses in which people would be taught to use all possible information systems at once. And security is the last thing that users care about: "As long as it works!".

Additionally, developers release newer versions of their information systems (or their parts), for example, every six months. As soon as users have learned to more or less confidently handle the system, when its new version comes out, which again should be dealt with, in which faults are fixed to which they are already used and adapted, and new ones are added to which they have yet to get used and adapt.

It is in connection with the human factor that products (both hardware and software components) are entering the market in a hurry, without exhaustive testing. Every man-

ufacturer is trying to calculate the life cycle of their products and competitor's products and select time for presentations of new products different from that of competitors âŁ“ such precise marketing tricks do not benefit the quality of the end product, developers and manufacturers who are constantly rushed by managers and marketers.

4. Conclusions

Currently, to reduce the risks of an abnormality in information security various technical and organizational measures are taken. In general, there is a tendency to create data processing centers, as well as the transfer of a number of functions of information systems of organizations to outsourcing.

In many cases, the transfer of a number of functions of the system to the organization, which is professionally engaged in the support of information systems, looks very attractive, as it solves the problems with the staff, purchase of expensive equipment, reliability of operation. Some of the drawbacks of outsourcing are the inevitability of some degree of violation of confidentiality of the data transmitted across the network and stored on the resources of the outsourcing services organization, as well as a controversial cost-effectiveness of such an infrastructure in the long term.

Data processing centers are a way to centralize the resources of information infrastructure of organizations. Introduction of such centers increases reliability of the overall system and availability of information, and as a rule reduces the load on the data transmission network of the organization.

At the same time, data processing centers are expensive and a relatively small number of organizations can afford it. Moreover, efficient establishment and operation of data processing centers clearly requires highly qualified personnel.

The above mentioned confirms the known fact that modern information systems are not perfect in terms of information security. The main reason for vulnerabilities of information systems is their complexity, related to the fact that information systems are composed of many interconnected components that are designed and manufactured separately by different groups of people. With each year of the development of civilization, the complexity is constantly increasing, so we need to develop measures to improve the quality of testing the components of information systems and their compatibility with each other.

References

[1] Mazov N.A., Revnivykh A.V., Fedotov A.M., Analysis of information security risks. *Vestnik NGU. Ser.: Information Technologies*. 2011; 9(2): 80–89.

[2] Brinkley D.L., Schell R.R., "What is there to worry about? An Introduction to the Computer Security Problem". Information Security: An Integrated Collection of Essays. 1995; 11–39.

[3] Revnivykh A.V. Fedotov A.M., Monitoring of information infrastructure of the

organizations. *Vestnik NGU. Ser.: Information Technologies*. 2014. ISSN 1818-7900. (in Russian)

[4] Mukhanova A.A., Revnivyh A.V., Fedotov A.M., Classification of threats and vulnerabilities of information security in corporate systems. *Vestnik NSU. Ser.: Information Technologies*. 2013. ISSN 1818-7900. (in Russian)

[5] Hogan C.B., "Protection Imperfect: The Security of Some Computing Environments". *ACM SIGOPS Operating Systems Rev*. 1988; 22(3), 7–27.

[6] Department of Defence Trusted Computer System Evaluation Criteria, DoD 5200.28-STD Supersedes CSC-STD-00l-83, dtd l5 Aug 83 Library No. S225,7ll (also known as "Orange Book").

[7] National Vulnerability Database. http://nvd.nist.gov/.

[8] MITRE Corp, Common Vulnerabilities and Exposures. http://www.cve.mitre.org/.

[9] Securityfocus. http://www.securityfocus.com.

[10] Witten B., Landwehr C., Caloyannides M., (2001, September/October). Does Open Source Improve System Security? *IEEE Software*, 57–61. Retrieved 5 May 2008, from Computer Database.

[11] Lawton G., (March 2002). Open Source Security: Opportunity or Oxymoron? Computer, 18–21. Retrieved 5 May 2008, from IEEE Computer Society Digital Library.

[12] Wagner David, Foster Jeffrey S., Brewer Eric A., Aiken Alexander., A first step towards automated detection of buffer overrun vulnerabilities. In: Network and Distributed System Security Symposium. San Diego, CA, February 2000; 3–17.

[13] Viega J., Bloch J.T., Kohno Y., Mcgraw G., Its4: a static vulnerability scanner for C and C++ code. In: Computer Security Applications. ACSAC '2000. 16Th Annual Conference. 2000; 257–267.

[14] Ball Thomas, Bounimova Ella, Cook Byron, Levin Vladimir, Lichtenberg Jakob, Mcgarvey Con, Ondrusek Bohus, Rajamani Sriram K., Ustuner Abdullah, Thorough static analysis of device drivers. *SIGOPS Oper. Syst. Rev.* 2006; 40(4): 73–85.

[15] Evans David, Larochelle David, Improving security using extensible lightweight static analysis. *IEEE Software*. 2002; 19(1): 42–51.

[16] Xie Yichen, Chou Andy, Engler Dawson R., Archer: using symbolic, path-sensitive analysis to detect memory access errors. In: ESEC / SIGSOFT FSE. 2003; 327–336.

[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 (LMSA2008)*, Leipzig, Germany, ACM, 2008.

[18] Aycock John, Computer Viruses and Malware. Springer. 2006. ISBN 978-0-387-30236-2.

[19] Filiol Eric, Computer viruses: from theory to applications. Springer. 2005. ISBN 978-2-287-23939-7.