Security Quantification of Complex Attacks in Infrastructure as a Service Cloud Computing

Doudou Fall, Takeshi Okada, Noppawat Chaisamran, Youki Kadobayoshi and Suguru Yamaguchi
Internet Engineering Laboratory, Nara Institute of Science and Technology, Nara, Japan

Keywords: Cloud Computing, Security, Vulnerability Tree, IaaS, Quantification, Reliability.

Abstract: It is a truism to single out the inherent security issues of cloud computing as the main hurdle to its adoption. Particularly, infrastructure clouds are composed of multiple components and applications where vulnerabilities are regularly discovered. We propose a probabilistic security quantification method, which allows quantifying the security level of a given Infrastructure as a Service cloud environment. We translate the vulnerable IaaS environment into a vulnerability tree that we built basing on fault tree analysis, which is a well established modeling tool. The analysis of the vulnerability tree leads us to the security quantification formula.

1 INTRODUCTION

Cloud computing has revolutionized the way society uses computing resources. Its faculty to help organizations reduce their computing resources costs is undisputed. However, while it has numerous benefits, especially in regards to infrastructure costs, it also has some disadvantages that must be addressed. Indeed, users are still reluctant to adopt cloud computing because of its security issues identified in numerous survey papers (Takabi et al., 2010; Vaquero et al., 2011; Enisa, www.enisa.europa.eu; Zhou et al., 2010; Pearson and Benameur, 2009). These security issues range from the security of the multi-tenant aspect of cloud computing (Ristenpart et al., 2009) to loss of control(chow et al., 2009). It is this inherent multi-tenancy aspect of cloud computing that also dissuades customers from utilizing the cloud. In this research, we focalize our attentions on attacks that are the results of exploited vulnerabilities. In practice, many vulnerabilities remain in a cloud environment after they are discovered. This issue is due to environmental factors (latency in releasing vulnerability patches), cost factors (such as money and administrative efforts required for deploying patches), or mission factors (organizational preference for availability and usability over security). Therefore, addressing the problem of security in cloud computing is a huge challenge. Cloud computing is widely accepted as having three preeminent service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). In this work, our focus is on IaaS because not only it is the foundation of any cloud infrastructure but also it presents the highest level of multi-tenancy with the different tenants sharing storage, CPU, network bandwidth, and memory. The main contribution in this paper is a novel approach for quantifying security in cloud computing, inspired by the Fault Tree Analysis (FTA), which is commonly used in Probabilistic Risk Analysis (PRA) (which in turn is used to quantify the risk of failure in highly mission-critical systems like those of a nuclear power plant). Our security quantification consists of representing a vulnerable IaaS system as a Boolean vulnerability tree. The analysis of the vulnerability tree leads to the extraction of the quantification formula. The rest of this paper is structured as follows: section 2 presents our proposal; section 3 explains the how-to use of the Common Vulnerability Scoring System in our model; section 4 describes the quantification of multiple vulnerabilities in one component; section 5 discusses a theoretical proof of our proposal; section 6 concludes this paper.

2 QUANTIFYING THE SECURITY OF AN IaaS

2.1 Establishing a Hypothesis

Computing resources in IaaS cloud computing are typically consumed using virtual machines. By the magic of virtualization, a physical machine is sepa-

Fall D., Okada T., Chaisamran N., Kadobayoshi Y. and Yamaguchi S..
Security Quantification of Complex Attacks in Infrastructure as a Service Cloud Computing.
DOI: 10.5220/0004406601450148
In *Proceedings of the 3rd International Conference on Cloud Computing and Services Science* (CLOSER-2013), pages 145-148
ISBN: 978-989-8565-52-5

rated between several virtual machines running their own operating system. The virtual machines are isolated from each other in a multi-tenant environment. The system providing the abstraction of the hardware and managing the virtual machines is called Hypervisor or Virtual Machine Monitor (VMM). Thus, the hypervisor plays a pivotal role in IaaS as it represents the most important link of the entire infrastructure chain system. The level of security of the shared resources significantly depends on the corresponding security strength or weakness of the hypervisor. These factors culminate into the following hypothesis:

Hypothesis 1. *In a multi-tenant IaaS cloud, unauthorized access occurs if and only if the attacker succeeds in exploiting a vulnerability on the virtual machine monitor.*

This hypothesis is a pretext that we use to define complex attacks, which are any kind of attacks that involve multiple vulnerabilities (at least two). This hypothesis is the cornerstone of our proposal; most of the sections we develop further are related to it.

2.2 Typical Vulnerability Tree of an IaaS

The vulnerability tree explored in this paper, is a blueprint of a method widely used in reliability called fault tree analysis (FTA). FTA is a modeling tool that was developed to assist human being to qualitatively and quantitatively evaluate the failure of mission-critical systems such as nuclear power plants, chemical plants, and aircraft systems. FTA provides a pictorial representation of a statement in Boolean logic. The graphic consists of a top event, which is the failure of the system, and different basic events that constitute the failures of the different components that compose the system. The aim is to produce a deterministic description of the occurrence of the top event in terms of the occurrence or non-occurrence of the basic events. The vulnerability tree, which we make use of in our study, represents structure functions of univocal systems and can be described by only AND and OR logic. The vulnerability tree is comprised of a top event, which is the attack of the entire IaaS, and basic events, which represent exploitable vulnerabilities in the different components that compose the IaaS. Figure 1 gives a pictorial description of the explanation. The Vulnerability Tree Analysis (VTA) aims to provide a probabilistic description of the occurrence of the top event in terms of the occurrence of the basic events (V_X$_1$...V_X$_n$), which are vulnerabilities in the system. The shape of the schema is dictated by the hypothesis, which states that the hypervisor or virtual machine monitor (VMM) is the most critical element in the system.

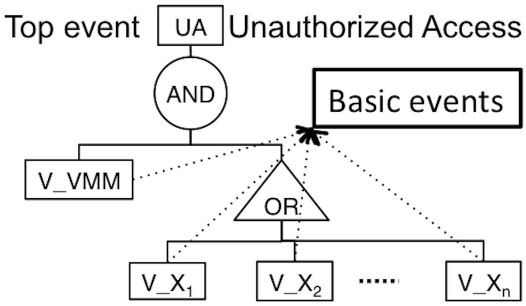

Figure 1: Vulnerability Tree.

2.3 Generating the Quantification Formula

The analysis of the vulnerability tree provides some insights to derive the quantification formula of the top event. In order to facilitate an understanding of our proposal, useful terms are defined below:

Definition 1. *A cut set is a collection of basic events such that if these events occur together then the top event will certainly occur.*

Definition 2. *A minimal cut set is a collection of basic events forming a cut set such that if any of the basic events is removed, then the remaining set is no longer a cut set.*

Considering the aforementioned definitions and the fact that the vulnerability tree follows Boolean rules, the derivation of the quantification formula is straightforward. The cut set from our vulnerability tree is given by Equation 1.

$$[UA] = [V\_VMM]_{AND}\{[V\_X_1]_{OR}[V\_X_2]_{OR}[V\_X_n]\} \quad (1)$$

By applying the probability rules to Equation 1, we obtain Equation 2, which represents our global quantification formula.

$$\begin{aligned} Q[UA] \quad = \quad & \sum_{i=1}^{n} Q[V\_VMM \cap V\_X_i] \\ & - \sum_{i<j} Q[V\_VMM \cap V\_X_i \cap V\_X_j] + ... + \\ & (-1)^{n+1} Q[V\_VMM \cap V\_X_1 ... \cap V\_X_n] \quad (2) \end{aligned}$$

We assume that the probabilities are independent but not mutually exclusive. Q denotes quantification.

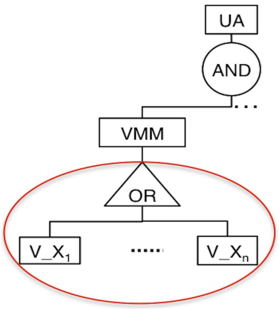

Figure 2: Example of multiple vulnerabilities present in one component.

2.4 Security Quantification of Multiple Vulnerabilities in One Component

The nature of cloud computing is reminiscent of that possibility for attackers to combine multiple vulnerabilities. In this section, our attention is on the multiple vulnerabilities an attacker may combine into a single entity like a VM, VMM or virtual network. Figure 2 illustrates this concept. In this situation, we opt for the addition rule of probability for the quantification of the aforementioned vulnerabilities. The sub-formula thereon is Equation 3.

$$Q_{MC} = \sum_{i=1}^{n} Q[V_i] - \sum_{i<j} Q[V_i \cap V_j] + \ldots + (-1)^{n+1} Q[V_1 \ldots \cap V_n] \qquad (3)$$

This sub-formula resembles to our global formula, however the construction is inherently different.

3 UTILITY OF THE COMMON VULNERABILITY SCORING SYSTEM

The Common Vulnerability Scoring System (CVSS) is a vendor-neutral open source vulnerability scoring system. It was established to help organizations to efficiently plan their responses regarding security vulnerabilities. The CVSS is comprised of three metric groups classified as base, temporal, and environmental. The base metric group contains the quintessential characteristics of a vulnerability. The temporal metric group is used for non-constant characteristics of a vulnerability, and the environmental metric group defines the characteristics of a vulnerability that are tightly related to the user's environment. In this paper, we focus more on the exploitability part of a vulnerability. The temporal and environmental base metric groups intervene after a vulnerability is exploited, therefore they do not feature prominently in our research. The remaining metric group regroups essential metrics that are used to compute the score of a vulnerability: Access Vector (AV), Access Complexity (AC), Authentication (Au), Confidentiality Impact (C), Integrity Impact (I), and Availability Impact (A). The formula of the score to date is a combination of two sub-formulas: Impact and Exploitability. The Impact sub-formula does not factor into our research because it is related to the damages that occur after a vulnerability is exploited. The Exploitability sub-formula of the CVSS is represented by Equation 4.

$$Exploitability = 20 * AV * AC * Au \qquad (4)$$

By default, these values range between 0 and 10 in the CVSS score guideline. As it pertains to our work, we are mainly concerned with probability look-alike values. Therefore, we have divided the original sub-formula by 10 to obtain our scoring formula defined in Equation 5.

$$Q(Exploitability) = 2 * AV * AC * Au \qquad (5)$$

The numerical values of the metrics *AccessVector*, *AccessComplexity*, and *Authentication* are set depending on different parameters (Mell et al., 2007). Table 1 summarizes the different criteria we apply to our security quantification formula depending on the values obtained.

4 A SOMEWHAT 'PERTINENT' EXAMPLE

In this case study, we use a cloud infrastructure that is running XEN-4.1 as hypervisor and has multiple virtual machines. After a security vulnerability scanner, the administrator discovers the vulnerabilities exposed in Table 2 with their exploitability value computed by using Equation 5. The results of the scanner revealed that there are vulnerabilities in the hypervisor, VM_1, VM_2, and VM_7. The vulnerability tree of this scenario is shown in Figure 3. The significance of this scenario is the presence of multiple vulnerabilities in some components of the infrastructure: hypervisor (two), VM_1 (three), and VM_7 (two). Before applying our global formula to the entire system, we perform the partial quantifications for those specific components by using Equation 3. Finally, we

Table 1: Qualitative and quantitative classification of scoring values.

Low	Medium	High
0 - 0.399	0.4 - 0.699	0.7 - 1

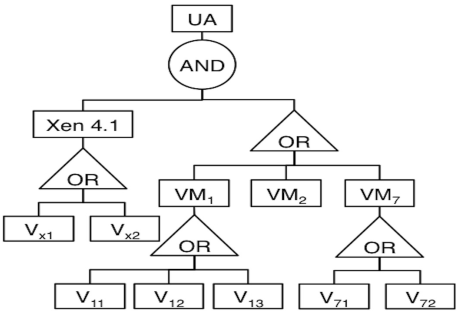

Figure 3: Tree of the concerned vulnerabilities in this example.

apply Equation 2 to quantify the security of the entire infrastructure. Table 3 summarizes all the results we acquired in this experiment. The resulting value, 0.6304, is particularly medium-to-high when we refer to our qualitative classification in Table 1. This means that the administrator of the system has to take rapid actions to patch the vulnerabilities, particularly if the reasons for not updating the system are mission or cost factors.

5 CONCLUSIONS

We introduced a unique approach for quantifying security in Infrastructure as a Service cloud computing. We developed our approach basing on industry and consumer needs and evaluated its applicability with the example described in section 5. Currently, many administrators of cloud systems use the CVSS to evaluate potential reported vulnerabilities, with the resulting score helping to quantify the severity of the vulnerability and to prioritize their response. The difference is that they do it with single isolated vulnerabilities, they do not have a response in case of mixed combined vulnerabilities. By contrast, our proposal is a response in such particular cases. We do not argue that our proposal is the ultimate security solution that will solve all the security problems in IaaS cloud systems. Our method allows quantifying security in IaaS environment when vulnaribilities are discovered in the system. In the case of unavailability of vulnaribilities, our proposal becomes inept. After the evaluation of the security level of a system, the latter still remains subject to successful attacks until the cloud administrator takes necessary measures. Therefore, our proposal does not technically prevent attacks. Furthermore, it is obvious that our method does not work for zero-day-attacks as the attacker exploits new vulnerabilities that are not referenced yet in any vulnerability databases.

Table 2: Discovered vulnerabilities and their exploitability values.

Components	Vulnerabilities	Renaming	Q (Exploitability)
Xen 4.1	CVE-2011-1898	V_{X1}	0.44
	CVE-2011-1583	V_{X2}	0.34
VM_1 (Apache 2.0)	CVE-2011-3192	V_{11}	1
	CVE-2011-4317	V_{12}	0.86
	CVE-2011-4415	V_{13}	0.19
VM_2 (MySQL)	CVE-2010-1626	V_{21}	0.39
VM_7 (Bind 9.8.0)	CVE-2011-2465	V_{71}	0.49
	CVE-2011-2464	V_{72}	1

Table 3: Summary of the results.

Components	Partial Quantification
Q[Xen 4.1]	0.6304
Q[VM_1]	1
Q[VM_7]	1
Q[IaaS]	0.6304

REFERENCES

Enisa cloud computing risk assessment. http://www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing- risk-assessment.

chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., and Molina, J. (2009). Controlling data in the cloud: Outsourcing computation without outsourcing control. In *Cloud Computing Security Workshop*. ACM Press.

Mell, P., Scarfone, K., and Romanosky, S. (2007). A complete guide to the common vulnerability scoring system version 2.0. In *FIRST*. http://www.first.org/cvss/cvss-guide.html.

Pearson, S. and Benameur, A. (2009). Privacy, security and trust issues arising from cloud computing. In *2nd international conference on cloud computing technology and science*. IEEE.

Ristenpart, T., Tromer, E., Shacham, H., and Savage, S. (2009). Hey you get out off my cloud: Exploring information leakage in third party compute clouds. In *CCS09*. ACM.

Takabi, H., Joshi, J. B. D., and Ahn, G.-J. (2010). Security and privacy challenges in cloud environments. In *Security and Privacy*. IEEE.

Vaquero, L. M., Merino, L. R., and Moran, D. (2011). Locking the sky: a survey on iaas cloud security. In *In Journal Computing - Cloud Computing Volume 91 Issue 1*. Springer-Verlag.

Zhou, M., Zhang, R., Xie, W., Qiang, W., and A.Zhou (2010). Security and privacy in cloud computing: A survey. In *6th International Conference on semantics, Knowledge and grids*.