Information Security Journal: A Global Perspective

ISSN: 1939-3555 (Print) 1939-3547 (Online) Journal homepage: https://www.tandfonline.com/loi/uiss20

Protection from reconnaissance for establishing information security systems

Mikhail Styugin

INFORMATION SECURITY JOURNAL: A GLOBAL PERSPECTIVE
https://doi.org/10.1080/19393555.2019.1630528

Taylor & Francis
Taylor & Francis Group

Protection from reconnaissance for establishing information security systems

Mikhail Styugin

Research Department, Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russia

ABSTRACT
The paper presents a generalized method for improving security of information systems based on protection of the systems from reconnaissance by adversaries. Attacks carried out by exploiting almost all vulnerabilities require particular information about the architecture and operating algorithms of an information system. Obstructions to obtain that information also complicates carrying out attacks. Reconnaissance-protection methods can be utilized for establishing such systems (continuous change of attack surface). Practical implementation of the techniques demonstrated their high efficiency in reducing the risk of information resources to be cracked or compromised.

KEYWORDS
Information security; moving terget defense; protection from reconnaissance; authentication system

1. Introduction

The paper reviews a significant security problem for information systems and presents a solution method based on the group of methods and technologies, which were described in the referred works (Styugin, 2014a, 2014b, 2015a, 2015b, 2016a, 2016b, 2017; Styugin & Kytmanov, 2015; Styugin & Parotkin, 2016; Styugin, Zolotarev, Prokhorov, & Gorbil, 2016).

In order to analyze information processes from the viewpoint of security, the analysis should be performed at the level of formalized models. Formalized models and their proofs of security are built with consideration of assumptions and restrictions applicable to the model. Such conditions and restrictions can be found in the formal analysis of information cryptosystems (Katz & Lindell, 2007), access management policy models (Benantar, 2006), analysis of information streams (Verbeek et al., 2015), etc. However, general ongoing complication of computer information systems does not allow conducting their full formalization and ensuring adequacy in practical implementation of the formalized models. For example, the absolutely secure Vernam cipher (one-time pads) (Katz & Lindell, 2007). At the formal model level, it has been proved that a ciphertext does not disclose any information related to the initial text. However, a cipher system may have many vulnerabilities at the implementation level. Those can be leaks through side channels, which can allow an attacker measure impulse signals and recover the encryption key or the internal parameters of the pseudorandom number generator used for creating the encryption key. Perhaps, the system's administrator would give the initial plaintext to the adversary or the adversary may plant an encryption key by a remote code injection. Thus, a system with a formalized proof of security has a great number of potential threats at the level of its implementation, and many of those threats may be unpredictable at the system's development stage. One of the speeches at the International Conference on Information Warfare and Security in 2013 (Kraft, Rohret, Vella, & Holston, 2013) presented research results, which demonstrated that the most complex systems are generally more vulnerable to the most simple attacks and exploits. The above observation was called the Adam and Eve Paradox.

Hence, the process of ensuring information security is increasingly becoming a "hole patching" process. All vulnerabilities cannot be detected and rectified at the development stage; therefore, systems have to be analyzed when they are in operation and vulnerabilities have to be eliminated when subjected to attacks. The above case is unacceptable for systems with processes and information of critical importance, and they require new approaches to security of information systems. In the past few years, those approaches began to be formed as protection of information systems from reconnaissance. Adversaries need to have

CONTACT Mikhail Styugin styugin@gmail.com Research Department, Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russia

some information about the system to carry out an attack. In case information cannot be obtained; therefore, vulnerabilities cannot be exploited. Such approaches may employ technologies for continuous modification of information systems or moving target technologies (Jajodia, Ghosh, Swarup, Wang, & Wang, 2011), informational noninfluence (Oheimb, 2004), and implementation of unique information processes (Styugin, 2014a).

One can find attempts to obtain similar methods in earlier publications. For example, Cho & Ben-Asher (2018) give a generalized method of developing security systems based on the technology of moving target defense as a way to reduce the cost of security systems with respect to the cost of making an attack on the system. However, it does not offer any specific algorithms for implementing such systems in the particular case.

In addition, there are many abstract methods for generating strategies that are protected from information system exploration, for example, based on the Markov game theoretic approach (Lei, Zhang, Wan, & Liu, 2018) or the Bayesian attack graph (Zangeneh & Shajari, 2018). However, these methods only offer effective scenarios for the application of specific security elements, but do not provide methods to build protection algorithms for particular information systems.

The present paper provides a summary of solutions for system protection from research, which we developed in the past few years, it presents the algorithm of their practical implementation, and the obtained results are evaluated.

2. The algorithm for transforming an information system into a reconnaissance-secure system

A general algorithm for transforming an information system (IS) into a reconnaissance-secure system is shown in Figure 1.

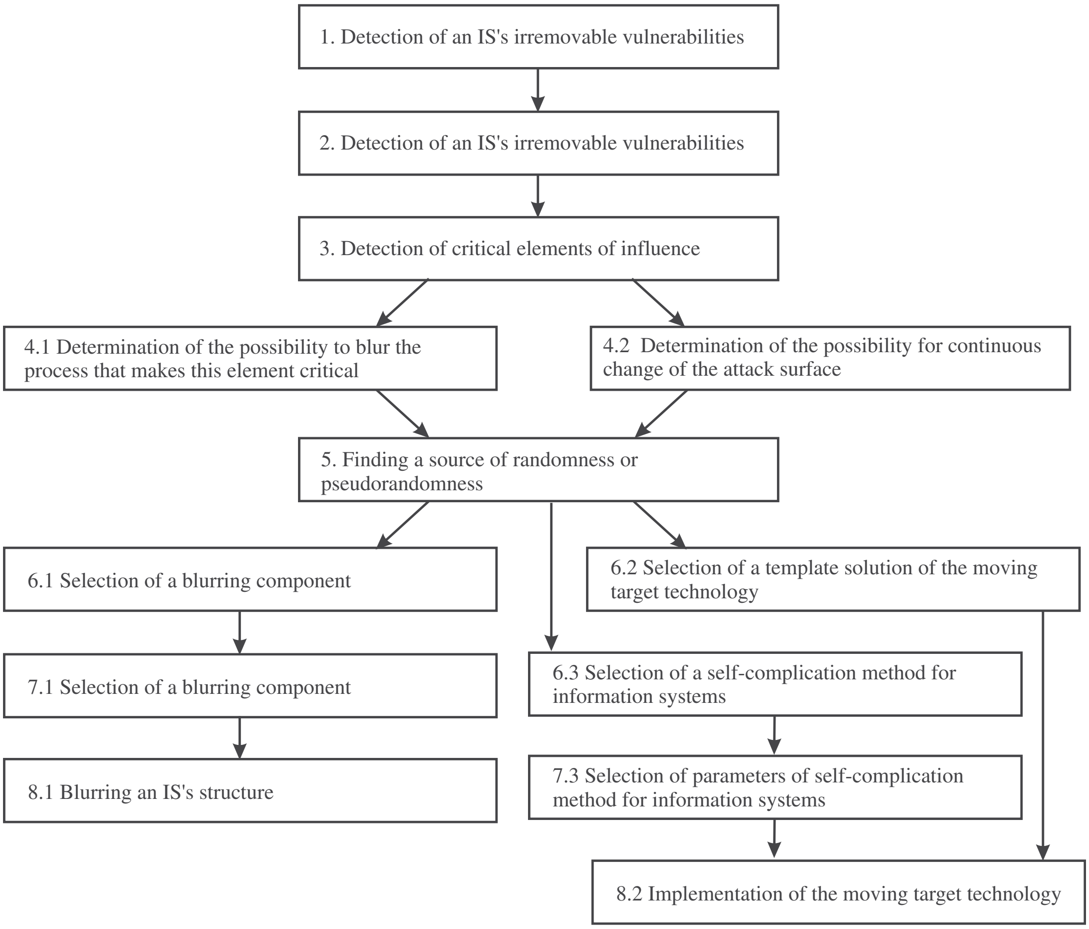
1. Detection of an IS's irremovable vulnerabilities
2. Detection of an IS's irremovable vulnerabilities
3. Detection of critical elements of influence
4.1 Determination of the possibility to blur the process that makes this element critical
4.2 Determination of the possibility for continuous change of the attack surface
5. Finding a source of randomness or pseudorandomness
6.1 Selection of a blurring component
6.2 Selection of a template solution of the moving target technology
6.3 Selection of a self-complication method for information systems
7.1 Selection of a blurring component
7.3 Selection of parameters of self-complication method for information systems
8.1 Blurring an IS's structure
8.2 Implementation of the moving target technology

Figure 1. The general algorithm for creating an information system (IS) protected from reconnaissance.

A detailed analysis of each stage of the above scheme is provided below.

At the first stage, IS's irremovable vulnerabilities should be detected. Presence of irremovable vulnerabilities means that the system has some information or data, which may compromise the system. It can be the case when the encryption key is stored in the system. The encryption key is the information, which can be used for hacking the IS. What is more, the key cannot be removed from the system, because in that case, the system will not be able to carry out one of its functions. Thus, storing the encryption key inside the IS is an unavoidable vulnerability. At this stage, it is not important for us how the information can be retrieved from the system. We may not know it at the stage of process development. The only important point is that there is information and it may be critical for the system if the information is leaked. Unverifiable assumptions, for example, can also be considered as unavoidable vulnerabilities. Unverifiable assumptions can include the fact that the pseudorandom number generator produces an unpredictable number sequence.

At the second stage, the IS's security-critical processes should be determined. Encryption algorithms, pseudorandom number generation algorithms, database interaction interface, etc.

At the third stage, critical elements, which create vulnerabilities described in the first stage, should be determined. The encryption key, the seed random number generator, the user data read operator for a database query, etc., may be those elements.

The first three stages are implemented to detect vulnerabilities, as well as the processes and the elements related to them. All of them are an integral part of the information system. Then, we should determine the methods for system protection against reconnaissance, which are employed not to let an attacker get any information sufficient for exploiting the vulnerabilities, while the critical elements and processes should remain unchanged. To do so, we have the two main areas, defined in (Styugin, 2014a, 2014b, 2015a, 2015b, 2016a, 2016b, 2017; Styugin & Kytmanov, 2015; Styugin & Parotkin, 2016; Styugin et al., 2016), which are process-blurring and continuous change of attack surface. At stage 4, the practicability of implementing both methods should be determined.

At stage 4.1, the possibility of blurring the process that makes the element critical should be determined. To do so, we should determine whether we could transfer a critical process or parameter into the area of a greater number of elements. An encrypted data transfer channel is taken as an example. The data are encrypted with some algorithm and transmitted to the recipient. The transmitted data can be enclosed into some other data or a random bit sequence. Thus, even having the key and the encryption algorithm will not allow finding the data encrypted in the channel. The generator's algorithm can be blurred in a similar manner using the random number generator used for encryption key generation. Then, the algorithm will be established in a space of a greater number or an infinite number of possible algorithms, and the critical seed generator's parameters obtained by an adversary will not allow predicting its generated sequence.

At step 4.2, the possibility of continuous change of the attack surface should be considered, e.g., when a key or an encryption algorithm, the data transmission channel, etc., changes continuously. The information obtained by an attacker becomes irrelevant after the next point in time.

When the possibility and the reconnaissance-protection technique are determined at step 4.1 and 4.2, then the source of random or pseudorandom numbers should be found. Paper (Styugin, 2016b) proved that the space of alternatives for reconnaissance protection is limited by the unpredictable random/pseudorandom values, which can be gathered within one system. In order to obtain pseudorandom components involving untrusted sources, the unpredictability can be achieved by their multiplication as shown in the paper (Styugin, 2016b).

Then depending on the reconnaissance-protection technique chosen, we go to step 6.1, 6.2 or 6.3.

Steps 6.1 and 7.1 imply establishing an information process-blurring scheme. The principal problem solved at that step is defining the method for building the space of alternatives in which the algorithms or parameters will be blurred. Given pseudorandom number generator G, which performs some algorithm A. The space of alternatives for $\mathbf{A}$ should be determined, so that A ⊠ $\mathbf{A}$. While $|\mathbf{A}|$ should be large enough to eliminate the possibility of exhaustive enumeration. Each one of $A \in \mathbf{A}$ should also have different output values because it is the critical blurring parameter. Therefore, determining the blurring structure implied the space generation principle for **A**. In some cases, standard solutions can be used, which are provided in papers (Carvalho & Ford, 2014; JafarHaadiJafarian & Al-Shaer, 2012; Li & Sekar, 2006; Paulos et al., 2013) or another algorithm for generation of value spaces can be developed.

When the possibility to change the process using the moving target technique is found, the process goes to steps 6.2 and 6.3. At step 6.2, a standard moving target defense solution can be selected. There over 200 solutions in this area appeared in the last 5 years. They concern, for example, processes like network addressing (Carvalho & Ford, 2014; JafarHaadiJafarian & Al-Shaer, 2012), generating unique program code (Styugin et al., 2016), and establishment of a changing document control structure (Li & Sekar, 2006), etc.

If no ready solutions with moving target defense are found; then, we proceed to steps 6.3 and 6.7. At this point, the universal technique for establishing self-complicating information systems presented in (Styugin, 2018) is suggested for implementation. The technique's principle is in generation of intermediary blocks in different processes. Such blocks can, for example, access a database and modify a query. Legal queries to a database are modified so that they have the correct form at the intermediary block's output. Whereas unauthorized queries cannot be generated correctly. At that stage, processes which can be processed functionally, should be selected. Hence, we can determine the principle of their direct and reverse modification. If, for example, it is an integer, then addition and subtraction operations can be performed with a specific invariable. If it is a database query, then every word, which is a command, can be concatenated with a specific index and then it can be subtracted when a query is made. When the direct and inverse modification principle is found, then its automatic generation can be made in a program. The result will be a self-complicating information system defined in (Styugin, 2018).

At the last steps 8.1 and 8.2, the developed moving target defense techniques and information blurring methods are implemented in software development or in ready template solutions.

Examples of implementing the technique in information systems security and evaluation of the results are reviewed in the next section.

3. Establishment of an authentication system protected from reconnaissance

The DKAuth authentication system (dkauth.com) is presented as an implementation example of a system protected from reconnaissance. The password authentication system stands as a separate element of the system. The unavoidable vulnerabilities are listed below:

- authentication data storage (user accounts);
- key generation for encrypting authentication data by using pseudorandom number generators.

The first point implies the obvious requirement that the authentication data, i.e., user accounts, must be stored all the time. There is a large number of techniques to avoid their exposed storage. For example, passwords are stored after encryption or after implementing one-way hash functions. This certainly makes it more difficult for an adversary to succeed, but it does not eliminate the vulnerability. Passwords are stored in the system anyway and the system must recognize a user that has been registered in the system. Collisions that advance the enumeration may be found in one-way hash functions, and data encryption requires encryption key storage, which should have a storage.

The second vulnerability appears in case encryption keys for user accounts should be generated or when random values are involved in account generation (will be shown later on).

Now we can determine critical processes connected to the above vulnerabilities. The first process compares the password entered by a user to the password previously registered in the system. The process cannot be avoided and it is a critical one. The other critical process is implementation of a pseudorandom bit sequence generator for obtaining an encryption key and using it for securing user accounts.

The DKAuth's system employed the moving target defense technology for the first vulnerability, and system parameter blurring for the second vulnerability (Styugin, 2016b). The technology of self-complicating systems was utilized as well. User accounts were stored as distributed pieces encrypted with keys that were generated by using self-complicating information systems technology. That is, the correct chain of keys, which is working only when the correct password is entered and it cannot be found at the system's administrator level regardless of full data access.

The pseudorandom number generation algorithm was blurred in the set of the same algorithms with different seed values. That is, the pseudorandom number generator was used for the whole-distributed system of remote-untrusted hosts, which was essentially functioning as one unit. This multiplication of generation algorithms does not allow predicting the generator's operation even when each separate algorithm is predictable.

Below is comparison of the password authentication system's security properties in the regular form and with implementation of DKAuth. The DKAuth was launched in December 2015. Now it has about 10 servers in several countries and 2630 user accounts. Its main function is to provide the AuthaaS service for password authentication at websites. Therefore, it would be more appropriate to consider parameters in this area.

As reported in (Kaspersky, 2015), user account hacking is a major problem on the Internet. Around 23% of all the news in the information security sphere concerns it. The total share of internet accounts that were cracked is 71%. Around 23% of the work user accounts on the Internet are hacked every year.

We have the DKAuth's statistical data for the past year and a half. There were no incidents within that period of time. Therefore, we have a decline as shown in Figure 2 in comparison with the average values. The average cracking value for 2630 user accounts in the same timeframe as mentioned above could have been 907. Certainly, the above statistical data cannot be accurate, as no accumulated data in a limited timeframe cannot be considered as sufficient. It may be possible that next month an incident could result in theft of the whole account database. Hence, other evaluations with consideration of lower risk factors.

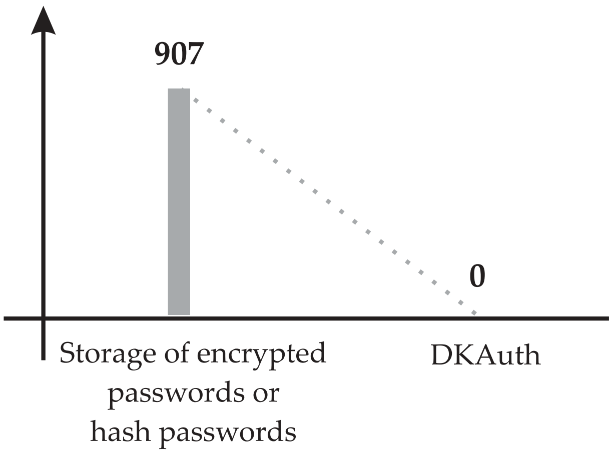

Figure 2. Statistical evaluation of user account hacking.

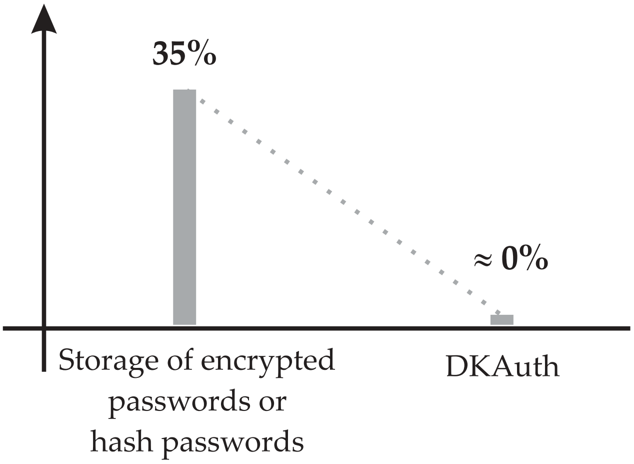

Figure 3. Evaluation for threats of compromising server data.

As we are considering only vulnerabilities of account theft directly from the database on the server, the comparison can be made only for the above list of threats. The results are shown in the plot in Figure 3.

Threats for DKAuth above are close to zero. That is so because after blurring and continuous change of key generation chains, we established a system, which basically has no centralized storage for user accounts. Consequently, compromising and cracking the server does not provide any advantages to an attacker as well as to the security administrator. However, it is possible that types of threats aimed particularly at this service, when, for example, the administrator would create and inject a script which intercepts a password when it is sent to the server during authentication. Thus, someone will not be able to copy the database, but they may get some separate accounts. Therefore, the threat stated in the second point is regarded as close to zero, however not inexistent.

Both the above evaluations are considering a decrease of some parameters, which indicate the system's potential hacking hazard, down to zero. Those parameters do not provide and objective evaluation of changes in the system's general security level. It seems obvious that the system hacking hazard did not become equal to zero after a body of threats is embedded. For obtaining a more objective evaluation, the security evaluation method for reconnaissance-protected systems.

4. The security evaluation method for reconnaissance-protected systems

Any kind of reconnaissance-protection systems imply increasing a system's security level. In order to define whether the system security has increased and by how much, it is required to choose the evaluation method of the security level.

When unauthorized access security is considered, only deliberate attacks are taken into account. Deliberate attacks are performed as a method for exploiting a system's vulnerabilities. The below structure can be defined:

(1) The system has unavoidable vulnerabilities (presence of protected information, algorithms, encryption keys stored, etc.).
(2) The vulnerability and the method to exploit in aggregate are the threat.
(3) Each threat can be exploited by an adversary (to attack) given it has sufficient information. There can be many attacks related to one threat.

Therefore, an unavoidable vulnerability V_1 can be defined in the system. There is a set of threats of exploit $\mathbf{T}^{\mathbf{V1}} = \{T_1^{V1}, T_2^{V1}, \ldots\}$ for the vulnerability. There is also a set of attacks $\mathbf{A}^{\mathbf{T1(V1)}} = \{A_1^{T1(V1)}, A_2^{T1(V1)}, \ldots\}$ for each threat. A combined set of **T** and **A** for all vulnerabilities is obtained, where,

$$\mathbf{T} = \mathbf{T}^{\mathbf{V1}} \cup \mathbf{T}^{\mathbf{V2}} \cup \ldots \cup \mathbf{T}^{\mathbf{Vn}}$$

$$\mathbf{A} = \mathbf{A}^{\mathbf{T1(V1)}} \cup \mathbf{A}^{\mathbf{T1(V2)}} \cup \ldots \cup \mathbf{A}^{\mathbf{Tm(Vn)}}$$

After all, real incidents in the system are defined by set **A**. The greater is the set's cardinality, the more secure the system would be. The problem of comprehensive evaluation of information systems is that the completeness of set **A** cannot be evaluated. However, the system can be checked for vulnerabilities that define threat subsets and the related attacks at the information system.

For example, the requirement for authentication data storage:.

$$\mathbf{T}^{\mathbf{auth}} \subset \mathbf{T} \Rightarrow \mathbf{A}^{\mathbf{T(auth)}} \subset \mathbf{A}$$

The requirement for transmission of data through public networks:

Assume the initial system is defined by sets $\{\mathbf{V_0}, \mathbf{T_0}, \mathbf{A_0}\}$. After implementation of some information system reconnaissance-protection technologies, set $\{\mathbf{V_{PfR}}, \mathbf{T_{PfR}}, \mathbf{A_{PfR}}\}$ is created. The main reason for implementing reconnaissance protection of information systems is, as stated above, the requirement for protection from vulnerabilities and threats, which are undetected at a given design stage. Therefore, a set of threats and vulnerabilities is assumed to be constant, that is, $\mathbf{V_0} = \mathbf{V_{PfR}}$, $\mathbf{T_0} = \mathbf{T_{PfR}}$. However, the system reconnaissance protection eliminates the possibility of existing threats by complicating acquisition of the information for carrying out attacks. Hence, the subsets of set **A** are removed, which correspond to the classes of threats and vulnerabilities. Such vulnerabilities in the DKAuth system are:

(1) User account data have to be stored in the authentication system;
(2) Encryption keys must be generated based on a pseudorandom number generators with an unprovable strength.

A set of possible attacks based on exploiting all the above vulnerabilities are fended off by protecting the system from reconnaissance. Hence, if one or several vulnerabilities stated above are included in set $\mathbf{V_0}$; then,

$$|\mathbf{A_{PfR}}| < |\mathbf{A_0}|$$

Hence, changes in the system's qualitative security indexes can be proved. It may be insufficient in some cases. When money is invested in a security system, it should be demonstrated how much the risk of being cracked is reduced after implementation of the techniques.

For obtaining quantitative indexes, elements from set **A** should be matched with weight factors K_A. Each factor demonstrates the probability of a particular attack in the aggregate of the attacks. That is, weight factors of the whole set A yield a sum of 1.

When a weight factor of an individual attack cannot be determined, the total weight factor for an individual threat $\mathbf{A}^{\mathbf{T1(V1)}} = \{A_1^{T1(V1)}, A_2^{T1(V1)}, \ldots \}$ of a vulnerability $\mathbf{A}^{\mathbf{V1}} = \{A_1^{T1(V1)}, A_2^{T1(V1)}, \ldots, A_m^{Tn(V1)}\}$ can be implemented.

For example, the following quantitative evaluation is considered for the DKAuth system. Threats, which may compromise a user account, shall be identified for establishing the evaluation.

According to (Gartner, 2016), such threats may be the following:

(1) Social engineering methods (T_1) employed by an attacker;
(2) Copying of user account database by the service administrator (T_2);
(3) Theft of user account database information by compromising the database's server (T_3);
(4) Password interception via an unsecured transmission channel (T_4);
(5) Password theft from a user's computer (T_5).

Thus, a combined set $\mathbf{T} = \{T_1, T_2, T_3, T_4, T_5\}$ is defined. Statistically (Kaspersky, 2015), 35% of all eventually stolen accounts is attributed to threats T_2 and T_3. Thus, the total factor can be found, $K_{\mathbf{A}(V1)} + K_{\mathbf{A}(V2)} = 0.35$. Considering that probability for all summary threats is 71%, as stated above; thus, a diagram shown in Figure 4 is resulting.

Thus, it has been established that the summary risk factor for system's account cracking is reduced by 25%.

The result of the above analysis is that it can be stated that security of password authentication was improved by implementing system blurring methods and the moving target defense.

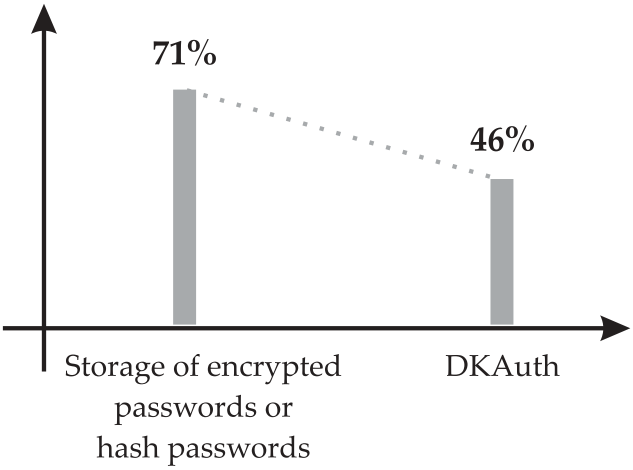

Figure 4. Evaluation by the summary amount of threats.

5. Establishing a reconnaissance-secured network and document flow

BSRouter is another implementation of a reconnaissance-secured system. The solution was developed using the facilities RTK-Sibir Closed Joint Stock Company and it is a network dynamic routing and addressing system. The system incorporates a small program installed in a router and a server part available at a public IP address. The software in a router sets up a secure connection with the server and receives commands from it.

Primary router's functions are the following:

(1) Modifying parameters of packets, which are transmitted to extranet, for protection from external scanning.
(2) Establishing a dynamic channel for Internet access via intermediary hosts, which act as proxy servers.
(3) Modifying real IP addresses of the network's devises when some specific circumstances occur.

The system implements the moving target defense technology for protection from internal and external network scanning and for protection from data reconnaissance of the internet channel.

The BSRouter hardware and software system was installed in the networks of GuardNet Company and Kairos Company. There appeared a problem of evaluating the changing security risks after the security solution was implemented.

The technique presented above is employed for evaluation of changing risks. Combined threats to an information system are considered here.

A very rough classification of threats to information facilities of companies is as follows:

(1) Social engineering attacks without implementation of technical exploits (compelling to provide a password or any other information).
(2) Social engineering attacks with implementation of technical exploits (compelling to launch a Trojan file).
(3) Internal attacks by taking advantage of vulnerabilities and errors of software.
(4) External attacks with a possibility to establish a connection to a company's host.
(5) Attacks at services that receive external queries.
(6) Exploiting the opportunity of scripts to be downloaded and executed by internal hosts.

Protection from reconnaissance provided by BSRouter covers threats mentioned in points 2, 4, 5, and 6 above, one way or another to a variable degree though. In order to demonstrate this, attacks should be considered, which would require information on network topology and services running there, format of packets transmitted to extranet and their source. This category includes attacks in point 2 which are aimed at launching Trojan software with a preset packet exchange algorithm in a company's network; attacks in point 4 which analyze transmitted network traffic; attacks in point 5, which inject scripts to interact with a corporate network; and attacks in point 6, which again are aimed at launching Trojan software with a preset packet exchange algorithm in a company's network.

Further to analyzing the Kaspersky Lab's 2016 Annual Report (Kaspersky, 2016), the attacks applicable to our problem can be defined and a list of Trojan malware and exploits according to the four items selected can be compiled. Statistical information in the above report covers all the attacks. The following results are obtained. Point 2: 1.3%; point 4: 72.1%; point 5: 12.3%; point 6: 3.1%. Then, incident probability in the annual average for each group of threats is determined as follows: Group 1: 63.4%, group 2: 88.4%, group 3: 10.2%, group 4: 21.5%, group 5: 33.8%, group 6: 31.2%. Therefore, if an incident occurs in a system (i.e., the summary probability of all events is 100%); then, probability of the incident can be attributed to each of the above threats $T_1 = 0{,}25$; $T_2 = 0{,}36$; $T_3 = 0{,}04$; $T_4 = 0{,}09$; $T_5 = 0{,}14$; $T_6 = 0{,}12$. Those values can be used for the probability of events for specific individual attacks A_i^{Vn}.

$A^{T2} = 0{,}25 \text{ x } 0{,}013 = 0{,}00325$

$A^{T4} = 0{,}09 \text{ x } 0{,}721 = 0{,}06489$

$A^{T5} = 0{,}14 \text{ x } 0{,}123 = 0{,}01722$

$A^{T6} = 0{,}12 \text{ x } 0{,}310 = 0{,}03720$

Finally, fending that group of potential threats the probability of incident occurrence in a system was reduced by 12.26%.

6. Conclusions

The problem of security for complex information systems, which concerns the fact that all potential attacks at a system cannot be detected at the system's design stage, is formalized in the paper. That problem requires new research trends in the field of information security. The solution to the problem is implementing external reconnaissance-protection techniques for information systems. The paper presented the algorithm for implementing a group of solutions and techniques reviewed in the past three years in papers (Styugin, 2014a, 2014b, 2015a, 2015b, 2016a, 2016b, 2017; Styugin & Kytmanov, 2015; Styugin & Parotkin, 2016; Styugin et al., 2016). The method for evaluating results after implementation of such systems and for increasing security level of computer information systems.

Funding

This work was supported by the Russian Foundation for Basic Research (RFBR) according to the research project No. 16-29-09456 ofi m.

ORCID

Mikhail Styugin http://orcid.org/0000-0001-9746-2719

References

Benantar, M. (2006). *Access control system. Security, identity, management and trust model.* Springer US, New York City.

Carvalho, M., & Ford, R. (2014). Moving-target defenses for computer networks. *IEEE Security and Privacy*, *12* (2), 73–76. doi:10.1109/MSP.2014.30

Cho, J.-H., & Ben-Asher, N. (2018). Cyber defense in breadth: Modeling and analysis of integrated defense systems. *Journal of Defense Modeling and Simulation*, *15* (2), 147–160. doi:10.1177/1548512917699725

Gartner. (2016). New paradigms of digital identity: Authentication and authorization as a service (AuthaaS). Retrieved from: https://www.elevenpaths.com/wp-content/uploads/2015/10/Telefonica_LVTI2N.pdf

JafarHaadiJafarian, Q. D., & Al-Shaer, E. (2012). Openflow random host mutation: Transparent moving target defense using software-defined networking. *Proceedings of the 1st Workshop on Hot Topics in Software Defined Networking (HotSDN*), Helsinki, Finland, (pp. 127–132).

Jajodia, S., Ghosh, A. K., Swarup, V., Wang, C., & Wang, X. S. (2011). *Moving target defense. Creating asymmetric uncertainty for cyber threats. Series: Advances in information security*. Springer-Verlag, New York.

Kaspersky. (2015). Security bulletin. Retrieved from: https://securelist.ru/files/2015/12/Kaspersky-Security-Bulletin-2015_FINAL_RUS.pdf

Kaspersky. (2016). Security bulletin. Retrieved from: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07182317/KASPERSKY_SECURITY_BULLETIN_2016.pdf

Katz, J., & Lindell, Y. (2007). *Introduction to modern cryptography, second edition*. New York, USA, NY: CRC Press.

Kraft, M., Rohret, D., Vella, M., & Holston, J. (2013). The adam and eve paradox. *8th International Conference on Information Warfare and Security, ICIW 2013*, Denver, Colorado, USA, (pp. 275–283).

Lei, C., Zhang, H.-Q., Wan, L.-M., & Liu, L. (2018). Incomplete information Markov game theoretic approach to strategy generation for moving target defense. *Computer Communications*, *116*, 184–199. doi:10.1016/j.comcom.2017.12.001

Li, J., & Sekar, R. (2006). *Address-space randomization for windows systems. In Preceedings of 2006 Annual Computer Security Applications Conference (ACSAC*), Miami Beach, FL, USA, (pp. 329–338).

Oheimb, V. D. (2004). Information flow control revisited: Noninfluence = noninterference + nonleakage". *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3193*, 225–243.

Paulos, A, Pal, P., Schantz, R. & Benyo B. (2013). Moving target defense (MTD) in an adaptive execution environment. *ACM International Conference Proceeding Series. 8th Annual Cyber Security and Information Intelligence Research Workshop: Federal Cyber Security R and D Program Thrusts, CSIIRW 2013,* Oak Ridge, TN, USA, (pp. 23–34). doi:10.3174/ajnr.A3158

Styugin, M. (2014a). Protection against System Research. *Cybernetics and Systems: an International Journal*, *45*(4), 362–372. doi:10.1080/01969722.2014.904139

Styugin, M. (2014b). The New method of security development for web services based on Moving Target Defense (MTD) Technologies. *Proceedings of the International Conference on Network Security and Communication Engineering (NSCE2014*), Hong Kong, (pp. 112–118).

Styugin, M. (2015a). Absolutely indiscernible data transfer channel. *Proceedings of The 14th European Conference on Cyber Warfare and Security (ECCWS-2015*), Hatfield, UK, (pp. 286–293).

Styugin, M. (2015b). Analysis of awareness structures in information security systems. *Proceeding of The Fourth International Conference on Cyber Security, Cyber Welfare, and Digital Forensic (CyberSec-2015*), Jakarta, Indonesia, (pp. 6–10).

Styugin, M. (2016a). Indistinguishable executable code generation method. *International Journal of Security and Its Applications*, *10*(8), 315–324. doi:10.14257/ijsia

Styugin, M. (2016b) Conditions for creating perfectly secure systems. *IOP Conference Series: Materials Science and Engineering*, Krasnoyarsk, Russia. vol. 155.

Styugin, M. (2017) Indistinguishability of actions in manipulated information systems. *Proceedings of The 12th International Conference on Cyber Warfare and Security (ICCWS-2017*), Dayton, USA (pp. 361–366). doi:10.2147/IJGM.S153794

Styugin, M. (2018). Establishing Systems Secure from Research with Implementation in encryption algorithms. *International Journal of Network Security*, *20*(1), 34–39.

Styugin, M., & Kytmanov, A. (2015). Mathematical modeling of user perception in information security systems. *Mathematics & Physics. Journal of Siberian Federal University*, *8*(4), 80–92. doi:10.17516/1997-1397-2015-8-4-454-466

Styugin, M., & Parotkin, N. (2016). Multilevel decentralized protection scheme based on moving targets. *International Journal of Security and Its Applications*, *10*(1), 45–54. doi:10.14257/ijsia

Styugin, M., Zolotarev, V., Prokhorov, A., & Gorbil, R. (2016) New approach to software code diversification in interpreted languages based on the moving target technology. *Proceedings of the 10th IEEE International Conference on Application of Information and Communication Technologies (AICT 2016*), Azerbaijan, Baku, (pp. 27–31).

Verbeek, F., Havle, O., Schmaltz, J., Tverdyshev, S., Blasum, H., Langenstein, B., Stephan, W., Wolff, B. & Nemouchi, Y. (2015). Formal API specification of the PikeOS separation kernel. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9058*, 375–389.

Zangeneh, V., & Shajari, M. (2018). A cost-sensitive move selection strategy for moving target defense. *Computers and Security*, *75*, 72–91. doi:10.1016/j.cose.2017.12.013