Complex systems and risk management

Stjepan Groš
Faculty of Electrical and Computing Engineering
University of Zagreb
Unska bb, 10000 Zagreb, Croatia
E-Mail: stjepan.gros@fer.hr

***Abstract*—The risk management process, and in particular, risk assessment is a very tedious and error prone process with no exact measure of how it progresses, or even the justification that it reflects the real situation. This is because the whole process heavily depends on the experience of the people doing it. Furthermore, simplifications are done that run just contrary to what the real systems are, *complex systems*! In this paper we argue that all this process has to be done with complexity in mind, as it is complex system, and we outline a novel risk management method based on those premises. It is possible to automate the risk assessment process presented in this paper to a high degree. Also, the risk method has better justifications and is less dependent on the skills of the people doing risk assessment. Finally, progress can be measured by measuring the complexity of the model.**

***Index Terms*—complex systems, information security, risk analysis**

I. Introduction

Complex systems science is a hot topic these days, evidenced, among others, by the growing number of papers dealing with it that can easily be found using any library search tool. As much as this theory holds promise of being very useful, the interest also generates some hype around the field [1]. Still, this topic is not new and it can be traced back to 1940ties [2]. What changed from then till now, and what basically made complex systems so interesting, is that the main tool for the research of complex systems is simulation and for that reason powerful computers, and especially clusters, enabled deeper insight into this subject.

The main premise lying behind complex systems science is that it is not possible to describe, or understand, behavior of a system by only knowing how it's constituent components behave. The reason is that part of the behavior of a whole is also in connections that are formed between it's components. This is quite opposite from the more traditional reductionist approach in which basic components are analyzed and then the behavior of the whole is induced based on that analysis.

The goal of this paper is to try to argue that information systems are complex systems and when risk assessment is performed this risk assessment is done for overly simplified system. In other words, it doesn't take into account complexity of the underlying information system. This in turn has the consequence that the results of the assessment process could significantly differ from the real situation. We also outline a risk assessment method that doesn't ignore complexity inherent in the system, but just on the contrary, takes it as the most important part.

The paper is structured as follows. First, in Section II we describe what complex systems are and we also give brief overview of the current approaches to risk assessment. Then, in the Section III the arguments are given why risk management in general, and risk assessment in particular, are complex and also why this complexity couldn't and shouldn't be removed. In Section IV we outline risk management method based on the premise that risk management should indeed be treated as a *part of the complexity theory*. Finally, the paper finishes with the conclusions and future work in the Section V and bibliography.

II. Preliminaries

A. Complex systems

Majority of today's analysis of systems is based on the, so called, *reductionist approach*. The basic premise of this approach is that it is possible to decompose a system into it's basic components, analyse them separately and, based on those analyses, predict the behavior of the system as a whole. But it turns out that this is not possible in a general case. In other words, there is more in a system than just what's contained in the subparts themselves. So, to understand system's behavior it has to be analyzed as one, single, unit.

The idea of complex systems was first proposed by cyberneticist W. Ross Ashby in the 1940s. In 1960s and 1970s it was further developed by physicists and chemists. In the 1980s emerging mathematics of nonlinear dynamics and chaos was used to further improve understanding of complex systems which made the field mostly quantitative, mathematical and practiced by physicists. During 1980s another approach also emerged, pioneered by the Santa Fe Institute, that is more in line with the Ashby's work and that is also more qualitative and rooted in computer science [2].

To give the definition of a complex system isn't so easy [3], if possible at all. It is intuitively clear that complex systems are situated somewhere between complete order and complete disorder [2], but this is not enough for a definition of a complex system. So the usual approach is to enumerate the key features of complex systems in general. Then, to find out if some system is complex or not, the presence of those features is checked, and if majority of them are present then the system is treated as a complex system.

The general features of any complex system are:

- The system has internal structure.
- The system has behavior that is not observed in it's constituent parts.
- System adapts to inputs and evolves.
- There is uncertainty in the system.

In other words, complex systems have structure which consists of interconnected, possibly heterogeneous, components. The interconnections between components could be asymmetric and could change over the time. These interconnections allow complex systems to exhibit behavior that couldn't be observed in it's constituent parts. Note that neither full connectivity between nodes, nor absence of any connectivity, makes system complex. It is important that that the connectivity lies somewhere in between those two extreme cases.

System's adaptation and evolution allows them to be flexible, autonomous and robust, which are features not usually found in non-complex systems. For example, every human being is capable of adapting to it's environment while robotic systems, that are not so complex and try to mimic human behavior, are not so adaptable. This shows that adaptability directly impacts the complexity.

Finally, uncertainty of a system comes from it's

Examples of complex systems are easy to find [4][5]. They can be natural, in a sense that they were made by some natural process like reproduction, mutation and selection, or man made. Examples of natural complex systems are metabolic processes within cells, brain, animals and groups in which they are living, human being. Examples of man made complex systems are power grid, traffic systems, supply chains, World Wide Web, Internet.

Measure of the complexity of some system is a very useful quantity. Namely, if we are making a model of some system and that model is improving with time then we could expect that the measure of complexity will increase. It is also possible that we somehow estimated complexity of a real system and then the difference between estimated value and a value given by the model will approach zero as model improves. The exact behavior and use depends on the measure itself. Unfortunately, it is not easy to find a measure of complexity even though there are dozens of proposals [6]. Algorithmic, or Kolmogorov, complexity measure is the most frequently mentioned but is measure of randomness more than complexity. Two modifications of Kolmogovor complexity are iteresting, *logical depth* and *algorithmic statistics*. Both of those modifications try to distinguish between random data and complexity. All the algorithmic approaches suffer from the same drawback, that it is hard to compute them. Apart from those algorithmic approaches there are also statistical, thermodynamical, etc. But in general no measure for the time being is satisfactory.

B. Risk assessment

Risk assessment is only a part of the whole security process whose purpose is to protect information resources of some organization. There are multiple possible approaches to this process, but in this paper we'll base our argumentation on NIST's Risk Management Guide [7]. That guide is quite popular and the other methods do not deviate radically from the process described in NIST's document.

In the given reference term *risk* is defined as the *function* of the *likelihood* of a given *threat-source*'s exercising particular potential *vulnerability*, and the resulting *impact* of that adverse event on the organization.

To determine this function's value, i.e. the risk, nine step process is defined of which, for our purpose, the first 7 are important:

1) System Characterization.
2) Threat Identification.
3) Vulnerability Identification.
4) Control Analysis.
5) Likelihood Determination.
6) Impact Analysis.
7) Risk Determination.

In system characterization step the goal is to identify components, features and system boundaries of the information system under the risk management process. The output of the threat identification step is a list of potential threat sources which are grouped into natural, human and environmental groups.[1] For each threat source an estimate of motivation, resources and capabilities should also be given. The vulnerability identification's step purpose is to identify weaknesses of the system identified in the first step. Weaknesses are not only software bugs, but can also be lack or inadequate policies and system procedures, misconfiguration, etc. There is a multitude of sources from where vulnerabilities can be gathered, e.g. vendors security lists, public forums, automated vulnerability scanning, pen testing, etc. In control analysis step all the existing (or planned) controls are examined. The goal is to be aware of all the available controls. In likelihood determination step for each vulnerability a probability that it is exercised by some threat source is evaluated. The suggested scale consists of three values: *high*, *medium*, and *low*. Also, it is necessary to determine the magnitude of the impact for each vulnerability if it is exercised successfully by some threat source. Finally, in the risk determination phase the output from all the previous steps is gathered and risk is determined for each combination of vulnerability and threat source. This allows risks to be rated and resources to be directed at the most important/highest risks.

There are several problems with this approach. *The first*, and the most important, is that vulnerabilities of resources are treated independently of each other. This means, for example, that risk of integrity violation of one component might be determined to be of a low importance, but this violation could have cascading effects which ultimately could bring to a compromise some much more important IT resource for which this attack vector hasn't been analyzed at all. *The second* one is that the outcome greatly depends on the knowledge and experience of a person, or persons, conducting risk assessment. Furthermore, the outcome also depends on the knowledge and experience of the interviewed persons. This makes the whole process highly subjective. *The third*, IT systems are in some way living organisms that constantly change and adopt and this means that risk assessment procedure is too slow to follow the changes. Then, as *the fourth problem*, it is hard to automate this procedure as it strongly depends on humans intervention. It could be done, but only to a small degree. Finally, as *the fifth problem* we'll note that is also interesting question of the quality of a particular risk assessment done in some organization, i.e. what is the measure how good it is, and did it improve over the time.

All these problems are, as we shell see in the next two sections, solvable if we treat IT system, and it's risk assessment, as a complex systems and if we don't try to reduce it to it's constituting elements.

III. Complexity of risk assessment

We'll start our analysis of how risk assessment should be done by arguing that information system, for which risk assessment is done, is a complex system. This, in turn, means that risk assessment is done for a complex system and that any method that prescribes how risk assessment should be done has to take this fact into account.

For a start, information system consists of resources. *Resource* is a generic term that encompasses anything that has any connection with information that should be protected. Resource can be people that use or maintain information system, computers that run applications or store data, applications, networking resources that transfer data, different more traditional information storage entities (papers, manuals, different cabinets and work tables). Resources interact in different and very complex ways in order to support *business processes* that in turn are the key component of any company. So, we have first element of a complex system, that is, that *the information system has internal structure*.

Next characteristic of complex systems is that the system has behavior that is not observed in it's constituent parts. This, so called *emergent* behavior, is not easy to find. It would be tempting to declare that business process that is run by the information system is emergent behavior, but almost any business process could be run on a single computer. Thus, it isn't likely candidate for this property. But, there is a case in which security breach could happen that can not be reduced to a single component, and that is a multistage attack. These attacks depend on different security breaches of multiple components in order for a target to be compromised. So, this could be treated as a emergent behavior.

The third property of a complex system is that it constantly adapts to inputs and evolves. This is certainly true as business requirements constantly change and in turn this requires information system to evolve and adapt to a new requirements.

Finally, as a fourth property, there should exist certain uncertainty in the system. This property is also easy to show. Namely, as anyone who deals with security knows, there is no absolute security. In other words, there

[1] We'll note here that malicious software is placed within human threat source group, but we think that this kind of a software is more or less autonomous and should be treated separately.

is always certain probability that security incident will happen.

Thus, it turns out that information system, and more specifically it's security state, has features of a complex system which leads us to the conclusion that it is a complex system and has to be treated as a such.

IV. Complexity based risk management method

The method proposed in this paper is based on the following premises:

- Information system is a complex system in which interconnections between it's components play an important role and thus have to be taken into account.
- Risk assessment itself is a complex procedure that additionally has to be done in as close as possible to a real time. Thus, method has to be designed with this fact in mind, i.e. it has to be easily to automate.
- Model of the information system used to make a risk assessment will hardly ever be the exact replica of a real system and thus there will always be an error. But, it has to strive to be one, and also, improvements in a model could be measured by measuring how complex the model is.
- Values of resources have to be measured independently of any person's subjective opinion. In other words, we try to make everything objective and repeatable in a sense that if two different persons perform security risk assessment of the same information system, they should produce almost the same results under the assumption they both used the same underlying model.

The general idea behind risk management method suggested in this paper is the following:

- Enumerate all the resources and build connections between them.
- Add vulnerabilities, threats and controls for all resources.
- Analyze where the security risks are by analyzing how threats from threat sources can spread through the system.
- Add controls to lower the highest risks (or accept them, depending on the management's decision).
- Improve model by adding more resources, connections and controls and more details.

In the following subsections we discuss elements of this risk management method.

A. Resources

The central concept of the risk assessment method proposed in this paper is a *resource*. Resource is anything, material or non-material that takes part in the information system. To compile a list of resources the following sources are minimally expected to be used:

- *Accounting books*. Accounting books hold information about all the material and some non-material assets that the company owns. Also in the books the value of resources is recorded which is annually amortized. So, this is an important input to compose a list of resources.
- *Human resources database*. Employees are also resources that have to be taken into account when composing resource list. Since all the employees in the company are managed by the human resources department or equivalent this is very important source of information.
- *Asset management software*. This source will overlap with the accounting books. But not all resources necessary for security risk assessment will be present in account books. For example, instance of some database will be in the asset management software but not in the account lists, while the database application itself will be in accounting books because licence had to be payed for it.
- *Document management system*. This is not directly usable for the risk management software as it is the case with the previous items but it is important source of management and operational controls that affects the outcome of risk assessment.

As we said in the previous section, resources are highly interconnected, that is there is dependency between them. For example, application that is running on some computer is dependent on that computer, which in turn is dependent on the network and power supply. Dependency of two resources is unidirectional link in which resource A depends in some way on resource B. The following relationships exist:

- *IS IN*. This dependency describes physical placement of one resource (e.g. computer, room) within some other resource (e.g. building).
- *ATTACHED TO*. This is dependency in a sense of attachment to computer network or power line. More generally, this dependency allows transfer of energy or information via wires or using wireless methods.
- *COMMUNICATES WITH*. This is communication on application level, i.e. more abstract (non-

material) than 'attached to' relationship.

- *CONNECTED TO*. This is a physical connectivity. For example, rooms are connected to each other or to hallways.
- *MAINTAINS*. Maintenance of buildings or computing and networking resources. This relationship assumes elevated privilege of one resource that performs maintenance on another resource that is maintained.
- *STORED ON*. Storage of application images or data on some medium.
- *EXECUTES ON*. Application or OS uses computer/CPU.
- *USED BY*. Similar to maintains relationship but without elevated privileges.

Dependencies between resources allow threat sources to spread and to reach other resources. For example, if some threat reached resource B, which in turn depends on resource A, then threat source can also use some vulnerability of resource B to compromise it as well.

To take into account failover situations we introduce the term *metaresource*. These are the resources that don't exist per se, but actually hide, or group, real resources. For example, there is sometimes power generator within some building in case main power supply fails. So, for the requirements of risk assessment, secondary generator and electrical substation are bound together into single power supply.

To get *the value of the resource*, and impact in case it is compromised, we use two complementary methods that give resource's final value. The first one is the accounting value. But, for non-material resources (e.g. instances of a certain application) the material value isn't taken into the account. The reason is simple, it is relatively easy to make a copy of installation media and a loss of a single instance isn't by itself a significant problem. On the other hand, for material resources like a personal computer accounting value is important as the loss of the computer translates into a material loss.

The second method to get resource's value is by adding all the values of resources depending on it. For example, if there is some data stored on a computer that the value of this data has to be added to the computer's own value and this is it's final value that we manipulate with in the risk assessment process.

Additionally, as a measure of a resource's value a time needed to replace it can also be taken into account.

To make management of threats and vulnerabilities easier, all the resources are categorized and each category has predefined a certain set of threats and vulnerabilities. The idea is that when a user adds new resource it automatically also binds to it threats and potential vulnerabilities.

B. Threats, Vulnerabilities and Controls

We treat each resource as a source of a threat and in the same time each resource has a set of vulnerabilities. For example any computing device with a power supply can cause fire, and fire is a threat, either intentional or, in this case, unintentional. To make system consistent, and self-contained, we include some resources that otherwise wouldn't be included. Some of those resources are also classified as a metaresource, introduced in the previous subsection. For example, Internet is a source of attackers and thus we include it in the list of resources, but on the other hand we don't evaluate risk level for it. Another example is a metaresource *nature* that is a source of natural disasters like flood, fire, and similar.

In any case, when new resource is added, the most specific category is searched for and resource is assigned to it. This in turn means that the resource is automatically assigned a set of threats and vulnerabilities. For example, if we add Windows installation, then the most specific category would be Windows operating system, that is by itself subcategory of Operating systems group. This would add all the threats, vulnerabilities and controls to this particular instance.

Controls are mechanisms that prevent threats from spreading among the resources. For example, locked doors prevent people from accessing certain rooms and armored door make this protection even stronger. Still, to access server it isn't physical access that is only problematic to the security of a server but also network access and there are different controls that manage that part, e.g. packet filters on a firewall, passwords, etc.

Controls themselves in a model of information system for a risk assessment can be placed between resources, or on a certain resource. In any case, controls are by default cumulative even though there are cases when they are mutually exclusive. As an example of mutually exclusive controls we can take the scenario in which authentication could be done using either passwords or smart cards. In such cases, OR combination has to be done of all the controls.

Finally, the complexity based risk assessment and management process allows a possibility of a new control, lowering the complexity of a system. For example, if it turns out that there are some connections that make the system too dependent on a certain resource, connections could be removed in order to lower the risk or changed to some other resource.

C. Risk assessment

Based on the model built as described in the previous subsections risk assessment is done. The risk assessment consists of evaluating a ways and probabilities of threat sources getting to each component of information system. The assessment exact way how is this done is still unknown because there certainly will be combinatorial explosion of possibilities.

Now we can state that there is a crucial difference between risk assessment done this way and the more traditional way as described in the Section II. In a traditional security risk assessment each resource is treated separately and in this separation the threats has to be evaluated. The result will certainly be strongly dependent on the person doing security assessment. In this model, the risk assessment process has better justifications and also, could be automated.

D. Improving the model of information system

The model of information system constructed for a risk assessment will always be simplified and approximate version of a real information system. This is inevitable as the real system is a live and constantly changing. Furthermore, sometimes it will be necessary to do simplifications. For example, when computer is added to the system in the model it will be modeled as a computer (hardware), OS running on top of it and application of top of the OS. As the model of information system matures, this simplified version will have to be replaced with a more fine grained version in order to better cope with certain vulnerabilities. Example of vulnerabilities that are not well included into simplified model are memory failures. If memory fails, it isn't the whole hardware that failed and it is a less risky than if hard disk failed. So, gradually, everything will have to be more and more detailed to better capture the essence and to give better risk assessment. This in turn means that the complexity of the model will grow and this growth, and value, could be used as a measure of the quality of the whole risk assessment process.

The model can also be improved, and checked against the real situation, by using automated tools like network topology scanners, intrusion detection systems, central logging systems and similar.

V. Conclusions and Future Work

In this paper we argued that the information system is a complex system, and by extension, that risk management process is itself a complex process because it deals with complex system. This has impact on the existing approaches since it basically says that we can not reduce this system to basic components and treat them separately as this approach can hugely differ from the real situation. This also means that the importance of some components could be missed, while of the others overstated.

We also presented a novel risk management process that takes into account complexity and doesn't try to simplify things when they are not simple. This risk management process has a potential of a high degree of automation which makes it more objective and also, very important, fast, allowing it to work in a real time.

Still, much work has to be done in order for this method to become useful in practice. For example, the exact interaction between resource dependency and controls has to be determined. Then, how threats spread through the system should also be more carefully evaluated.

Acknowledgment

This work has been carried out within project 036-0361994-1995 Universal Middleware Platform for Intelligent e-Learning Systems funded by the Ministry of Science and Technology of the Republic Croatia.

References

[1] M. Mitchell, "Complex systems: Network thinking," *Artificial Intelligence*, vol. 170, no. 18, pp. 1194 – 1212, 2006, special Review Issue. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYF-4M936PV-1/2/884217ac%2eb3d328e41e2e0d96dbcfa5

[2] F. Heylighen, "Complexity and self-organization," in *Encyclopedia of Library and Information Sciences*, M. J. Bates and M. N. Maack, Eds. Taylor & Francis, 2008.

[3] J. Guckenheimer and J. M. Ottino, Eds., *Foundations for Complex Systems Research in the Physical Sciences and Engineering*. NSF, September 2008, Report from an NSF Workshop.

[4] S. N. Dorogovtsev and J. F. F. Mendes, "Evolution of networks," September 2001. [Online]. Available: http://arxiv.org/abs/condmat/0106144

[5] M. E. J. Newman, "The structure and function of complex networks," March 2003. [Online]. Available: http://arxiv.org/abs/cond-mat/0303516

[6] C. R. Shalizi, "Methods and techniques of complex systems science: An overview," in *Complex Systems Science in Biomedicine*, ser. Topics in Biomedical Engineering International Book Series, E. Micheli-Tzanakou, T. S. Deisboeck, and J. Y. Kresh, Eds. Springer US, 2006, pp. 33–114, 10.1007/978-0-387-33532-2_2. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-33532-2_2

[7] G. Stonebumer, A. Goguen, and A. Feringa, *NIST Special Publication 800-30: Risk Management Guide for Information Technology Systems*, National Institute of Standards, U.S. Department of Commerce, July 2002. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf