Computer Standards & Interfaces 38 (2015) 44–50

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Scalable risk assessment method for cloud computing using game theory (CCRAM)

Evrim Furuncu, Ibrahim Sogukpinar *

Gebze Institute of Technology Computer Engineering Department, P.K. 141, Gebze, 41400 Kocaeli, Turkey

ARTICLE INFO

Article history:
Received 13 January 2013
Received in revised form 9 November 2013
Accepted 16 August 2014
Available online 27 August 2014

Keywords:
Cloud computing security
Scalable security using cloud service models
Game theory security modeling

ABSTRACT

Cloud computing is one of the most popular information processing concepts of today's IT world. The security of the cloud computing is complicated because each service model uses different infrastructure elements. Current security risk assessment models generally cannot be applied to cloud computing systems that change their states very rapidly. In this work, a scalable security risk assessment model has been proposed for cloud computing as a solution of this problem using game theory. Using this method, we can evaluate whether the risk in the system should be fixed by cloud provider or tenant of the system.

1. Introduction

Cloud computing has been increasingly used in recent years by organizations to deliver new services, enter new markets, get closer to customers and decrease IT operation costs. Generally, cloud computing is defined as usage of another computer's resources as a service that is delivered using a network. Technological advances in broadband connections made it possible to use for normal users of the Internet for cloud computing.

Since more than one entity uses these computer resources, its security becomes more important than normal IT resources that are used by one entity. By the definition of the National Institute of Standards and Technology (NIST), typically there are three different service models presented as follows for cloud computing [1].

- Software as a Service (SaaS): Software delivery model using cloud infrastructure. Since there is no need to install anything extra, users can access to this service from anywhere where they have Internet. Some examples are mail services, office applications, Customer Relationship Management (CRM) and collaboration, etc.
- Platform as a Service (PaaS): In this service model, tenant gets a platform where he/she can develop and run their application on. Cloud provider provides complementary services and required technological infrastructure to develop and run the application. Google AppEngine, Force.com and Microsoft Azure are known PaaS providers.
- Infrastructure as a Service (IaaS): In IaaS, cloud vendors provide the infrastructure to the tenant in the form of computing power or storage. Infrastructure comes from the data centers which are used virtualization to divide and distribute its resources. Rackspace Cloud, Google Computing Engine and Amazon EC2 are some examples for IaaS service model.

In each service model, different layers are needed to execute the service stack. Since each service model requires different computing resources, security measures which are used for each of these service models may be varied. Some security measures in some service models must be implemented by the cloud provider. However, the other security implementations are not necessarily needed to be done by cloud provider; instead, they must be implemented by the tenants. These security precautions can be different depending on Service-level Agreement (SLA) which is a negotiated agreement between tenant and the cloud provider.

Security requirements for the service models that are defined by NIST [1] are given in Table 1. However, a point to be made here is that cloud computing does not consist of only three models. Apart from the NIST defined models SaaS, PaaS and IaaS, there are other models currently used by providers such as:

- Storage as a Service (STaaS or SaaS): In this model, the service provider rents space in its infrastructure to another party or individual.
- Desktop as a Service (DaaS): Delivers a "virtualized" desktop to the user; thus, all the programs, applications, processes and data are kept on centralized server.
- Network as a Service (NaaS): This model includes application accelerating, security measures or mobile device management, etc.
- Data a Service (DaaS): Providing data on demand to the tenant

* Corresponding author. Tel.: +90 262 605 2201; fax: +90 605 2205.
E-mail addresses: efuruncu@bilmuh.gyte.edu.tr (E. Furuncu), ispinar@bilmuh.gyte.edu.tr (I. Sogukpinar).

http://dx.doi.org/10.1016/j.csi.2014.08.007
0920-5489/

Table 1
Security requirements for cloud computing.

IT security requirements (X requirement, * optional requirement)	Cloud deployment models					
	Public			Private		
	Cloud service models					
	IaaS	PaaS	SaaS	IaaS	PaaS	SaaS
Availability	X	X	*	X	X	X
Authorization	X	X	X	*	*	X
Confidentiality	*	*	X	*	X	X
Integrity	X	*	X	*	X	X
Authentication	X	*	X	X	*	X

regardless of geographic or organizational separation of the provider or tenant.

Security measures for these service models are different than each other because of the requirements for different resources. For example, availability requirement for NaaS is more important than the other requirements. Because, it is the elementary need for that service to provide bandwidth and the network. The cloud provider is not responsible for ensuring confidentiality and integrity of the passing data. But, since all computation is done by the provider in SaaS model, all the security properties presented in Table 1 must be implemented by the SaaS provider.

Network attackers are generally known as intelligent and rational human beings. They consider the cost and profit of their attacks. Defenders profit when a harmful attack is blocked by their security systems. But, if such an attack doesn't happen, they can lose money because of the unnecessary security measures. These properties make it possible to model this behavior in game theory [14,15]. Like in network security, there is an important game connection between attackers and defenders in the cloud computing. Ideal defensive strategy and ideal offensive strategy may be changed depending on each other.

Game theory techniques are used in economy, biology, mathematics, psychology and other social and behavioral sciences. In computer science, many works used game theory have been realized on intrusion detection systems (IDS) [2], security scheduling [3] and network/cyber security [4, 15–20]. In the recent years, studies between game theory, economic theory and computer science have given way to a new field, Algorithmic Game Theory [5].

In this work, a model has been proposed to determine defensive and offensive ideal strategies using properties of defender and attackers that are mentioned above and considered the security measures taken. Strategies increasing gain or reducing damage are presented to the corresponding players using this model. As a result, cloud computing security staff can determine which security measures should be taken depending on their gain or loss. The proposed model is a novel solution for security of cloud computing. Evaluated security risks using proposed method in the cloud computing system should be mitigated by a cloud provider or a tenant of the system.

The rest of the paper is organized as follows: Section 2 includes related works. Section 3 introduces the proposed model. Section 4 gives a brief practical example and discussion. Section 5 presents our conclusions and future work.

2. Related works

Since risk assessment in cloud computing is a hot topic, most of researches in this topic are built on grid computing infrastructure. Research that is based on grid computing generally do not cover the storage of data which is an important aspect of cloud computing because most grids are used to solve a single task and do not cover storage of the data, and also most of these research focuses on static risk assessments. In ref [6], the FPVA model applies mostly to grid middleware which is used to separate the work load of a program into more physical machines. Although it seems similar to a cloud, because users create platform or change the platform in IaaS and PaaS service models, FPVA model becomes ineffective for such targets. Basically, this model first creates a tree that consists of the interactions between applications and assets. After, each node in this tree is analyzed for relationship with each other and examined for the possible security vulnerabilities. Next, each node's programming code is manually inspected by experts considering the possible security vulnerabilities that tree relation presented. One of the biggest problems of this model is that manually inspecting the cloud APIs would take so much time considering the magnitude of IaaS and PaaS APIs. Another problem is that even though cloud and grid technologies seem similar, the risk associated with them is different with each other.

Peiyu and Dong proposed a three layered risk assessment model summarized as follows using AHP in ref. [7].

- Level one: Formulates the problem in a hierarchical structure. The overall objective is placed on the top of the level. In this model, it corresponds to the overall assessment of the cloud computing system platform.
- Level two: Includes eight attributes consisting of major factors identified for assessing level one.
- Level three: The last level is for concrete assessment factors in the decision framework. Thirty nine factors were identified corresponding to higher levels and specific local conditions.

Implementation of the AHP requires three principles: decomposition, pair-wise comparisons and synthesis of weights. Some advantages using AHP process in cloud computing are:

- Able to break problem into heretical pieces,
- Able to quantify the decision-marker's experimental judgments, particularly when the objectives lacked quantifiable data.

Decision makers evaluate the assessment considering the factors defined in level three and each one is given a weight and put together in a matrix to get a weight factor. The problem in here, decision makers have to manually change the vectors until assessments pass verification of consistency.

J. Oriol Fitò and Jordi Guitart [8] proposed a semi-quantitative approach to risk assessment for cloud computing. Decisions are made by considering business level objects such as; maximizing profit and user satisfaction. In this semi-quantitative approach, some risks even turn into gain for the business. Impact of the risks changes between two factors: benefit and threat. Before comparison, risks are grouped considering the following factors:

- The probability of occurrence of a risk event: Takes values between 1 and 5. Expressed by means of very unlikely (1, e.g. once in 20 years), unlikely (2, e.g. yearly), possible (3, e.g. monthly or weekly), likely (4, e.g. daily), and frequent (5, e.g. at any moment).
- The impact of that event: Either a threat, a benefit, or both, semi-quantified between very high (−5 or 5, for negative and positive impact, respectively), high (−4/4), medium (−3/3), low (−2/2) and very low (−1/1).
- The risk-level estimation: This is proportional to the probability of a given event and its business level object in question.

In this method, probability of the risk is multiplied with the impact of the risk to get risk level estimation which is between the values of −25 and +25 [8]. Considering the impact of the risks (benefit and threat), this is a narrow interval. In our method, considering the service model of the provider and the value of the system, our interval size is larger. Also, the proposed model does not take risks as a benefit in any way in order to eliminate any problems that may occur in the future.

M. Kiran et al. [9] explained in their paper that most proposed risk assessments in cloud computing considers heavily on user side of

things. Both in their model and the method proposed here try to look in the perspective of the cloud provider. Even though our models look into the same thing, their paper separates cloud providers into two categories: Service provider and infrastructure provider. Service providers are defined as the middle institution that borrows infrastructure from infrastructure providers and use that base to build a service that users can work on. In this way, service providers and infrastructure providers can make risk assessment by giving different weights to certain factors. This assessment continues by separating risks into different categories. Next, elimination or mitigation of risks can be realized by considering these categories.

3. Cloud computing risk assessment method using game theory (CCRAM)

In IT security, an attack is defined as the interaction between the attacker and the defense system that is used to protect the target. In this situation, the defense system and the attacker are active players that their outcomes change depending on their interaction with each other. Game theory is an interdisciplinary approach to inspect the behavior between two players or a group using its own attributes. Game theory deals with finding optimal strategies, which increases outcome or decreases damage, in response to another group's or entity's behavior. To choose the optimal strategy, some of assumptions presented as follows are needed to be made.

- Each player is able to take two or more well-specified actions or series of actions.
- Every action that players take leads to a well-defined end state, even if the game seems continuous.
- Each player is associated with a specified payoff in each end state.
- Each player, which makes decisions, knows the rules of the game and has the knowledge of other player's payoffs.
- Given two actions, it is assumed that players will strive to maximize their payoffs (assumption of rationality and maximization).

The reason for game theory modeling is to explain why humans behave in a specific way and to guess what action they will take according to their behaviors.

In the field of security, game theory can be used to help for choosing optimal strategies for defending and attacking according to the probability of actions that defender or the attacker will take [14,15]. Considering the game's criteria, a utility function is defined for each player by the game theory. These utility functions are used to show the outcomes of the actions made by players. If a player chooses strategy A that maximizes his outcome depending on the other person's or group's strategy B, strategy A is called best response. If this is applicable to all the players, in other words, if no player has an incentive to deviate from his/her decision after considering opponent's choice, these profiles of strategies are called Nash Equilibrium.

One of the important parts of the security model is to calculate the risks factors in risk assessments. Therefore, following assumptions are needed to be made:

- Attacker and the defender are the players of the game. If there is more than one attacker, they are assumed that they cooperate with each other. If there is more than one defender, they also are assumed to cooperate with each other.
- Every attacker's skills are considered as equal. So, even on typical attacks, attackers considered having the same attack possibility and each threat poses an important risk on the provider.
- Every attack is considered as successful unless there is a security measure for it.
- If the defender takes a security measure for that type of attack, every attack in that type is considered to be detected and can be stopped.

CCRAM is an imperfect information non-cooperative non-zero static game model. Certainly, the attacker and the defender do not have a cooperative intent so the game is non-cooperative. The attacker's and the defender's movements have a cost. So, just using gains in the utility functions won't produce the outcome properly. Therefore, one side's loss does not always become other side's gain. So the game is nonzero. The main purpose of this model is to give the defender to choose a defense strategy in the event of an attack which affects either the provider's own systems or system's bought from another cloud provider.

Definition 1. ***G***:(***P***,***S***,***S***)

a) $P = (1,2,\ldots,n)$ represents the players named as attackers and defenders in the game. If there is more than one attacker or defender, they are considered to cooperate with other attackers or defenders respectively.
b) $S = (s_1,s_2,\ldots,s_n)$ represents the strategy space of the players and $\forall x \in P$, $Sx \neq \emptyset$, $S_x = (S_1^x, S_2^x, \ldots, S_m^x)$ is the strategy set of the player x.
c) $U = (U_1,U_2,\ldots,U_n)$ is the utility function of the players in the game.

Definition 2. CCRAM is an imperfect information non-cooperative non-zero static game model, which is defined as:

$$CCRAM = \{(a,d),(S_a,S_d),(U_a,U_d)\}$$

where, a represents the attacker and d represents the defender. The set $S_a = (S_1^d, S_2^d, \ldots, S_m^a)$ is the strategy space of the attacker and s_j^a is a strategy that can be used by the attacker. $S_d = (S_1^d, S_2^d, \ldots, S_m^d)$ is the strategy space of the defender and s_j^d is a strategy that can be used by the defender. U_a and U_d are the utility functions of the attacker and the defender respectively.

3.1. Attack and defense modeling

In the study of MIT Lincoln Laboratory, the impact on a host caused by an attack cannot be expressed with a parameter. In response to this, our method will use information securities core principles, namely, confidentiality, integrity and availability to measure the impact on the target system.

Confidentiality, Integrity and Availability (CIA) can be used to calculate the impact of an attack on a system. National Vulnerability Database [10] uses CIA as impact metrics in their CVSS v2 Vector Definitions. Attack classification is done by considering what the attacker want to do with that kind of attack. Considering the strategic objectives of the attack, a weigh is given each point in the CIA between 0 and 1 to differentiate with other type of attacks (Table 2).

Defensive methods are separated into three types presented in Table 3 according to the time needed to apply the security measure and impact that the security measure makes on the normal operation of the system.

3.2. Calculating the value of the system

The basis of the cloud computing is virtualization. Virtualization is creating a virtual version of hardware (CPU, RAM etc.), network resources (routers, switches etc.), operating systems or storage devices.

Table 2
Some examples for classes of attack.

Category	Definition	C	I	A
DoS	Denial of service	0	0	1
User	Gaining user privilege	0.5	0.1	0.05
Data	Non-permitted data access and write	1	1	0.2
Administrative	Gaining administrator user privilege	1	0.1	0.05
Scan	Getting information about the target system	0.3	0	0.2

Table 3
Some examples for defense classes.

Category	Definition	Properties
Basic	Change the configuration of a virtual or a physical asset	Time needed to change configuration is low and has no effect on the system.
Mid-level	Restarting machine, security policy change, changing the configuration using administrative level account	Time needed to apply changes is longer and requires professional knowledge; system may need to be stopped.
Advanced Level	Heavy I/O operations like scanning virus and backing up system, or changing system entirely i.e. upgrading the system	Time needed is longer than the other three; system generally needed to be stopped in order to do such operations and generally needed to be done by experts.

Virtualization is done basically by splitting the asset into a different size of assets to utilize them better. For example, traditionally a server's CPU generally does not utilize more than %20 if it is doing a single job. Thus, the %80 of it actually goes to waste. However, if the CPU is split into five and each piece is given a job, the CPU utilization would become close to %100.

If we inspect a system that is considered as a virtual machine running on a hypervisor, there are a couple of things that need to be considered before defining the value of that system. One of these things is live migration of the system. Live migration is transferring a resource from a physical location to another physical location without interrupting the service. This is done for a couple of reasons such as; load balancing, some other virtual asset on the system gets attacked or the physical system that hosts the virtual one has a problem, etc.

The proposed model requires the value of the system in CIA form. We can see that if the system in question live migrates to another physical location, its CIA values cannot stay the same. Each live migration changes both the physical and virtual system's CIA value.

To solve this problem, each host's confidentiality, integrity and availability values are stored in a database. When a new virtual machine is created, a CIA value will be given to it depending on the service level that the tenant requires and also the policies that the cloud provider have. Thus, when it is given in the cloud infrastructure, it has a CIA value given by the provider depending on the importance. When the system migrates, its CIA value will change depending on the other virtual systems on the physical location. By storing these values in a database, the computer's importance which is changing over time can be determined.

Considering the scale of cloud computing, it can be said that the size of database will increase too quickly. Round Robin Database Tool (RRD tool) can be used to solve this problem. What RRD tool does it as the data gets older? It takes the average of a set of points and reduces that average to a single point. Once the data is a year old, for example only weekly average is needed instead of daily average from the old data. Basically, seven points is reduced down to a single point. So, as the data gets older and older, it may get less accurate, but we can still retrieve enough information from it. And it most certainly is better than no data.

3.3. Comparison of risks

Our model calculates the impact of risk as a quantitative value, so, unlike other models, it can also be used to compare risks. This can be used to determine which risk is more important than others. It is done by using:

The value of the system in the hands of the provider as represented in the CIA form (V_C,V_I,V_A) multiplied by the impact of the risk on the host in CIA form (L_C,L_I,L_A). The total of the products is multiplied by the exploitability sub score (E_{fac}) which can be obtained from National Vulnerability Database [11].

$$\text{Impact of the risk on the system} = E_{fac} * ((L_C * V_C) + (L_I * V_I) + (L_A * V_A)) \quad (1)$$

Using these quantitative values, the attacks on that target system can be compared with each other. In addition, we can compare the importance of same risks between cloud service models (SaaS, PaaS etc.) with each other, for example, if risks are defined as v, w, x, y, z, based on the magnitude of the risks. SaaS could take more damage from x, y, z and PaaS could take more damage from v,z,w.

3.4. Calculating the payoff matrix

To calculate a player's payoff, cost and benefit should be considered. If U^a_{kl} is assumed to represent the utility function of the attacker, B^a_{kl} becomes the benefit of the attacker. The cost of attack is represented by C^a_{kl}. Symbols k and l show that the attacker chooses the k^{th} strategy in the attacker's strategy space; correspondingly, the defender chooses the l^{th} strategy in the defender's strategy space. Likewise, U^d_{kl} represents the utility function of the defender, B^d_{kl} represents the benefit of the attacker and C^d_{kl} represents the cost of security measures. And again symbols k and l show that the attacker chooses the k^{th} strategy in the attacker's strategy space; correspondingly, the defender chooses the l^{th} strategy in the defender's strategy space.

Definition 3. The player's utility functions are expressed as:

$$U = B - C \quad (2)$$

So, the attacker's utility function becomes

$$U^a_{kl} = B^a_{kl} - C^a_{kl} \quad (3)$$

The defender's utility function becomes.

$$U^d_{kl} = B^d_{kl} - C^d_{kl} \quad (4)$$

3.5. Benefit of defender

In the attack classification, we propose that an attack can damage the assets' CIA properties. If the level of the damage done to the asset is represented by L_C, L_I, L_A and the value of the asset in the eyes of the cloud provider is represented by V_C, V_I, V_A, then their multiplication shows the how much value does the asset lose in the occurrence of an attack.

If the asset is recovered by using a security measure, R is defined as the value recovered from the attack by using a security measure,

$$\text{Recovered Server Value} = (R_C * V_C) + (R_I * V_I) + (R_A * V_A)) \quad (5)$$

In Table 4, s^d_l represents a successful defensive strategy against the s^a_k attack strategy. $-\ s^d_l$ represents an unsuccessful security measure or no security measure for the s^a_k attack strategy. No attack situation is represented by $-\ s^a_k$.

Table 4
Benefit of the defender.

Strategy	Benefit
s^a_k, s^d_l	$B^d_{kl} = (R_c - L_c) * V_c + (R_I - L_I) * V_I + (R_A - L_A) * V_A$
s^a_k, $-\ s^d_l$	$B^d_{kl} = -((L_C * V_C) + (L_I * V_I) + (L_A * V_A))$
$-\ s^a_k$, s^d_l	$B^d_{kl} = 0$
$-\ s^a_k$, $-\ s^d_l$	$B^d_{kl} = 0$

Table 5
Cost of the defender.

Strategy	Cost
s_k^a, s_l^d	$C_{kl}^d = C_P + C_M + C_C$
$s_k^a, -s_l^d$	$C_{kl}^d = 0$
$-s_k^a, s_l^d$	$C_{kl}^d = C_P + C_M + C_C$
$-s_k^a, -s_l^d$	$C_{kl}^d = 0$

3.6. Cost of defender

The security measures that are taken by the defender will have an impact on the system. These costs change depending on the classes of defense given in Table 5. These costs are:

- Process costs (C_p): Include time taken and computational resources needed to take security precautions.
- Material cost (C_M): Includes direct material cost needed to take security measure or money.
- System continuity cost (C_C): This costs covers the damage to availability which happens when the system in question needed to be closed in order to apply the security measures.

3.7. Benefit of attacker

The benefit of the attacker is the loss of the defender's system. In other means, it is the loss of defender's asset's CIA value. But, not all the loss of defender can be gained by the attacker. To express this, a cofactor $_k \in [0,1]$ is defined to show how much of the loss is converted into the benefit of the attacker. The benefit formula becomes as $B^a = -k * B^D$ for the attacker (Table 6).

3.8. Cost of attacker

In order to execute the attack, the attacker needs to have some sort of equipment (C_E) or time (C_T). In normal situations, the sum of these is the cost of attacker. However, if a security measure implemented by the defender detects the attack, the attacker would be exposed depending on the attack type. The penalty of exposure is represented by C_{PE} (Table 7).

4. Experimental results and comparison

To demonstrate the proposed method, a case study is constructed around two different cloud providers that use different types of service models. It is considered the end use case of resources rented from the providers to be used in database related work

A SaaS provider is responsible for nearly all of the security requirements, because in a SaaS model both data and the computation are done on provider side. However, the IaaS provider is generally responsible for the availability of the resources not the security side of it. When a security breach happens in IaaS model, if it does not spread to other tenant's assets, it is the problem of the tenant whose assets are exploited. Considering these facts, the CIA values for the resources (or simply virtual machines) used in this experiment are given in Table 8.

Table 6
Benefit of the attacker.

Strategy	Benefit
s_k^a, s_l^d	$B_{kl}^a = k((R_c - L_c) * V_c + (R_I - L_I) * V_I + (R_A - L_A) * V_A)$
$s_k^a, -s_l^d$	$B_{kl}^a = k((L_C * V_C) + (L_I * V_I) + (L_A * V_A))$
$-s_k^a, s_l^d$	$B_{kl}^a = 0$
$-s_k^a, -s_l^d$	$B_{kl}^a = 0$

Table 7
Cost of the attacker.

Strategy	Benefit
s_k^a, s_l^d	$C_{kl}^a = C_E + C_T + C_{PE}$
$s_k^a, -s_l^d$	$C_{kl}^a = C_E + C_T$
$-s_k^a, s_l^d$	$C_{kl}^a = 0$
$-s_k^a, -s_l^d$	$C_{kl}^a = 0$

4.1. Risk comparison

Five risks are compared with each other in different cloud service models in order to see the importance level on different service models. These risks are actually real vulnerabilities that can affect a host in the cloud. For the sake of this experiment, CIA damage levels are taken from the example given in Table 2. Real vulnerabilities in daily life are given for each type; DoS (CVE-2012-1820), user (CVE-2012-2752), data (CVE-2012-4579), admin (CVE-2011-4005), and scan (CVE-2010-1638). The exploitability sub score is taken for each risk from the corresponding scoring pages. Results are shown in Table 9.

The order of importance of risks that can be seen in Table 9 for SaaS model is different from IaaS model. For SaaS model, the data and admin type of attacks are more serious than the others. The order for importance for SaaS is data, admin, scan, DoS and user. However, the order for importance of risks for IaaS is DoS, data, admin, user and scan.

4.2. The risk game

For this case, it is assumed that both providers have CVE-2012-0116 vulnerability in their systems. As a basis, this vulnerability is considered as a data attack presented in Table 2. Thus, its damage levels in CIA form become $L_C = 1, L_I = 1, L_A = 0, 2$. Some parameters are given in light of the experience $R_C = 1, R_I = 1$, and $R_a = 0, 4$.

We consider the security measure which can stop this attack to have costs of $C_P = 60$, $C_M = 70$ and $C_C = 50$. The attacker's costs becomes $C_E = 25$, $C_T = 5$ for this type of attack. If the attacker is exposed due to a security measure, he/she will pay as $C_{PE} = 50$ punishment cost.

After calculating the utility functions for both the attacker and the defender, result matrices are presented in Tables 10 and 11.

According to Nash equilibrium, every finite game has at least one equilibrium point. The solution matrix for IaaS shown in Table 10 can be solved using the best response. If the attacker chooses to execute the attack, the defender will choose no measures because it benefits more. If the defender chooses to take no security measures, executing the attack would benefit the attacker more. (Because "No Measure" is a dominant strategy for the Defender.) Since these strategies point the same cell (A,NM) in the matrix, it can be said that this is a pure strategy Nash equilibrium point.

In this type of games, there is either a pure Nash equilibrium or a mixed Nash equilibrium. By taking a close look at Table 11, we can say that SaaS solution matrix includes a pure strategy Nash equilibrium (NA,NM). But, in this case the attacker and the defender won't make anything. So, we will use the mixed strategy Nash equilibrium [12] to determine the result of game. Using the mixed strategy Nash equilibrium, Eqs. (6) and (7) can be defined.

$$(-168) * \mathbf{p_A} + (-180) * (1-\mathbf{p_A}) = (-224) * \mathbf{p_A} + (0) * (1-\mathbf{p_A}) \quad (6)$$

Table 8
Asset value of the servers in different service models.

	Confidentiality	Integrity	Availability
SaaS system	100	100	60
IaaS system	20	80	100

Table 9
Calculated effect of risks.

Attack type	Total score for SaaS	Total score for IaaS
DoS	330	550
User	245.7	89.7
Data	1441.6	326.4
Admin	971.8	283.8
Scan	420	80

Table 10
Solution matrix for IaaS model.

		Defender	
		Take measures (TM)	No measures (NM)
Attacker	Attack (A)	−100, −160	110, −140
	No action (NA)	0, −180	0, 0

$$(-92) * \mathbf{p_D} + (194) * (1-\mathbf{p_D}) = (0) * \mathbf{p_D} + (0) * (1-\mathbf{p_D}) \quad (7)$$

where $\mathbf{p_A}$: Attacker's Attack (A) probability and $\mathbf{p_D}$: Defender's Take Measure (TM) probability.

After solving Eqs. (6) and (7) we obtain $\mathbf{p_A} = 0{,}76$ and $\mathbf{p_D} = 0{,}68$. So, we can say that if the defender chooses the move "take measures" with a probability of 0.68, the attacker's outcome does not differentiate between attacking and taking no action. So, from the attacker's view point, attacking and taking no action have the same outcome. Likewise, if the attacker chooses to execute the attack with a probability of 0.76, the defender's outcome won't change whether if it takes the measures or not.

In other words, if the defender takes measure on %68 of the entire system, the attacker would deviate from attacking. Likewise, from the defender's view point, if let's say more than %76 of the entire systems are attacked, taking security measures will cost more than taking no action.

4.3. Annual loss expectancy

One of the most common methods to assess the risk is Annualized Loss Expectancy (ALE). ALE represents the expected asset value loss due to a risk over a year.

$$\text{ALE} = \text{SLExARO} \quad (8)$$

ALE is the multiplication of the Single Loss Expectancy (SLE) and Annualized Rate of Occurrence (ARO).

The system presented in Table 11 shows that taking security measures on the %68 of the system, the attacker's outcome does not change whether he/she attacks or not. We can use ALE to determine the loss of the unprotected %32 of the system. If the unprotected system's loss becomes bigger than the security measures' cost for that system, the rest of the system may as well be taken under protection.

4.4. Comparison with other methods

We can separate each given service type in our method. So, cloud services can be separated beyond the two types that are defined by M. Kiran et al. in ref [7]. As mentioned before, there are more models than the NIST defined ones such as; SaaS that is used for backing up data, NaaS that is used for renting network assets and DaaS which puts client's computing and data into the cloud and serve data to the client when the client needs. If the service provider defined in ref [7] is seen as a SaaS provider, the rest of the services is considered as an infrastructure provider forced differences between IaaS and STaaS to be ignored. In our method, we compare every service model, which can be damaged from such a risk, with each other in a scalable manner.

Table 11
Solution matrix for SaaS model.

		Defender	
		Take measures (TM)	No measures (NM)
Attacker	Attack (A)	−92, −168	194, −224
	No action (NA)	0, −180	0, 0

Generally, methods of risk assessment take a long time and processed slowly. Unlike other methods, our method creates a game out of the effects of the risks which we can solve using game theory. Also using the history of the provider, our method can determine which assets will be attacked. Our model ensures fast and provider specific calculations.

5. Conclusion and future works

Rapid adaptation of the cloud computing also brings security problems with it [13]. Today's risk assessment methods for cloud computing do not put forward the cost and benefit of attackers and defenders. Our work tries to solve the problem of deciding an ideal strategy to take security measures when using cloud computing. The proposed method uses game theory to model the outcome of the defender and the attacker. We calculate the defender's ideal strategy using the asset value in the eyes of the cloud provider and the risks that can happen to the asset.

Our model can be expanded by calculating different risks in the same time and putting these numbers into a solution matrix. Using this method, we can calculate different security measures that can mitigate the same type of risks. Also this way, if another security measure costs are less, we can choose it. Parameters that make up the utility functions also need some work to provide better answers. Especially, the history of the CIA values that are given by the cloud provider is crucial because it directly affects our model's effectiveness.

References

[1] P. Mell, T. Grance, The NIST definition of cloud computing, National Institute of Standards and Technology Special Publication 800–145, 2011.
[2] B. Wang, J. Cai, S. Zhang, J. Li, A network security assessment model based on attack–defense game theory, International Conference on Computer Application and System Modeling (ICCASM), 2010, (Taiyuan Shanxi, China).
[3] J. Pita, M. Jain, J. Marecki, Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport, 2008. (Estoril, Portugal).
[4] S. Shiva, S. Roy, D. Dasgupta, Game theory for cyber security, Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '10), ACM, 2010, (New York, NY, USA).
[5] R.J. Lipton, V.V. Vazirani, M. Mihail, C. Tovey, E. Vigoda, Algorithmic game theory, 2007.
[6] J.A. Kupsch, B.P. Miller, E. Heyman, E. César, First principles vulnerability assessment, Proceedings of the 2010 ACM workshop on Cloud computing security workshop CCSW '10, 2010, (New York, NY, USA).
[7] L. Peiyu, L. Dong, The new risk assessment model for information system in cloud computing environment, Procedia Eng. 15 (2011) 3200–3204.
[8] J.O. Fitó, J. Guitart, Business-driven management of infrastructure-level risks in Cloud providers, Futur. Gener. Comput. Syst. 32 (2014) 41–53.
[9] M. Kiran, M. Jiang, D.J. Armstrong, K. Djemame, Towards a service lifecycle based methodology for risk assessment in cloud computing, Dependable, Autonomic and Secure Computing (DASC), Dec 2011, pp. 449–456.
[10] CVSS v2 vector definitions, http://nvd.nist.gov/cvss.cfm?vectorinfo&version=2.
[11] National vulnerability database version 2.2, http://nvd.nist.gov/.
[12] M.J. Osborne, An introduction to game theory, Oxford University Press, USA, 2003.
[13] Cloud Security Alliance, Security guidance for critical areas of focus in cloud computing 3.0, https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf.
[14] T. Alpcan, T. Başar, Network security: a decision and game theoretic approach, Cambridge University Press, Jan. 2011.
[15] M. Tambe, Security and game theory: algorithms, deployed systems, lessons learned, Cambridge University Press, Dec. 2011.
[16] T. Spyridopoulos, G. Karanikas, T. Tryfonas, G. Oikonomou, A game theoretic defense framework against DoS/DDoS cyber attacks, Comput. Secur. 38 (2013) 39–50.
[17] M.H. Manshaei, Q. Zhu, T. Alpcan, T. Başar, J.-P. Hubaux, Game theory meets network security and privacy, ACM Comput. Surv. 45 (3) (2013) 25:1–25:39.

[18] D. Zissis, D. Lekkas, Addressing cloud computing security issues, Futur. Gener. Comput. Syst. 28 (2012) 583–592.
[19] C.Y.T. Ma, N.S.V. Rao, D.K.Y. Yau, A game theoretic study of attack and defense in cyber-physical systems, Infocom Workshops, 2011.
[20] G. Fan, H. Yu, L. Chen, D. Liu, A game theoretic method to model and evaluate attack–defense strategy in cloud computing, IEEE 10. Int. Conference on Services Computing (SCC), 2013, pp. 659–666.