Security Estimation of the Simulation Polygon for the Protection of Critical Information Resources

Bogdan Korniyenko[1], Liliya Galata[2], Lesya Ladieva[3]

[1] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
[2] Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
[3] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
bogdanko@i.ua, galataliliya@gmail.com, lrynus@yahoo.com

Abstract. In the article the question of Security estimation of information system protection through risk analysis is considered. An analysis of information risks is conducted for testing information security system, which allows to identify threats to information security. At present, different methods of analyzing information risks exist and are used, the main difference of which is in the scale of risk assessment: quantitative or qualitative. Based on analyzed existing methods of testing and assessing the vulnerabilities of the automated system, their advantages and disadvantages, for the possibility of further comparing the spent resources and information system security, a conclusion is made for the definition of an optimal method of testing the information security system method in the context of a constructed simulation polygon for the protection of critical information resources. The simulation polygon for the protection of critical information resources was developed and implemented based on the GNS3 application software. It is also concluded that the assessment of network security with mixed (complex) methods is not feasible. The optimal iRisk methodology for testing the information security system based on the simulation polygon for protection of critical information resources has been identified, among the considered methods for testing and analysis of automated system risks. The quantitative method iRisk is considered for Security estimation of information system protection. The general risk assessment iRisk is calculated considering the following parameters: Vulnerability Assessment, Threat Assessment, assessment of security tools. The methodology contains the general CVSS v3 vulnerability assessment system, which allows you to use constantly relevant coefficients to calculate vulnerabilities, and also have a list of all the major vulnerabilities that are associated with all modern software products that can be used in the automated system. The known vulnerabilities of used software and hardware are considered and the stability of the built simulation polygon for the protection of critical information resources to specific threats is calculated by iRisk method.

Keywords: Simulation Polygon, Critical Information Resources, Security, Vulnerability, Threat, Control.

1 Introduction

Periodic analysis of information risks is conducted for the research of information security system, it allows to identify threats to information security and in turn use and implement appropriate measures for their neutralization [1].

Based on the research and development of the simulation polygon for the protection of critical information resources by GNS3 application software, we can conclude that testing and evaluation of the constructed a secure network should be considered in the context of testing performance, impacting settings on the automated system security level, and in the context of used information protection tools [2]. This is due to the fact that in this case the emphasis is on the technical part, practically not considering organizational measures related to information security in the AS. Given that the emphasis is on hardware, software and network level of information protection, so network security evaluation by mixed (complex) methods is not appropriate.

Based on the fact that quantitative methods in conducting a risk analysis at software and technical protection level and if not consider organizational and technical component, are more effective, it should choose a quantitative evaluation method of protection [3, 4].

Among the main quantitative methods for analyzing information risks RiskWatch, Digital Security, ISRAM and iRisk, the iRisk method is more acceptable. The reason for this is, first of all, that this technique is free, informative enough, includes another CVSS v3 vulnerability assessment method, which is actively supported by the National Institute of Standards and Technology, and contains up-to-date information about the critical vulnerabilities of software and hardware, which in turn allows for an effective assessment of the level of network security.

The task that needs to be solved is to research of the simulation polygon for the protection of critical information resources by iRisk method for effectively assess the level of network security, considering the fact that the emphasis is on the hardware-software and network levels of information security.

2 iRisk Method

The iRisk method is formally one of the simplest estimates of information security quantitative risks for automated system. In general, it is calculated by the following equation:

$$iRisk = (Vulnerability \cdot Threat) - Controls \quad (1)$$

where *Vulnerability* - vulnerability assessment, *Threat* - threat assessment, *Control* - assessment of security tools. This technique uses a different Common Vulnerability Scoring System v3.0 (CVSS V3) methodology for vulnerability assessment.

When assessing the threat, the probability of realization of the threat and the degree of its influence are being assessed. The degree of impact of the threat is estimated through the indicators of losses. To assess the probability of implementing a specific threat, there are two indicators: ARO is the expected number of threats during the year, and the level of knowledge and the offender's access level in the AS.

Formally, the calculation is not a complicated equation, but this methodology contains a general CVSS vulnerability assessment system, which is supported by market leaders in the field of information security in practice, that allows you to use constantly relevant coefficients for calculating vulnerabilities, and also have a list of all the major vulnerabilities associated with all modern software products that can be used in an automated system [5].

2.1 Vulnerability

First of all, we have calculated Vulnerability, by using the standard CVSS v3. The calculation takes place according to the scheme presented in Fig. 1. During the calculation, a large number of coefficients are used, so for convenience we will use the software of the National Institute of Standards and Technologies, then correct parameters setting will allow to get the result of calculations in the form of a scale from 1 to 10, where 1 it's a low level (no vulnerability), and the value 10 it's the critical vulnerability that needs to be eliminated. The standard includes three groups of metrics required for calculation: base, temporal and environmental.

Fig. 1. General scheme of vulnerability calculation in CVSS v3

The value of the metric is accepted as a pair of vector (specific values of individual indicators) and a numerical value, which is calculated basing on all indicators and using the equation defined in the standard. Fig. 2 shows all the necessary parameters for calculating the environmental metric of the polygon for the protection of critical information resources.

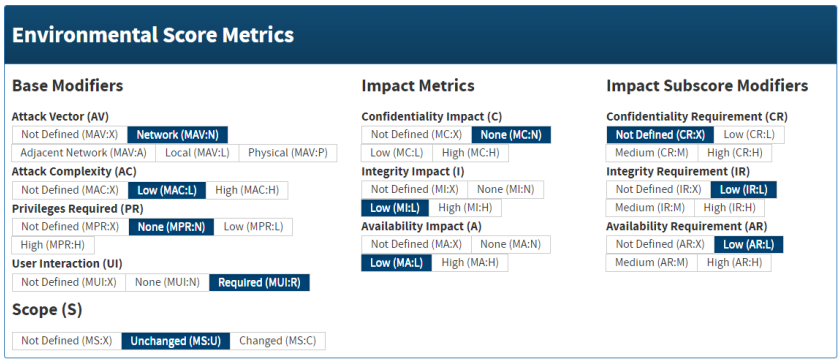

Fig. 2. The environmental metric of the polygon for the protection of critical information resources

2.2 Threat Assessment

According to this standard, the threat is explained as a negative event that may result of the vulnerability benefits. In order to make the equation as simple as possible, the iRisk method focuses on two main components: *impact* and *likelihood*. Fig. 3 is presented the scheme of threats estimation in iRisk method.

Fig. 3. Scheme for threats estimating in the iRisk method

Impact is the amount of damage that this incident will bring to the organization. Within the iRisk SecureState equation, today the following criteria are used to determine the impact. By default, the following values are assigned, but they can be changed according to the needs of the evaluated object:

- financial (25) - whether threats destroy the organization financial flows;
- strategic (15) – whether threats lead to long-term strategic losses;
- operational (25) – whether threats influence on the work continuity;

- law compliance (25) - whether threats affect the ability to keep to the standards;
- reputation (10) - whether threats affect the relationship with customers.

Likelihood is another major component of the threat. The iRisk method uses two factors to estimate the probability: the annual expected number of threat implementations and the attacker's level of knowledge and access (correlation table between the level of knowledge/access and the annual number of threat implementations ARO (annualized rate of occurrence) [6]).

The threat is calculated by the Eq. (2), where Likelihood (correlation from table ARO [6]). If the threat is on a scale from 100 to 50 - the level of risk is high, from 50 to 10 – medium**,** from 1 to 10 - low.

$$Threat = Impact \cdot LikeLihood \tag{2}$$

2.3 Control (Assessment of Security Tools)

Based on the definition of the ISACA organization, preventive, detection, correction or deterrence means for security may be used in iRisk. The structure of the Control parameter (assessment of security tools) is presented on Fig. 4.

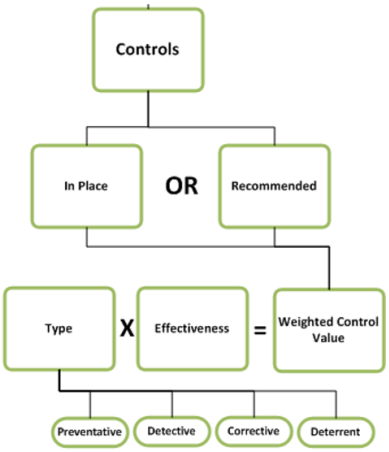

Fig. 4. Structure of the Control parameter of the iRisk method

According to the standard, the tools have the following ratings: preventive - 5, detection - 4, correction - 3, deterrence -3.

The next step is to define the *Controls* (efficiency), it has a five-point scale by the standard: 5 - if the information security tools in the network significantly exceed the goal, 4 - exceed the goal, 3 - the implementation corresponds to the goal, 2 – the implementation is not fully satisfying its goal, 1 - slightly up to its goal.

Adding indicators by CVSS we will get the following values:

- optimized (801 - 1000) - the tool can't be developed or implemented better;
- managed (601 - 800) - the tool continues to improve;

- defined (401 - 600) - the security tools are clearly defined and reduce the risk to medium;
- initial / Ad-Hoc (1 - 200) – the tool provides only some protection value.

Thus, the three main components, which appears in the method iRisk, balance each other. The highest possible score for the threat is 100, which is multiplied by the maximum vulnerability (10). That is 1000 points potential, which is compensated by the potentially perfectly implemented protection, at the end will leave zero risk. In practice, this is almost not achievable and, in any case, left a part of the residual risk. That is, the risk varies in values from 0 to 1000, in this case the smaller value means the more secure automated system.

3 Software and Hardware Vulnerabilities

The designed simulation cybersecurity polygon hasn't so many vulnerabilities due to the high-quality equipment, the access control that divides the network into the demilitarized zone, the internal and external network, and the network settings, that limit access to the network from the outside, limit number of half-connections, which reduces the effectiveness of DdoS attacks, network scan, etc. [2]. And still, the vulnerabilities remain on the software and hardware level. Next, we will look at some of them, the calculation of the security of the polygon for the protection of critical information resources will be done using iRisk [7-10].

3.1 Cisco IOS Arbitrary Command Execution Vulnerability (CVE-2012-0384)

The vulnerability occurs due to error in HTTP/HTTPS authorization that allows an authenticated user to execute any Cisco IOS software commands configured for user privilege levels.

We will calculate the base metric for Vulnerability calculation, and for more correctness, according to the security of the cybersecurity polygon we will calculate the temporal and environmental metric, as described above [11-14].

Base Score Metrics {Attack Complexity = Low; Privileges Required = Low; User Interaction = None; Scope= Unchanged; Confidentiality Impact = High; Integrity Impact = High; Availability Impact = High}

Temporal Score Metrics Score Metrics {Exploitability = Functional exploit exist}

Environmental Score Metrics {Base Modifiers {Attack Vector = Local; Attack Complexity = Low; Privileges Required = Low; User Interaction = None} {Scope = Unchanged} {Impact Metrics {Confidentiality Impact = Low; Integrity Impact = Low; Availability Impact = High}} {Impact Subscore Modifiers {Confidentiality Requirement = Low; Integrity Requirement = Low; Availability Requirement = Low}}}

The resulting calculation of the base level Vulnerability assessment equal 7.8 out of 10, which is shown on Fig. 5.

Considering that the threat should be realized from inside and first of all is oriented to a normal user without administrator rights and the expected number of threats is estimated as high, then from the ARO table [6] we choose the correlation value *Impact* = 0.9. So, according to the Eq. (2): $Threat = 0.9 \cdot 100 = 90$.

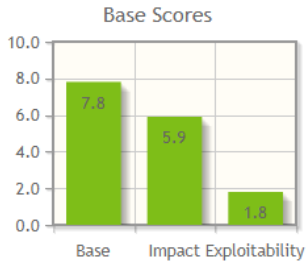

Fig. 5. The Base CVE-2012-0384 vulnerability metric for the cybersecurity polygon

As described above, the value *Controls* is estimated at *650*, which will mean - the tool continues to improve.

That is, the value *iRisk = (7.8 · 90) - 650 = 50* for Cisco IOS Arbitrary Command Execution Vulnerability (CVE-2012-0384).

3.2 Cisco Access Control Bypass Vulnerability (CVE-2012-1342)

The vulnerability of Cisco routers allows remote attacks to bypass the Access Control List (ACL) and send network traffic that should be rejected. Implementation of vulnerability leads to a violation of the automated system integrity.

In the same way as for the CVE-2012-0384 vulnerability, we will calculate the *iRisk* value.

Base Score Metrics {Attack Vector = Network; Attack Complexity = Low; Privileges Required = None; User Interaction = None; Scope= Changed; Confidentiality Impact = None; Integrity Impact = Low; Availability Impact = Impact None}

The value *Vulnerability* = 5.8, by the CVSS v3.0 calculator (Fig. 6).

The calculation of the value *Threat = 1.4 · 0.72 · 100 = 108*, so the value *iRisk = (5.8 · 108) - 610 = 16.4*, which means that the vulnerability will be approximately equal to zero, that is we can conclude that this vulnerability can be exploited by an attacker with little probability.

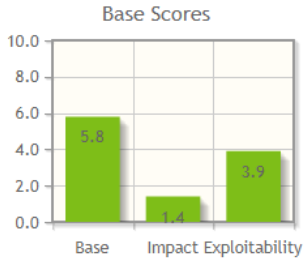

Fig. 6. The Base CVE-2012-1342 vulnerability metric for the cybersecurity polygon

3.3 EternalBlue Vulnerability (CVE-2017-0144)

This vulnerability uses the vulnerability in the implementation of the Server Message Block v1 protocol (SMB). An attacker, having formed and transmitted to a remote host a specially prepared package, is able to get remote access to the system and run any code.

Calculate the iRisk value for CVE-2017-0144 EternalBlue vulnerability.

The base EternalBlue vulnerability metric will have the following parameters. The result is shown in Fig. 7

Base Score Metrics {Attack Vector = Network; Attack Complexity = High; Privileges Required = None; User Interaction = None; Scope= Unhanged; Confidentiality Impact = High; Integrity Impact = High; Availability Impact = High}

Since the attack is conducted from the outside and its' probability is very high, the attacker should be an hacking expert, according to the iRisk method in this case, the value *Impact = 100*, and the value *Likelihood = 0.7* and the value *Threat =70*,

So, you can calculate the *iRisk* value for CVE-2017-0144, without the security patch from March 14, 2017: $iRisk = (8.1 \times 70) - 0 = 567$.

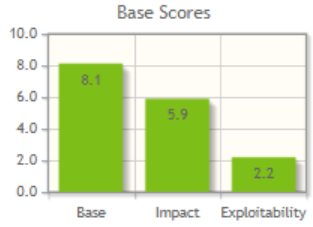

Fig. 7. The Base CVE-2017-0144 EternalBlue vulnerability metric for the cybersecurity polygon

3.4 Meltdown Vulnerability (CVE-2017-5754)

Vulnerability exploits the effect of out-of-order execution in modern processors. Attack doesn't depend on the operating system and doesn't exploit software vulnerabilities. Meltdown actually breaks down the entire security system based on the isolation of the address area, including the virtual one. Meltdown allows you to read part of the memory of other processes and virtual machines. The KAISER patch excludes this vulnerability, but reduces CPU performance.

Calculate the *iRisk* value for a cybersecurity polygon, without KAISER patch.

Calculate the base metric for Meltdown vulnerability (CVE-2017-5754), the result is shown in Fig. 8.

Base Score Metrics {Attack Vector = Local; Attack Complexity = High; Privileges Required = Low; User Interaction = None; Scope= Changed; Confidentiality Impact = High; Integrity Impact = None; Availability Impact = Impact None}

Considering that the attacker can act both from the outside and inside and the attack can be executed frequently, and the attacker can have just an advanced level of

skills and the attack code is shown in large numbers of articles, all of this will give a correlation value of *Impact = 0.9,* and the value of *Threat* will be equal to *100 · 0.9 = 90.*

The resulting value of *iRisk* for Meltdown (CVE-2017-5754) will be equal to *iRisk = (5.6 · 90) -0 = 504*, because without the KAISER patch this Vulnerability doesn't show itself, and is included in the architecture of most modern processors.

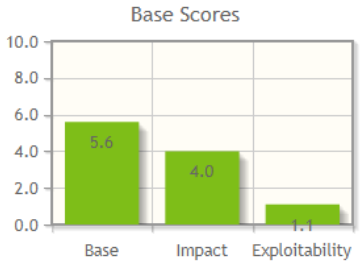

Fig. 8. The Base CVE-2017-5754 Meltdown vulnerability metric for the cybersecurity polygon

3.5 SPECTRE Vulnerability (CVE-2017-5753, CVE-2017-5715)

This vulnerability is assigned two identifiers CVE-2017-5753, CVE-2017-5715. By its nature, it is similar to Meltdown, but with some differences, in particular, by during a speculative code execution, the processor can execute instructions that it would not perform under strictly consistent (non-speculative) calculations, and although in the future the result of their performance is discarded, its imprint remains in the processor cache and can be used.

The Specter vulnerability is not easy to implement - however, it can be implemented, under the condition of attack on a specific software, known to the attacker and, if possible, available in an open source code in the same version and on the same system, which provides an attack.

Another way for Specter implementing is to "predict branching" - the processor has a similar transition prediction block, it predicts the transition address for the next instruction of the indirect transition (Meltdown, but here they play a different role).

For simplicity, this unit does not broadcast between virtual and real addresses, which means it can be trained in the address space of the attacker on certain actions.

After some time, the real transition address will be deducted, the processor identifies the error and rejects the results of the speculative execution, however, as in all other instances of the use of Meltdown and Specter, most performance results remain in the cache.

Calculate the *iRisk* value for the Specter vulnerability. The base metric in both versions of the vulnerabilities implementation is the same, the results of the calculation are presented in Fig. 9.

Base Score Metrics {Attack Vector = Local; Attack Complexity = High; Privileges Required = Low; User Interaction = None; Scope= Changed; Confidentiality Impact = High; Integrity Impact = None; Availability Impact = Impact None}

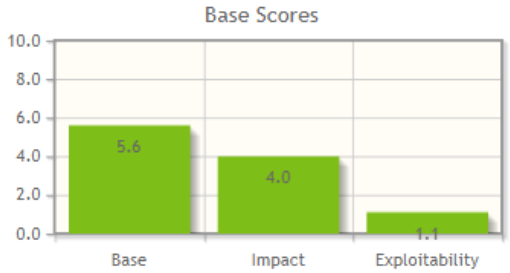

Fig. 9. The Base Spectre CVE-2017-5753 i CVE-2017-5715 vulnerability metric for the cyber-security polygon

In both cases with Spectre, we are concerned with the fact that the processor learns fast to execute one process by using as an example another process, thereby actually allowing the second process to control the progress of the first one. There are no universal patches to fix Specter, and ways of protection from CVE-2017-5715 are the permanently clearing the cache and cleaning the code from the core.

Calculate the *iRisk* value for CVE-2017-5715, given the complexity of the exact implementation and the impact only on the information confidentiality. So the value of *Impact = 50* (including financial, reputational and strategic impact). Given that the vulnerability will be try to use mainly from the outside and the attacker must have advanced technical skills, the correlation value *Likelihood = 0.64*. These parameters are typical for both CVE-2017-5753 and CVE-2017-5715.

However, the *Controls* parameters in this case need to be evaluated in different ways. There are patches for CVE-2017-5715 vulnerability, which partially solve this problem only in some cases, so value *Controls* can be considered *Initial/Ad-Hoc = 100,* but it's provides only some protection value. As to CVE-2017-5753 vulnerability, value *Controls* can be considered as *0*, as this problem is not resolved at this time.

So, for CVE-2017-5715 $iRisk = (5.6 \cdot 50 \cdot 0.64) - 100 = 79.2$.

For CVE-2017-5753 $iRisk = (5.6 \cdot 50 \cdot 0.64) - 0 = 179.2$

4 Conclusions

The iRisk method was chosen for the research, first of all because this technique is free, enough informative, includes another CVSS v3 vulnerability assessment method, which is actively supported by the National Institute of Standards and Technology. Automated system has been tested for the following vulnerabilities: Cisco IOS Arbitrary Command Execution Vulnerability (CVE-2012-0384), Cisco Access Control Bypass Vulnerability (CVE-2012-1342), EternalBlue (CVE-2017-0144), Meltdown (CVE-2017-5754), Specter (CVE-2017-5753) (CVE-2017-5715). Conclusions have been shown about the stability of the designed network to specific threats by the iRisk method. It uses the values from 0 to 1000 scope, where 0 corresponds to automated system, in which it is possible to neglect this vulnerability, whereas at the maximum

value, if it exceeds 100, it is necessary to solve this vulnerability. The results of calculations are given in Table 1.

Table 1. Table of iRisk values for a builted cybersecurity polygon

Vulnerability	*Value iRisk*
Cisco IOS Arbitrary Command Execution Vulnerability (CVE-2012-0384)	50
Cisco Access Control Bypass Vulnerability (CVE-2012-1342)	16.4
EternalBlue (CVE-2017-0144)	567
Meltdown (CVE-2017-5754)	504
Spectre (CVE-2017-5715)	79.2
Spectre (CVE-2017-5753)	179.2

The higher the value iRisk the vulnerability is the more critical and has a higher priority for automated system protection.

References

1. Klaus Wehrle, James Gross. Modeling and Tools for Network Simulation. Hardcover: 256 p. (2010).
2. Korniyenko, B. Model of Open Systems Interconnection terms of information security. Science intensive technology, № 3 (15), pp. 83 – 89., doi.org/10.18372/2310-5461.15.5120 (ukr) (2012).
3. Korniyenko, B., Yudin, O., Novizki,j E. Open systems interconnection model investigation from the viewpoint of information security. The Advanced Science Journal, issue 8, pp. 53 – 56. (2013).
4. Korniyenko, B., Yudin, O. Galata, L. Research of the Simulation Polygon for the Protection of Critical Information Resources. CEUR Workshop Proceedings, Information Technologies and Security, Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security (ITS 2017), Kyiv, Ukraine, November 30, 2017, Vol-2067, - P.23-31, urn:nbn:de:0074-2067-8 (2017).
5. Chris Clymer, Ken Stasiak, Matt Neely, Stephen Marchewitz. IRisk Equatuion Available via https://securestate.en/iRisk-Equation-Whitepaper.pdf
6. Common Vulnerability Scoring System v3.0: User Guide. Available via https://www.first.org/cvss/user-guide
7. Korniyenko, B., Yudin, O. Implementation of information security a model of open systems interconnection. Abstracts of the VI International Scientific Conference "Computer systems and network technologies» (CSNT-2013), p. 73. (2013).
8. Korniyenko, B. Information security and computer network technologies: monograph. ISBN 978-3-330-02028-3, LAMBERT Academic Publishing, Saarbrucken, Deutschland, 102 p. (2016).
9. Korniyenko, B., Galata, L., Kozuberda, O. Modeling of security and risk assessment in information and communication system. Sciences of Europe, V. 2., No 2 (2), pp. 61 -63. (2016).
10. Korniyenko, B. The classification of information technologies and control systems. International scientific journal, № 2, pp. 78 - 81. (2016).

11. Korniyenko, B., Yudin, O. Galata, L. Risk estimation of information system. Wschodnioeuropejskie Czasopismo Naukowe, № 5, pp. 35 - 40. (2016).
12. Korniyenko, B., Galata, L., Udowenko, B. Simulation of information security of computer networks. Intellectual decision making systems and computing intelligence problems (ISDMCI'2016): Collection of scientific papers of the international scientific conference, Kherson, Ukraine, pp. 77 - 79. (2016).
13. Korniyenko, B. Cyber security - operating systems and protocols. ISBN 978-3-330-08397-4, LAMBERT Academic Publishing, Saarbrucken, Deutschland, 122 p. (2017).
14. Korniyenko, B., Galata, L. Design and research of mathematical model for information security system in computer network. Science intensive technology, № 2 (34), pp. 114 - 118. (2017).