Including Security Monitoring in Cloud SLA

Amir Teshome, Inria, IRISA
Louis Rilling, DGA
Christine Morin, Inria, IRISA

1. Introduction

Before the introduction of cloud computing, organizations used to host their own computing resources (e.g., networks, servers, storage, applications, and services). By moving to clouds, companies can achieve advantages like cost reduction (both building and management cost), getting elastic infrastructure and broad network access but they also face new problems in terms of security and lack of open standards.

A study [1] in 2014 shows that 60% of small and medium businesses were already using cloud services. However, in the same study lack of trust in service providers and security concerns were shown to be major barriers for nearly 40% of the companies. Hence it becomes clear that addressing trust and security issues will benefit not only the users but also expand revenue of the provider by attracting new customers.

We focus our work on security monitoring in clouds. Cloud security refers to mechanisms, technologies and controls deployed to protect data, applications and the associated infrastructure of cloud computing. Security mechanisms aim to achieve security principles including Confidentiality, Integrity and Availability. *Security Monitoring* is the collection, analysis, and escalation of indications and warnings to detect and respond to intrusions.

One of the risks of moving to a public cloud is losing full control of the information system infrastructure. The service provider will be in charge of monitoring the actual infrastructure and provide the required service to clients. In our work, we aim to allow providers to provide customers with guarantees on security monitoring of their outsourced information system.

1.1 SLA and Security Monitoring in Cloud

When customers need to get service from cloud providers they sign an agreement called Service Level Agreement (SLA). In general SLAs describe the provided service, the rights and obligations of both parties and state penalties for when the specified terms are not respected. Hence, an SLA helps providers to build more trust.

For example Amazon cloud service (EC2 and EBS) offers SLA terms of availability of more than 99.95% and 10% service credit in return if not respected. Also, 30% in return if the availability is less than 99.0%. However, usually SLA terms are *not related to security monitoring*. This raises trust issues for companies outsourcing their information system in the cloud.

Since you can never get ahead of the threat, monitoring cloud infrastructure's behavior is important. By monitoring a system it is possible to detect suspicious behaviors and take action before severe damage. Devices like Intrusion Detection Systems (IDS) and logs from firewalls are often used for this purpose.

To include security monitoring terms into an SLA at least the following tasks are required. First, a way for providers/clients to specify their security monitoring parameters/requirements, second mechanisms to enforce these requirements in a cloud infrastructure and finally a verification method to check if the requirements are respected at any given time.

The rest of the paper is organized as follows. Section 2 describes some related works on cloud IDSs, IDS evaluation, SLA and languages used to define SLAs. In Section 3 we discuss the challenges on inclusion of monitoring terms into SLAs. Section 4 presents the proposed methodol-

ogy and we will finish by conclusion and future work on Section 5.

2. Related works

There exists some works on both creating security monitoring devices for a cloud [11] and defining languages and frameworks for SLA description [7]. The domain specific language proposed in [11] describes the detection algorithms of the IDS rather than Service-Level Objectives (SLO, for example a set of rules) that can be negotiated before figuring in an SLA. In other words, the language is too low-level to describe SLOs.

In the field of security SLA definition Karin Bernsmed *et al.* [3] discussed what terms could be included in relation to security without describing how to include them. In [4, 6] Cascella, Jegou *et al.* describe a method to deploy applications in a federated cloud under a restriction of a given SLA. But the SLA definition is restricted to localization of data storage.

To our knowledge, there have been no attempts to include security monitoring terms in SLA. The difficulty is that SLA terms related to security monitoring devices need to be measurable and verifiable in the cloud setup.

Regarding IDS evaluation metrics, Stefan Axelsson [2] and Gu *et al* [5] presented a theoretical approach to measure IDSs and showed that metrics that don't include base rates (defined as the ratio between the number of attack network packets and the total number of network packets) do not truly describe the ability of an IDS to be practically usable. This problem is known as the *Base-Rate Fallacy* [2]. The latter one proposed a single unified metrics called Intrusion Detection Capability (C_{ID}).

In the field of security monitoring there have been different studies to develop evaluation mechanisms for existing monitoring devices and adapt them to the cloud environment. In [9, 10] Probst *et al.* describe IDS evaluation method in two phases: analysis of network access control phases and the IDS evaluation phase based on the set of services running in the virtual infrastructure.

In [8] Massicotte *et al.* also proposed an evaluation method. They used virtual infrastructure to generate traffic traces and they used the traces to evaluate IDSs in traditional servers (non-cloud environment). Both approaches measure the efficiency of an IDS, the former in a given virtualized infrastructure and the latter as a generic product, but neither of them take the *base rate* into account.

3. Challenges

We are searching for a way to allow cloud providers to offer SLA terms related to security monitoring. From the technical aspect, there are a number of challenges to include security monitoring terms in SLAs, which include:

1. The malleability of virtualized infrastructures: By its nature the cloud is very dynamic. Creation, deletion and migration of VMs is frequent. Security monitoring terms must anticipate such changes.
2. Difficulty of expressing security monitoring properties using precise terms. Lack of standards to express such terms makes the process difficult. Also, expressing monitoring terms at an abstract policy level is difficult since it will be enforced at a lower level.
3. Policy enforcement is done at the lower level: The actual monitoring happens at the packet level. At this stage, context and semantic knowledge are unknown. Also, packets could be lost which creates data loss making the monitoring difficult.
4. There is a lack of method to evaluate security monitoring setups, specifically in clouds.

Taking IDSs as an example, to include terms related to IDSs, we need to have a way to state IDS rules to apply in SLAs, mechanisms to *enforce* terms in a given infrastructure and a way to *verify* if the terms are respected at a given time. All these methods should be *independent* from the specific IDS implementation.

4. Proposed Approach

As a design requirement, making the security monitoring process - definition, enforcement and verification of SLA terms - automatic is essential, because manual management of security properties in a cloud is tedious and error prone.

First, we found measurable parameters for a given monitoring device, e.g. an IDS, and verification mechanisms for these parameters. For an IDS, *Intrusion Detection Capability* is a single unified metric, which aggregates a base rate in its formula in addition to other traditionally used parameters like detection and precision rate. Since the exact value of the base rate is unknown we used a range of statistically proposed values.

4.1 Intrusion Detection Capability (C_{ID})

C_{ID} is a metric used to evaluate IDSs, which was introduced by Gu *et al* [5]. Let *'x'* be the random variable representing the IDS input where it can be either part of an attack or a legitimate packet and *'y'* representing the IDS output where it can be detected as an intrusive or non-intrusive packet by the IDS.

- The entropy (a measure of uncertainty of information content) *H(x)* of a discrete random variable *'x'* is defined as follows, the higher value indicating the more uncertain.

$$H(x) = -\sum_x p(x) \log p(x)$$

Note that:

$$\textbf{\textit{base rate (B)}} = p(x = \text{'is an attack packet'})$$

- The mutual information *I(x;y)* which measures the amount of information shared between the two random variables is defined as:

$$I(x; y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

- The Intrusion Detection Capability (C_{ID}) can be defined as:

$$C_{\text{ID}} = \frac{I(x;y)}{H(x)}$$

Its value ranges in [0,1] and a higher value indicates a better IDS ability in accurately classifying the input packets.

4.2 Verification Mechanism

The verification mechanism runs attacks against a given configuration but without damaging the production environment. An example of the attack running environment is shown in Figure 1.

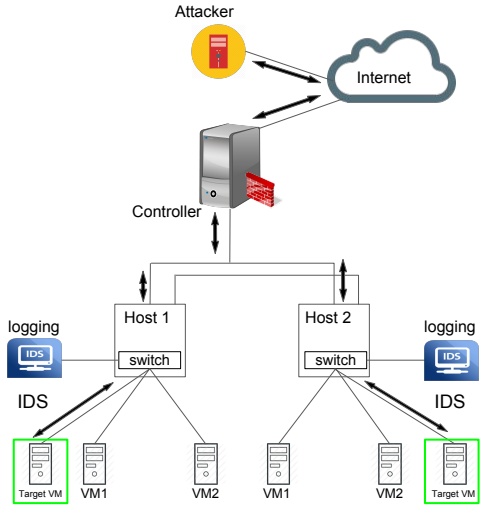

Figure 1. Attack Running Mechanism

In a given infrastructure we add a target VM (shown in a green box) after an IDS to be verified. This VM exhibits the behavior of other VMs under that IDS. Multiple target VMs could also be added in a case where a single VM is unable to exhibit all the required behaviors. An attacker machine is also added. This machine could be located inside or outside the cloud. The attacker runs a set of representative attacks and the virtual switch is configured to redirect all the attack packets towards the target VMs. Since the attack running mechanism uses the production infrastructure network resources, we must take care that the attacks have a reasonably low impact on those resources.

The rate of the occurrence of attack packets is determined by a given base rate. In this process all the out going packets from the attacker and the output of the IDS are logged. Using information from the attack packets we can differentiate true positives from false positives in the output of IDS. Using these values and the injected base rate we calculate the C_{ID}.

The verification could be done either by the provider or by the client. But clients should trust providers, in the case where the provider perform the verification.

5. Conclusion and Future Work

In conclusion there is a need to include security monitoring into SLAs. In our work we chose the C_{ID} as a usable metric to describe the efficiency of an IDS, because it takes the base rate into account. We also presented an evaluation mechanism to measure C_{ID} of an IDS dynamically using attack injection. This method is used as SLA verification mechanism.

The attack packets are redirected, as a result it will not damage the production VMs. But there is a trade-off between the evaluation methodology and performance of production infrastructure. A care should be taken since the evaluation process uses production network infrastructure (not a cloned or simulated one). In particular it needs caution for not creating unacceptable traffic load. The trade-off could also be specified in the agreement (SLA).

In this work we described SLA terms related to the efficiency of an IDS. Other aspects of the IDS could also be expressed in SLA terms, which we are planning as future work. Also, in this work we focused on IDSs, which are one of the most used monitoring devices. In the future we plan to describe other monitoring devices in SLA terms.

References

[1] Amdocs. Cloud Adoption in Small to Medium Sized Businesses, May 2014.

[2] S. Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 1–7. ACM, 1999.

[3] K. Bernsmed, M. G. Jaatun, and A. Undheim. Security in Service Level Agreements for Cloud Computing. In *CLOSER*, pages 636–642, 2011.

[4] R. G. Cascella, L. Blasi, Y. Jegou, M. Coppola, and C. Morin. *Contrail: Distributed application deployment under SLA in federated heterogeneous clouds*. Springer, 2013.

[5] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skorić. Measuring intrusion detection capability: an information-theoretic approach. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 90–101. ACM, 2006.

[6] Y. Jegou, P. Harsh, R. G. Cascella, F. Dudouet, and C. Morin. Managing OVF applications under SLA constraints on contrail virtual execution platform. In *Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualiztion management (svm)*, pages 399–405. IEEE, 2012.

[7] K. T. Kearney, F. Torelli, and C. Kotsokalis. SLA: An abstract syntax for Service Level Agreements. In *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, pages 217–224. IEEE, 2010.

[8] F. Massicotte, F. Gagnon, Y. Labiche, L. Briand, and M. Couture. Automatic evaluation of intrusion detection systems. In *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, pages 361–370. IEEE, 2006.

[9] T. Probst, E. Alata, M. Kaâniche, and V. Nicomette. An Approach for the Automated Analysis of Network Access Controls in Cloud Computing Infrastructures. In *Network and System Security*, pages 1–14. Springer, 2014.

[10] T. Probst, E. Alata, M. Kaâniche, and V. Nicomette. Automated Evaluation of Network Intrusion Detection Systems in IaaS Clouds. In *Dependable Computing Conference (EDCC), 2015 Eleventh European*, pages 49–60. IEEE, 2015.

[11] D. Riquet, G. Grimaud, and M. Hauspie. DISCUS: A massively distributed IDS architecture using a DSL-based configuration. In *Information Science, Electronics and Electrical Engineering (ISEEE), 2014 International Conference on*, volume 2, pages 1193–1197. IEEE, 2014.