An Economic model of security threats for cloud computing systems

Latifa Ben Arfa Rabai, Mouna Jouini, Marwa Nafati
Department of computer science
ISG
Tunis, Tunisia
latifa.rabai@isg.rnu.tn, jouini.mouna@yahoo.fr, marwanaff@yahoo.fr

Anis Ben Aissa
Department of computer science
ENIT
Tunis, Tunisia
anis_enit@yahoo.fr

Ali Mili
College of Computing Sciences
New Jersey Institute of Technology
Newark NJ 07102-1982 USA
mili@cis.njit.edu

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, propose a solution related to the vulnerabilities in cloud computing in order to reduce the probability that the components fail.

Keywords—cloud computing, cyber security, mean failure cost, security requirements, security threats.

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.

This modus operandi 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, 16, 18].

Like traditional computing environments, cloud computing brings risks like loss of security and loss of control [5, 7, 8, 13, 15, 20, 21]. 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: availability may be affected if the subscriber’s data is unavailable when needed, due for example, to a denial of service attack or merely to a loss; confidentiality may be affected if subscriber data is inadvertently or maliciously accessed by an unauthorized user, or otherwise unduly exposed; integrity may be affected if subscriber data is inadvertently or maliciously damaged or destroyed.

In this paper, we propose a security metric that enables service providers and service subscribers to quantify the risks that they incur as a result of prevailing security threats and system vulnerabilities. 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. Note that telephone service also involves the transfer of (vocal) information from subscribers to providers, and it
raises security concerns too (possibility of wiretapping),
though on a smaller scale.

The security metric we propose in this paper is
quantified in economic terms, thereby enabling providers
and subscribers to weight these risks against rewards, and
to assess the cost effectiveness of security
countermeasures. This paper is organized as follows: In
section 2, we discuss how to quantify security threats using
some quantifying models. In section 3, we adopt the Mean
Failure Cost (MFC) as a measure of cyber security. In
section 4, we illustrate the use of MFC in a cloud
computing system. In section 5, we debate the
vulnerabilities in cloud computing 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, 22].

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 [2].

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 [2].

The mean time to discovery (MTTD):
The mean time to discovery (MTTD) refers to the mean
time between successive discoveries of unknown
vulnerabilities [22].

The mean time to exploit (MTTE):
The mean time to exploit (MTTE) refers to the mean time
between successive exploitations of a known vulnerability
[22].

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
[22].

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 [23].

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 [23].

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 H₁, H₂, H₃, … Hₖ,
be stakeholders of the system, i.e. parties that have a stake
in its operation. We let \( R_1, \ R_2, \ R_3, \ldots \ R_n \) be security requirements that we wish to impose on the system, and we let \( ST_{i,j} \), for \( 1 \leq i \leq k \) and \( 1 \leq j \leq n \) be the stake that stakeholder \( H_i \) has in meeting security requirement \( R_j \). We let \( PR_j \), for \( 1 \leq j \leq n \), be the probability that the system fails to meet security requirement \( R_j \), and we let \( MFC_i \) (Mean Failure Cost), for \( 1 \leq i \leq k \), be the random variable that represents the cost to stakeholder \( H_i \) 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 \( H_i \) can be written as:

\[
MFC_i = \sum_{1 \leq j \leq n} ST_{i,j} \times PR_j.
\]

If we let \( MFC \) be the column-vector of size \( k \) that represents mean failure costs, let \( ST \) be the \( kn \) 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 (+):

\[
MFC = ST \circ PR.
\]

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 \( C_1, \ C_2, \ C_3, \ldots \ C_h \) 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:

- \( E_i \), \( 1 \leq i \leq h \), is the event: the operation of component \( C_i \) is affected due to a security breakdown.
- \( E_m+1 \): No component is affected.

Given a set of complementary events \( E_1, \ E_2, \ E_3, \ldots \ E_h, \ E_{h+1} \), we know that the probability of an event \( F \) can be written in terms of conditional probabilities as:

\[
P(F) = \sum_{k=1}^{h+1} P(F \mid E_k) \times P(E_k).
\]

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 \( PR_j \)):

\[
PR_j = \sum_{k=1}^{h+1} P(F_j \mid E_k) \times P(E_k).
\]

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 \( R_j \) given that component \( C_k \) 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

\[
PR = DP \circ PE.
\]

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 fail 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 \( T_1, \ T_2, \ T_3, \ldots \ T_p \) represent the event that a cataloged threat has materialized, and we let \( T_{p+1} \), be the event that no threat has materialized. Also, we let \( PT \) be the vector of size \( p+1 \) such that

- \( PT_q \), for \( 1 \leq q \leq p \), is the probability that threat \( T_q \) has materialized during a unitary period of operation (say, 1 hour).
- \( PT_{p+1} \) 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:

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

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 \( C_k \) 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 \( PT_q \) is the probability of event \( T_q \), then we can write

\[
PE = IM \circ PT.
\]

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:

\[ MFC = ST \odot DP \odot IM \odot PT, \]

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 matrices 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.

1. The stake matrix

As for security requirements, we consider the security concerns that are most often cited in connection with cloud computing [7, 15, 20], 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).

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.

<table>
<thead>
<tr>
<th>Stakeholders</th>
<th>AVC</th>
<th>AVA</th>
<th>INC</th>
<th>INA</th>
<th>CC</th>
<th>CP</th>
<th>CB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR</td>
<td>500</td>
<td>90</td>
<td>800</td>
<td>150</td>
<td>1500</td>
<td>1200</td>
<td>120</td>
</tr>
<tr>
<td>CS</td>
<td>150</td>
<td>40</td>
<td>220</td>
<td>80</td>
<td>250</td>
<td>180</td>
<td>60</td>
</tr>
<tr>
<td>GS</td>
<td>60</td>
<td>20</td>
<td>120</td>
<td>50</td>
<td>2500</td>
<td>30</td>
<td>12</td>
</tr>
<tr>
<td>IS</td>
<td>0.05</td>
<td>0.015</td>
<td>0.30</td>
<td>0.20</td>
<td>0.30</td>
<td>0.10</td>
<td>0.01</td>
</tr>
</tbody>
</table>
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.

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

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 types of attacks which can be classified into three categories [6, 8, 15, 19, 20]:

- 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. For the values in those table

<table>
<thead>
<tr>
<th>Dependency Matrix</th>
<th>Components</th>
<th>Browser</th>
<th>Proxy server</th>
<th>Router/Firewall</th>
<th>Load balancer</th>
<th>Web server</th>
<th>Appl server</th>
<th>Database server</th>
<th>Backup server</th>
<th>Storage server</th>
<th>No failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVC</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.44</td>
<td>0.28</td>
<td>1</td>
<td>0.01</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>AVA</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.44</td>
<td>0.28</td>
<td>0.28</td>
<td>0.01</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>INC</td>
<td>0.14</td>
<td>0.14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.44</td>
<td>0.14</td>
<td>1</td>
<td>0.01</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>INA</td>
<td>0.14</td>
<td>0.14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.44</td>
<td>0.14</td>
<td>0.14</td>
<td>0.01</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>CC</td>
<td>0.44</td>
<td>0.44</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.44</td>
<td>0.44</td>
<td>0.44</td>
<td>0.01</td>
<td>0.44</td>
<td>0</td>
</tr>
<tr>
<td>CP</td>
<td>0.44</td>
<td>0.44</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.44</td>
<td>0.44</td>
<td>0.44</td>
<td>0.01</td>
<td>0.44</td>
<td>0</td>
</tr>
<tr>
<td>CB</td>
<td>0.44</td>
<td>0.44</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.44</td>
<td>0.44</td>
<td>0.44</td>
<td>0.01</td>
<td>0.44</td>
<td>0</td>
</tr>
</tbody>
</table>
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:

\[ MFC = ST \times DP \times IM \times PT \]

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.

### 5. Vulnerabilities in cloud computing

There are several significant vulnerabilities that should be considered when an organization is ready to move their critical applications and data to a cloud computing environment. We can link the vulnerabilities in cloud computing to three types of security:

- **VM Security**: Related to the virtual infrastructure vulnerabilities.
- **Data Security**: Related to data storage vulnerabilities
- **Software Security**: Application vulnerabilities.

In this paper we will focus on VM security. To ensure a good VM security, the organization can adopt one of the following services:

- **Adding vulnerability scanning service (VMS)****
VMS is provided in two distinct types of scanning which can be employed together or separately:
- External: scanners can be used to scan your public facing IP addresses and Web applications and are designed to provide vulnerability detection of security risk exposures open to the Internet.
- Internal: allows you to assess the state of security vulnerabilities within your enterprise network.

- **Renting double virtual machines (RDVM)**

Renting double virtual machines is creating duplication in the cloud computing system, and as service, is putting the same components and configuration of the system in distinct physical machines. This solution reduces the probability that the components fail to the half. In our case we compute the MFC using (stakes, dependency and impact) matrix and the threat vector. If we adopt this solution each probability in impact matrix will be as follow:

\[ PE_k = \sum_{q=1}^{n} \frac{1}{2} \times P(E_k \mid T_q) \times PT_q \]

<table>
<thead>
<tr>
<th>Stakeholders</th>
<th>MFC($K/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR</td>
<td>7,6022</td>
</tr>
<tr>
<td>CS</td>
<td>1,7691</td>
</tr>
<tr>
<td>GS</td>
<td>4,4925</td>
</tr>
<tr>
<td>IS</td>
<td>0,001705</td>
</tr>
</tbody>
</table>

The new values of MFC (Table 6) appear more interesting but we can’t decide if this solution is better than other’s without computing the return on investments.

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 an 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 face with the security of their assets and to make security related decisions on the basis of quantitative analysis.

We envision extending our current work by refining 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.

REFERENCES