International Journal of Cyber-Security and Digital Forensics (IJCSDF) 1(3): 265-279
The Society of Digital Information and Wireless Communications, Nov. 2012 (ISSN: 2305-0012)

Towards quantitative measures of Information Security: A Cloud Computing case study

Mouna Jouini[1], Anis Ben Aissa[2], Latifa Ben Arfa Rabai[3], Ali Mili[4]

Department of computer science
ISG
Tunis, Tunisia
[1]jouini.mouna@yahoo.fr
[3]latifa.rabai@isg.rnu.tn

Department of computer science
ENIT
Tunis, Tunisia
[2]anis_enit@yahoo.fr

College of Computing Sciences
New Jersey Institute of Technology
Newark NJ 07102-1982 USA
[4]mili@cis.njit.ed

ABSTRACT

Cloud computing is a prospering technology that most organizations consider as a cost effective strategy to manage Information Technology (IT). It delivers computing services as a public utility rather than a personal one. However, despite the significant benefits, these technologies present many challenges including less control and a lack of security. In this paper, we illustrate the use of a cyber security metrics to define an economic security model for cloud computing system. We, also, suggest two cyber security measures in order to better understand system threats and, thus, propose appropriate counter measure to mitigate them.

KEYWORDS

Cloud computing, cyber security metrics, mean failure cost, security requirements, security threats, threats classification.

1 INTRODUCTION

With the rapid development of processing and storage technologies and the emergence of the Internet, computing resources have become cheaper, more powerful and more ubiquitously available than ever before. As a consequence, IT service providers are faced to challenges of expanding the structures and infrastructures with small expenditure and short a time in order to provide rising demands from their customers. To address these business challenges, cloud computing architecture was developed. In this technology, end users avail themselves of computing resources and services as a public utility, rather than a privately run small scale computing facility. In the same way that we use electricity as a public utility (rather than build our own generators), and that we use water as a public utility (rather than dig our own well), and that we use phone service as a public utility (rather than build and operate our own cell tower), we want to use computing services as a public utility. Such a service would be available to individuals and organizations, large and small, and would operate on the same pattern as other public utilities, namely:

- Subscribers sign up for service from a service provider, on a contractual basis.
- The service provider delivers services of data processing, data access and data storage to subscribers.
- The service provider offers warranties on the quality of services delivered.

- Subscribers are charged according to the services they use.

It offers the usual advantages of public utilities, in terms of efficiency (higher usage rates of servers), economies of scale (time sharing of computing resources), capacity (virtually unlimited computing power, bounded only by provider assets rather than by individual user assets), convenience (no need for users to be computer-savvy, no need for tech support), dependability (provided by highly trained provider staff), service quality (virtually unlimited data storage capacity, protected against damage and loss) [1, 11, 12, 15, 16].

Like traditional computing environments, cloud computing brings risks like loss of security and loss of control [5, 7, 8, 13, 14, 18, 19]. Indeed, by trusting its critical data to a service provider, a user (whether it is an individual or an organization) takes risks with the availability, confidentiality and integrity of this data. In addition to that, the aim of Cloud computing is to deliver its applications and services to users through the internet and therefore it is prone to various kinds of external and internal security risks such as denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks that affect especially the subscriber' data.

In this paper, we propose two security metrics based on threats classification that enable service providers and service subscribers not only to quantify the risks that they incur as a result of prevailing security threats and system vulnerabilities but also to know the origin of threats. The reason why security is a much bigger concern in cloud computing than it is in other shared utility paradigms is that cloud computing involves a two-way relationship between the provider and the subscriber: whereas the water grid and the electric grid involve a one-way transfer from the provider to the subscriber, cloud computing involves two-way communication, including transferring information from subscribers to providers, which raises security concerns.

The security metrics we discuss in this paper quantifies in economic terms the loss resulted in security breaches, thereby enabling providers and subscribers to weight these risks against rewards to assess the cost effectiveness of security countermeasures, and then, to identify the source of threats to propose appropriate security solutions. This paper is organized as follows: In section 2, we discuss how to quantify security threats using some quantitative models. In section 3, we will use the Mean Failure Cost (MFC) as a cyber security measure. In section 4, we apply the MFC in a cloud computing system. In section 5, we proceed to threat classification to propose appropriate security countermeasure and we conclude by summarizing our results, focusing on strength of the cybersecurity measure and sketching directions of further research.

2 QUANTIFYING DEPENDABILITY AND SECURITY ATTRIBUTES

The most computer failures are due to malicious actions and they have increased during the last decade. Lord Kelvin stated "If you cannot measure it, you cannot improve it." In other words, security cannot be managed, if it cannot be measured. This clearly states the

importance of metrics to evaluate the ability of systems to withstand attacks, quantify the loss caused by security breach and assess the effectiveness of security solutions. Hence, there are quantitative models that estimate the dependability of a system which can be measured according to the reliability, availability, usability and security metrics such as the mean time to failure (MTTF), the mean time to discovery (MTTD) and the mean failure cost (MFC) [2, 14].

The mean time to failure (MTTF):

The mean time to failure (MTTF) describes the expected time that a system will operate before the first failure occurs. It is the number of total hours of service of all devices divided by the number of devices [21].

The mean time between failures (MTBF):

The Mean time between failures (MTBF) describes the expected time between two consecutive failures for a repairable system. It is the number of total hours of service of all devices divided by the number of failures [21].

The mean time to discovery (MTTD):

The mean time to discovery (MTTD) refers to the mean time between successive discoveries of unknown vulnerabilities [20].

The mean time to failure (MTTE):

The mean time to exploit (MTTE) refers to the mean time between successive exploitations of a known vulnerability [21].

Average Uptime Availability (or Mean Availability):

The mean availability is the proportion of time during a mission or time period that the system is available for use [20].

These models reflect the failure rate of the whole system, they ignore the variance stakes amongst different stakeholders, the variance in failure impact from one stakeholder to another. They also make no distinction between requirements. Besides, they consider that any failure to meet any requirement is a failure to meet the whole specification. To estimate the MTTF of a system, we only need to model its probability of failure with respect to its specification.
Consequently, the mean failure cost takes into account:

- The variance in failure cost from one requirement to another.
- The variance in failure probability from one component to another
- The variance in failure impact from one stakeholder to another.

The mean failure cost (MFC) presents many advantages:

- It provides a failure cost per unit of time (mean failure cost): it quantifies the cost in terms of financial loss per unit of operation time (e.g. $/h)
- It quantifies the impact of failures: it provides cost as a result of security attacks.
- It distinguishes between stakeholders: it provides cost for each system's stakeholder as a result of a security failure.

3 MFC A MEASURE OF CYBER SECURITY

Computing systems are characterized by five fundamental properties: functionality, usability, performance, cost, and dependability. Dependability of a computing system is the ability to deliver service that can justifiably be trusted.

A systematic exposition of the concepts of dependability consists of three parts: the threats to, the attributes of, and the means by which dependability is attained.

Despite the existence of quantitative metrics that estimate the attributes of dependability like the Mean Time To Failure MTTF for reliability and the Mean Time To Exploitation MTTE (a measure of the security vulnerability), there is no way to measure directly the dependability of the system or to quantify security risks.

3.1 The Mean Failure Cost (MFC)

In [3], Ben Aissa et al introduce the concept of Mean Failure cost as a measure of dependability in general, and a measure of cyber security in particular.

3.1.1 The Stakes Matrix

We consider a system S and we let H_1, H_2, H_3, … H_k, be stakeholders of the system, i.e. parties that have a stake in its operation. We let R_1, R_2, R_3, … R_n, be security requirements that we wish to impose on the system, and we let $ST_{i,j}$, for $1 \le i \le k$ and $1 \le j \le n$ be the stake that stakeholder Hi has in meeting security requirement R_j. We let PR_j, for $1 \le j \le n$, be the probability that the system fails to meet security requirement R_j, and we let MFC_i (Mean Failure Cost), for $1 \le i \le k$, be the random variable that represents the cost to stakeholder Hi that may result from a security failure.

We quantify this random variable in terms of financial loss per unit of operation time (e.g. $/hour); it represents the loss of service that the stakeholder may experience as a result of a security failure. Under some assumptions of statistical independence, we find that the Mean Failure Cost for stakeholder Hi can be written as:

$$MFC_i = \sum_{1 \le j \le n} ST_{i,j} \times PR_j. \qquad (4)$$

If we let MFC be the column-vector of size k that represents mean failure costs, let ST be the k×n matrix that represents stakes, and let PR be the column-vector of size n that represents probabilities of failing security requirements, then this can be written using the matrix product (◦):

$$MFC = ST \circ PR \qquad (5)$$

The Stakes matrix is filled, row by row, by the corresponding stakeholders. As for PR, we discuss below how to generate it.

3.1.2 The Dependency Matrix

We consider the architecture of system S, and let C_1, C_2, C_3,… C_h, be the components of system S. Whether a particular security requirement is met or not may conceivably depend on which component of the system architecture is operational. If we assume that no more than one component of the architecture may fail at any time, and define the following events:

- Ei, $1 \le i \le h$, is the event: the operation of component Ci is affected due to a security breakdown.

- Em+1: No component is affected.

Given a set of complementary events E_1, E_2, E_3, … E_h, Eh+1, we know that the probability of an event F can be written in terms of conditional probabilities as:

$$P(F) = \sum_{k=1}^{h+1} P(F \mid E_k) \times P(E_k). \quad (6)$$

We instantiate this formula with F being the event: the system fails with respect to some security requirement. To this effect, we let F_j denote the event that the system fails with respect to requirement R_j and we write (given that the probability of failure with respect to R_j is denoted by PR_j):

$$PR_j = \sum_{k=1}^{m+1} P(F_j \mid E_k) \times P(E_k). \quad (7)$$

If

- we introduce the DP (Dependency) matrix, which has n rows and h+1 columns, and where the entry at row j and column k is the probability that the system fails with respect to security requirement j given that component k has failed (or, for k=h+1, that no component has failed),
- we introduce vector PE of size h+1, such that PE_k is the probability of event E_k, then we can write

$$PR = DP \circ PE \quad (8)$$

Matrix DP can be derived by the system's architect, in light of the role that each component of the architecture plays to achieve each security goal. As for deriving vector PE, we discuss this matter in the next section.

3.1.3 The Impact Matrix

Components of the architecture may fails to operate properly as a result of security breakdowns brought about by malicious activity. In order to continue the analysis, we must specify the catalog of threats that we are dealing with, in the same way that analysts of a system's reliability define a fault model. To this effect, we catalog the set of security threats that we are facing, and we let T1, T2, T3, … Tp, represent the event that a cataloged threat has materialized, and we let Tp+1, be the event that no threat has materialized. Also, we let PT be the vector of size p+1 such that

- PTq, for $1 \leq q \leq p$, is the probability that threat Tq has materialized during a unitary period of operation (say, 1 hour).
- PTp+1 is the probability that no threat has materialized during a unitary period of operation time.

Then, by virtue of the probabilistic identity cited above, we can write:

$$PE_k = \sum_{q=1}^{p+1} P(E_k \mid T_q) \times PT_q. \quad (9)$$

If

- we introduce the IM (Impact) matrix, which has h+1 rows and p+1 columns, and where the entry at row k and column q is the probability that component Ck fails given that threat q has materialized (or, for q=p+1, that no threat has materialized),
- we introduce vector PT of size p+1, such that PTq is the probability of event Tq, then we can write

$$PE = IM \circ PT \quad (10)$$

Matrix IM can be derived by analyzing which threats affect which components, and assessing the likelihood of success of each threat, in light of perpetrator behavior and possible countermeasures. Vector PT can be derived from known perpetrator behavior, perpetrator models, known system vulnerabilities, etc. We refer to this vector as the *Threat*

Configuration Vector or simply as *the Threat Vector*.

3.1.4 Summary

Given the stakes matrix *ST*, the Dependency matrix *DP*, the impact matrix *IM* and the threat vector *PT*, we can derive the vector of mean failure costs (one entry per stakeholder) by the following formula:

$$MFC = ST \circ DP \circ IM \circ PT \quad (11)$$

where matrix *ST* is derived collectively by the stakeholders, matrix *DP* is derived by the systems architect, matrix *IM* is derived by the security analyst from architectural information, and vector *PT* is derived by the security analyst from perpetrator models. Figure 1 below illustrates these matrices and their attributes (size, content, indexing, etc).

4 ILLUSTRATION: CLOUD COMPUTING SYSTEM

We illustrate the use of our cyber security metrics on a practical application, namely a Cloud Computing system. We derive in turn the three matrixes of interest and the threat vector. To this effect, we identify the security requirements, the stakeholders and their stakes in meeting these requirements and the architectural components of this system.

4.1 The stake matrix

As for security requirements, we consider the security concerns that are most often cited in connection with cloud computing [7, 14, 16], namely: availability, integrity, and confidentiality. We further refine this classification by considering different levels of criticality of the data to which these requirements apply:

- Availability: it refers to the subscriber's ability to retrieve his/ her information when he/she needs it. Un-availability may be more or less costly depending on how critical the data is to the timely operation of the subscriber. Thus, we distinct two types:
  - Critical Data
  - Archival Data
- Integrity: it refers to the assurances offered to subscribers that their data is not lost or damaged as a result of malicious or inadvertent activity. Violations of integrity may be more or less costly depending on how critical the data is to the secure operation of the subscriber. Accordingly, we distinct two types:
  - Critical Data
  - Archival Data
- Confidentiality: it refers to the assurances offered by subscribers that their data is protected from unauthorized access. Violations of confidentiality may be more or less costly depending on how confidential the divulged data. The data can be classified into:
  - Highly Classified Data
  - Proprietary Data
  - Public Data

For the purposes of our model, we then assume that we are dealing with seven generic security requirements, namely:

- AVC: Availability of Critical Data.
- AVA: Availability of Archival Data.
- INC: Integrity of Critical Data.
- INA: Integrity of Archival Data.

- CC: Confidentiality of Classified Data.
- CP: Confidentiality of Proprietary Data.
- CB: Confidentiality of Public Data.

We assume that the provider makes different provisions for these requirements, putting more emphasis on critical requirements than on less critical requirements. We further assume, for the sake of argument, that for each requirement, the provider makes the same provisions for all its subscribers; hence if the provider fails to meet a particular requirement, that failure applies to all the subscribers that are dependent on it.

We consider three classes of stakeholders in a cloud computing situation, namely: the service provider, the corporate or organizational subscribers, and the individual subscribers. For the sake of illustration, we consider a fictitious running example, where we have a cloud computing provider (PR), and a sample of three subscribers:

- A corporate subscriber (CS),
- A governmental subscriber (GS),
- An individual subscriber (IS).

Table 1: Stakes Matrix: cost of failing a security requirement stakes in $K/h

	Requirements						
	AVC	***AVA***	***INC***	***INA***	***CC***	***CP***	***CB***
Stakeholders							
PR	500	90	800	150	1500	1200	120
CS	150	40	220	80	250	180	60
GS	60	20	120	50	2500	30	12
IS	0,05	0,015	0,30	0,20	0,30	0,10	0,01

Based on a quantification of these stakes in terms of thousands of dollars ($K) per hours of operation, we produce the following stakes matrix as shown in Table 1.

4.2 The Dependency matrix

In the cloud computing system, we focus on two parts: the front end and the back end connecting to each other through the Internet. The front end is the side of the computer user or client including the client's computer and the application required to access to the cloud computing system. The back end is the "cloud" section of the system which are the various physical/virtual computers, servers, software and data storage systems that create the "cloud" of computing services. The most common approach [6, 9] defines cloud computing services as three layers of services:

- Software as a Service (*SaaS*) offers finished applications that end users can access through a thin client like Gmail, Google Docs and Salesforce.com
- Platform as a Service (PaaS) offers an operating system as well as suites of programming languages and software development tools that customers can use to develop their own applications like Microsoft Windows Azure and Google App Engine.
- Infrastructure as a Service (IaaS) offers end users direct access to processing, storage and other computing resources and allows them to configure those resources and run operating systems and software on them as they see fit like Amazon Elastic Compute Cloud (EC2) and IBM Blue cloud.

Table 2: Dependency Matrix

Dependency Matrix	Components									
	Browser	*Proxy server*	*Router/ Firewall*	*Load balancer*	*Web server*	*Appl server*	*Database server*	*Backup server*	*Storage server*	*No failure*
Security Requirements										
AVC	1	1	1	1	0,44	0,28	1	0,01	1	0
AVA	1	1	1	1	0,44	0,28	0,28	0,01	1	0
INC	0,14	0,14	1	1	0,44	0,14	1	0,01	1	0
INA	0,14	0,14	1	1	0,44	0,14	0,14	0,01	1	0
CC	0,44	0,14	1	1	0,44	0,44	0,44	0,01	0,44	0
CP	0,44	0,14	1	1	0,44	0,44	0,44	0,01	0,44	0
CB	0,44	0,14	1	1	0,44	0,44	0,44	0,01	0,44	0

The cloud computing paradigm optimizes in costs of physical resources (servers, CPUs, memories…) by the virtualization techniques. This lets users put numerous applications and functions on a PC or server, instead of having to run them on separate machines as in the past. The cloud computing architecture contains three layers [9, 10]:

- Core foundational capabilities: it includes a browser, a proxy server and a router/Firewall and load balancer.
- Cloud services: it includes a web server, an application server, a database server, a backup server and a storage server.
- User tools.

Assuming no more than one component fails at a time, and considering the additional event that no component has failed, the dependency matrix has (9 + 1 = 10) columns and 7 rows (one for each security requirement), for a total system, described in [4], to fill the dependency matrix as we do in table 2.

4.3 The impact matrix

The following step in our model is to deriver the impact matrix ie, the derivation of the set of threats that we wish to consider in our system. As we mentioned above, Cloud Computing is based on virtualization technology, but this later causes major security risks and thus, this system is threatened by many

Table 3: Impact Matrix

	Threats														
	MVH	*CVH*	*VMm*	*VMS*	*MVV*	*VMC*	*VMM*	*DoS*	*FA*	*DL*	*MI*	*ASTH*	*ANU*	*IAI*	*NoT*
Components															
Brws	0	0	0	0	0	0	0	0,02	0,01	0	0,03	0,02	0	0,03	0
Prox	0,01	0,05	0	0,01	0,01	0,05	0,05	0,02	0,01	0	0,005	0,02	0,01	0	0
R/FW	0,03	0,05	0,033	0,03	0,03	0,05	0,05	0,06	0,04	0	0,005	0,02	0,01	0,01	0
LB	0,02	0,003	0	0,01	0,02	0,003	0,003	0,06	0,04	0	0,005	0,02	0,01	0,01	0
WS	0,03	0,003	0,033	0	0,03	0,003	0,003	0,02	0,04	0	0,01	0,02	0,01	0,01	0
AS	0,02	0,003	0,033	0,06	0,02	0,003	0,003	0,036	0,04	0	0,05	0,02	0,01	0,07	0
DBS	0,001	0	0,033	0,04	0,001	0	0	0,036	0,04	0,05	0,03	0,02	0,01	0,06	0
BS	0,001	0	0	0,04	0,001	0	0	0,036	0,04	0,05	0,03	0,02	0,01	0,06	0
SS	0,04	0,05	0	0,04	0,04	0,05	0,05	0,036	0,04	0,05	0,03	0,02	0,01	0,06	0
NoF	0,06	0,04	0,03	0,03	0,06	0,04	0,04	0,01	0,02	0,01	0,02	0,05	0,06	0,005	1

types of attacks which can be classified into three categories [6, 8, 14, 17, 18]:

- Security threats originating from the host (hypervisor): This class includes Monitoring Virtual Machines from host, Virtual machine modification and Threats on communications between virtual machines and host.
- Placement of malicious VM images on physical systems: it includes Security Threats Originating Between the Customer and the Datacenter attack, Flooding attacks, Denial of service (DoS) attack, Data loss or leakage, Malicious insiders, Account, service and traffic hijacking and Abuse and nefarious use of cloud computing.
- Insecure application programming interfaces: it includes Security threats originating from the virtual machines, Monitoring VMs from other VMs, Virtual machine mobility and Threats on communications between virtual machines

In this section we have catalogued fourteen distinct types of threats. To compute the MFC we need to know the probability of the attack for each threat during one hour. Also we need to fill the values in that table 4 (150 entries), it comes from our empirical study [4] which has an immense source of references.

Table 4: Threat Vector

Threats	**Probability**
Monitoring virtual machines from host (MVM)	$8{,}063\ 10^{-4}$
Communications between virtual machines and host (CBVH)	$8{,}063\ 10^{-4}$
Virtual Machine modification (VMm)	$8{,}063\ 10^{-4}$
Placement of malicious VM images on physical systems (VMS)	$8{,}063\ 10^{-4}$
Monitoring VMs from other VM (VMM)	$40{,}31\ 10^{-4}$
Communication between VMs (VMC)	$40{,}31\ 10^{-4}$
Virtual machine mobility (VMM)	$40{,}31\ 10^{-4}$
Denial of service (DoS)	$14{,}39\ 10^{-4}$
Flooding attacks (FA)	$56{,}44\ 10^{-4}$
Data loss or leakage (DL)	$5{,}75\ 10^{-4}$
Malicious insiders (MI)	$6{,}623\ 10^{-4}$
Account, service and traffic hijacking (ASTH)	$17{,}277\ 10^{-4}$
Abuse and nefarious use of cloud computing (ANU)	$17{,}277\ 10^{-4}$
Insecure application programming interfaces (IAI)	$29{,}026\ 10^{-4}$
No Threats (NoT)	0,9682

Using the 3 Matrix (Stakes, Dependency and Impact) and the threat vector we can compute the vector of mean failure costs for each stakeholder of Cloud Computing system using the formula:

$$MFC = ST \circ DP \circ IM \circ PT \qquad (11)$$

Table 5: Stakeholder Mean Failure Cost

Stakeholders	**MFC($K/h)**
PR	15,20443
CS	3,53839
GS	8,98502
IS	0,00341

From Table 5 above we can see that the cost of failure for each stakeholder is so high.

To avoid the high cost of failure and reduce risks, we start by identifying vulnerabilities which help to understand how an attacker might exploit these vulnerable points. The attacker provides an efficient countermeasure to mitigate these vulnerabilities at their earliest stages before they become more harmful. He also, analyzes their effects on activities or stakeholders goals. Hence critical vulnerabilities in cloud computing system have been identified.

However, these vulnerabilities are dispersed among two intrusion spaces: internal and / or external, which we allow to identify and then apply appropriate countermeasures.

5 MFC MODEL EXTENSION

We illustrate, in this section, an extension of our MFC model by suggesting a classification of the identified threats to propose two types of measures: the Internal MFC and the External MFC in order to know the source of threats shaped the CC system to develop appropriate strategies to prevent, or mitigate their effects.

5.1 Classification methods

Threat assessment is an essential component of an information security risk evaluation. In order to identify vulnerabilities and to fix mitigation techniques, it is important to well understand potential threat sources or classes.

In threats classification, threats are presented together with the appropriate security services and a recommended solution [24]. The main aim of threats classification is to contribute to the understanding of the nature of threats by grouping it into classes depending in many criteria like source, modifying factors, resources, consequences [24]. In fact, identifying and classifying threats helps in the assessment of their impacts and the development of strategies to prevent, or mitigate the effects of threats to the system.
Threat classification is a planned activity for identifying and assessing system threats and vulnerabilities and then defining countermeasures to prevent, or mitigate the effects of threats to the system.

A threat is the adversary's goal, or what an adversary might try to do to a system [25]. It is also described as the capability of an adversary to attack a system [25]. Thus, a threat may be defined by two ways: techniques that attackers use to exploit the vulnerabilities in applications or impact or effect of threats to your assets. Therefore, there are some threat classification methods that are based on the first definition and others based on the second one.

For the threat classification methods that are based on the effect of threats, we cite:

- In [25, 26], Microsoft developed a method, called as STRIDE, for classifying computer security

threats. It is a classification scheme for characterizing known threats according to the goals and purposes of the attacks (or motivation of the attacker). The STRIDE acronym is formed from the first letter of each of the following classes: Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service and Elevation of privilege.

For the threat classification methods that are based on the techniques of threats, we cite:

- In [27], Visveswarn Chidambaram proposed a review of information system threats and then organized into three classes: network threats, server or host threats, and application threats.
- In [28], Lukas Ruf et al. proposed the three orthogonal dimensional model: that classify the threat space into subspaces according to a model of three orthogonal dimensions labeled Motivation (accidental, deliberate), Localization (external, internal) and Agent (human, technological, force majeure) to alleviates the risk assessment.
- In [23], Fariborz Farahmand et al. considered threats to a network system from two points of view: Threat agent, and penetration technique. In fact, a threat is caused by a threat agent using a specific penetration technique to produce an undesired effect on the network. An agent may be an unauthorized user, an authorized user and an environmental factor and threat techniques are classified into physical, personnel, hardware, software, and procedural.
- In [24], Karen Loch et al. proposed the four-dimensional model for information system security that classifies threats by source (internal, external) and perpetrator (human, non human), intent of the actions of the perpetrator, irrespective of the source (accidental or intentional) and consequences of threat on resources (disclosure, destruction, modification, denial of use).
- In [22], Antoon Rufi proposed a model to classify networks security threat. The model contains four main classes: unstructured threats, structured threats, external threats and internal threats.
- In [31], Kishor Trivedi et al. proposed a model that classifies threats into four classes: faults or attacks (physical faults, software bugs), errors (overload, misconfiguration), failures (physical attacks, software based attacks, man in the middle, jamming) and accidents. It is an extension of Laprie [32] taxonomy who classifies it into

three types: faults, errors, failures.

For the purpose of our system, we propose to classify the threat space into subspaces according to a model of three dimensions labeled Internal, External and Internal/External. This classification allows to localize the origin (or source) of a threat. In fact threat is either caused from within an organization, system or/ and architecture or from an external point of origin.

5.1.1 Internal threats

Internal threats occur when someone has authorized access to the network with either an account on a server or physical access to the network. A threat can be internal to the organization as the result of employee action or failure of an organization process.

Regarding internal attacks, McNamara lists, in [29], the following insider threats: theft of proprietary information, accidental or non-malicious breaches, sabotage, fraud, and eavesdropping/snooping.

5.1.2 External threats

External threats can arise from individuals or organizations working outside of a company. They do not have authorized access to the computer systems or network. They work their way into a network mainly from the Internet or dialup access servers. The most obvious external threats to computer systems and the resident data are natural disasters: hurricanes, fires, floods and earthquakes. External attacks occur through connected networks (wired and wireless), physical intrusion, or a partner network.

Table 6 : Probability of space intrusion

Threats	**Probability outsider committed**	**Probability insider committed**
(MVM)	1	0
(CBVH)	1	0
(VMm)	0.6	0.4
(VMS)	1	0
(VMM)	0.5	0.5
(VMC)	0.5	0.5
(VMM)	0.6	0.4
(DoS)	0.136	0.864
(FA)	1	0
(DL)	0.8	0.2
(MI)	0	1
(ASTH)	1	0
(ANU)	0	1
(IAI)	0.8	0.2

Lacey et al. provide, in [30], an updated profile of sophisticated outside attacks which can compromise the security of Mobile Ad-hoc Network (MANET). They include eavesdropping, routing table overflow, routing cache poisoning, routing maintenance, data forwarding, wormhole, sinkhole, byzantine, selfish nodes, external denial of service, internal denial of service, spoofing, Sybil, badmouthing, viruses, and flattering.

5.1.3 Internal/ external threats

Internal/ external threats take place when someone has authorized access to the network (for example an employee of the organization) causes external threats to the system.

5.2 MFC Computing

Using empirical data from [3] we can decompose the probability of event threat committed in two complementary probabilities (outsider/insider system committed) as showing in table 6.

5.3 Results and discussion

The MFC formula can be extended in two significant results:

- Mean failure cost of extern threats

$$MFC_{ext} = ST \circ DP \circ IM \circ PT_{ext} \quad (12)$$

Stakeholders	MFC_{ext}($K/h)
PR	10,61051
CS	2,46562
GS	6,278502
IS	0,002382

- Mean failure cost of intern threats

$$MFC_{int} = ST \circ DP \circ IM \circ PT_{int} \quad (13)$$

Stakeholders	MFC_{int}($K/h)
PR	4,5932
CS	1,07261
GS	2,7060
IS	0,001035

Computing the new values of the MFC extensions can give us the critical space of intrusion. The extensions of the MFC are more helpful for the countermeasures in our case we can adapt some solutions like adding more firewalls, proxy servers and antivirus servers.

6 CONCLUSION

Cloud computing is an emerging computing paradigm that provides an efficient, scalable, and cost-effective way for today's organizations to deliver consumer IT services over the Internet. A variety of different cloud computing models are available, providing both solid support for core business functions and the flexibility to deliver new services. However, the flexibility of cloud computing services has created a number of security concerns. In fact, it does not offer is absolute security of subscriber data with respect to data integrity, confidentiality, and availability.

In this paper we have illustrated the use of the MFC model on a practical application, namely a cloud computing system. This quantitative model enables cloud service providers and cloud subscribers to quantify the risks they take with the security of their assets and to make security related decisions on the basis of quantitative analysis.

We envision in previous work to refine the generic architecture of cloud computing systems, and use cloud-specific empirical data to refine the estimation of the dependency matrix and the impact matrix.

7 REFERENCES

1. Armbrust, M., Fox, A., Griffith, R., D. Joseph, A., Katz, R.: Above the Clouds: A Berkeley View of Cloud Computing. Technical report EECS-2009-28, UC Berkeley, (2009).
2. Barry W, Johnson.: Design and analysis of fault-tolorant digital systems. Barry W. Johnson, Addison-Wesley Longman Publishing Co., INC. Boston, MA, USA, (1989).
3. Ben Aissa, A., Abercrombie, RK., Sheldon, FT., Mili, A.: Quantifying security threats and their potential impacts: a case study. in Innovation in Systems and Software

Engineering: A NASA Journal, 6:269--281, (2010).
4. Ben Aissa, A.: Vers une mesure économétrique de la sécurité des systèmes informatiques. Doctoral dissertation, Faculty of Sciences of Tunis, submitted, Spring (2012).
5. Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., Molina, J.: Controlling data in the cloud: Outsourcing computation without outsourcing control. The 2009 ACM Workshop on Cloud Computing Security, Chicago, Illinois, USA, (2009).
6. Cloud Security Alliance.: Top Threats to Cloud Computing V 1.0. (2010), https://cloudsecurityalliance.org/topthreats
7. Hanna, S.: Cloud Computing: Finding the silver lining. (2009).
8. Ibrahim, A. S., Hamlyn-Harris, J., Grundy, J.: Emerging Security challenges of cloud virtual infrastructure. the Asia Pacific Software Engineering Conference 2010 Cloud Workshop, (2010).
9. Jinesh, Varia.: Cloud Architectures. Technology Evangelist Amazon Web Services, (2008).
10. Jaio, Orea. et al.: VisioTCI Reference Architecture (v2.12). Cloud Security alliance, (2011).
11. Mell, P., Grance, T.: Effectively and Securely Using the Cloud Computing Paradigm. In ACM Cloud Computing Security Workshop, (2009).
12. Mell, P. Grance, T.: The nist definition of cloud computing. Communications of the ACM. 53(6), 50--50, (2010).
13. Sean, C. Kevin, C.: Cloud Computing Security. International Journal of Ambient Computing and Intelligence, 3(1), 14--19, (2011).
14. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, (2010).
15. Vaquero, L M., Rodero-Merino, L., Caceres, J., Lindner, M.: A Break in the Clouds: Towards a Cloud Definition. ACM SIGCOMM Computer Communication Review, 39(1), 50--55, (2009).
16. Wang, L., von Laszewski, G., Kunze, M., Tao, J.: Cloud computing: A Perspective study. Grid Computing Environments (GCE) workshop, (2008).
17. Wayne, J., Timothy, G.: Guidelines on Security and Privacy in Public Cloud Computing. Information Technology Laboratory, (2011).
18. Wooley, P: Identifying Cloud Computing Security Risks. University of Oregon, Master's Degree Program, (2011).
19. Xuan, Z., Nattapong, W., Hao, L., Xuejie Z.: Information Security Risk Management Framework for the Cloud Computing Environments. 10th IEEE International Conference on Computer and Information Technology (CIT 2010), (2010).
20. The Center for Internet Security (CIS).: The CIS Security Metrics v1.0.0. (2009)
21. Speaks, S.: Reliability and MTBF Overview. Vicor Reliability Engineering, (2010).
22. Rufi, A.: Vulnerabilities, Threats, and Attacks, Rufi, A.: Network Security 1 and 2 Companion Guide (Cisco Networking Academy). Cisco Press, (2006).
23. Farahmand, F., Navathe, S., Sharp, G., Enslow, P.: A Management Perspective on Risk of Security Threats to Information Systems. Information Technology and Management archive, 6: 203--225 (2005).
24. Loch, K., Carr, H., Warkentin, M.: Threats to Information Systems: Today's Reality, Yesterday's Understanding. Management Information Systems Quarterly 16(2), 173--186.
25. Swiderski, F., Snyder, W.: Threat Modeling. Microsoft Press, (2004).
26. Meier, J., Mackman, A., Vasireddy, S., Dunner, M., Escamilla, R., Murukan, A.: Improving web application security: threats and counter measures. Satyam Computer Services, Microsoft Corporation, (2003)
27. Chidambaram, V.: Threat modeling in enterprise architecture integration. 2004, http://www.infosys.com/services/systemintegration/ThreatModelingin.pdf
28. Ruf, L., Thorn, A., Christen, T., Gruber, B.: Threat Modeling in Security Architecture - The Nature of Threats. ISSS Working Group on Security Architectures
29. McNamara, R., Networks—Where does the real threat lie?. Information Security Technical Report, 3(4), 65--74 (1998).
30. Lacey, T.H., Mills, R.F., Mullins, B.E., Raines, R.A., Oxley, M.E., Rogers, S.K.: RIPsec – Using reputation-based multilayer security to protect MANETs. Computers and Security, 31(1), 122--136 (2011).
31. Trivedi, K., Kim, D., Roy, A., Medhi, D.: Dependability and security models. International Workshop of Design of

Reliable Communication Networks (DRCN), IEEE, 11--20 (2009).
32. Avizienis, A., Laprie, J., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable and Secure Computing, 1(1), (2004).