Computer Networks 93 (2015) 480–491

Contents lists available at ScienceDirect

Computer Networks

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

Verification of firewall reconfiguration for virtual machines migrations in the cloud

Yosr Jarraya[a,*], Arash Eghtesadi[a], Sahba Sadri[a], Mourad Debbabi[a], Makan Pourzandi[b]

[a] *Computer Security Laboratory, CIISE Concordia University, Montreal, Quebec, Canada*
[b] *Ericsson Security Research, Ericsson Canada, Montreal, Quebec, Canada*

ARTICLE INFO

Article history:
Received 16 February 2015
Revised 1 October 2015
Accepted 6 October 2015
Available online 19 October 2015

Keywords:
Security
Verification
Stateful firewalls
Distributed firewalls
Cloud computing
Constraint satisfaction problem

ABSTRACT

While elasticity is valuable to the cloud, it may introduce security flaws due to misconfiguration after virtual machines migration. In this paper, we propose an automated approach to verify distributed firewalls reconfiguration after migration. To this end, we elaborate a language that captures distributed stateless and stateful firewalls with their underlying semantics. Integrated to Cloud Calculus, it allows specifying distributed firewalls topology. We also define semantic equivalence over stateful firewalls that forms the base for our verification approach. Furthermore, we define the property of network access control and state preservation using the concepts of soundness and completeness of firewall configurations. Additionally, we use constraint satisfaction problems to reason about our defined preservation property. Finally, we investigate the correctness and scalability of our approach.

1. Introduction

The appeal of cloud computing lies in its elasticity, i.e. its capacity to increase or decrease the allocated resources for accommodating workload spikes or resource shortages via seamless migration of virtual machines (VMs). In spite of these benefits, elasticity may cause critical security issues due to misconfigurations. More precisely, multiple security appliances including packet filters and stateful firewalls are typically deployed to protect the data center resources and its hosted VMs. When a VM migrates, filtering rules should be reconfigured at source and destination locations and state-related information (e.g. existing connections) stored by stateful security appliances and called security context, should be correctly moved to the destination appliances. The importance of security context migration was endorsed by the research community and by the industry. Stateful firewalls use connection state information derived from past communications to make dynamic control decisions. While moving VMs, removing the previous configuration and simply reapplying the policy will cause loss of important state information, dynamically configured rules, and legitimate ongoing connections. As the migration mechanisms may fail, their correctness should be guaranteed (e.g. using auditing mechanisms). This is better illustrated by a bug discovered in a widely used cloud infrastructure management system, namely OpenStack, which prevents security groups from being reapplied after live migration[1]. As scale and complexity of data centers are continually increasing and VMs are dynamically and frequently moved, manually verification is unrealistic.

In this paper, we propose a reliable framework for detecting and reporting misconfiguration problems in distributed stateful and stateless firewalls due to VM migrations.

[*] Corresponding author. Tel.: +1 514 848 2424; fax: +1 514 848 3171.
E-mail address: y_jarray@encs.concordia.ca (Y. Jarraya).

[1] OSSA-2013-030: https://security.openstack.org/ossa/OSSA-2013-030.html.

http://dx.doi.org/10.1016/j.comnet.2015.10.008
1389-1286/© 2015 Elsevier B.V. All rights reserved.

Previous works on the verification of stateless firewalls proposed to do it with respect to the security policy. However, stateful firewall's connection states are dynamic information that cannot be captured by the security policy. We advocate that the verification of stateful firewalls compliance after migration can be performed by checking the configuration and state after migration with the last known secure configuration and state, before migration. The main contributions of this paper are manifold:

- Elaborate a specification language for stateless and stateful firewalls that captures the filtering rules and the security context states with their underlying semantics described using a denotational style [1].
- Define formally network access control and state preservation property and use the aforementioned semantics to reason about it.
- Derive a systematic encoding of distributed stateful and stateless firewalls into a network of constraints expressed in the language of the Sugar constraint solver [2].
- Elaborate an algorithm to generate constraint satisfaction problems (CSPs) formulas that allows the verification of network access control and state preservation property and identifying errors, if any.

The paper is organized as follows. Section 2 reviews the related work. Section 3 describes the case study. Section 4.3 presents the firewall specification language. Section 5 defines network access control and state preservation and how to reason about it. Section 6 presents our verification approach. Section 7 discusses the complexity of our approach. Section 8 demonstrates the application of our approach on the working case study and how it can be used to detect misconfiguration. Section 9 presents performance evaluation of our approach. We conclude the paper in Section 10.

2. Related work

Two main research streams on firewall analysis exist: anomalies detection and resolution, and compliance verification. Several works target anomaly detection and resolution in stateless (e.g. [3–7]) and stateful firewalls (e.g. [8,9]). While these works are valuable, they cannot detect rule addition or omission. Our work addresses the second topic.

Brucker et al. [10] use test case generation for stateless firewalls; however, this is not suitable for highly dynamic environments such as the cloud. Hassan and Hudec [11] propose equivalence checking between security policy and stateless firewall access control rules. Gawanmeh and Tahar [12] use Event-B and invariants checking to verify firewall configurations consistency against the policy. Acharya and Gouda [13] show that any algorithm solving equivalence of stateless firewall verification can be also used to solve the firewall redundancy checking problem, and vice versa with the same time and space complexities. They highlighted that whether these results would apply on stateful firewalls is still an open problem. Satisfiability verification has been proposed in other works [14–16]. Al-Shaer et al. [17] model network access control policies as binary decision diagrams (BDDs) and use symbolic model-checking to verify reachability and other security requirements.

Gouda et at. [18] verify the correctness of network of stateless firewalls with tree topologies using their own defined formal model, namely firewall decision diagram, and custom verification algorithm to verify accept and discard properties between each pair of domain nodes. While all these works target stateless firewalls, we address the problem of statefull firewalls, which encompasses the verification of dynamically created information that cannot be captured by static security policy. Instead of comparing the configuration with the security policy, we compare the current security configuration with the last known secure configuration, which will be shown to be well-suited for frequently changing environments.

Several works propose a language for stateless firewalls (e.g. [13,19]), and others for stateful firewalls (e.g. [8,20]). The language proposed in [8] is mainly used for firewalls design. Operational semantics is proposed in [20] for stateful firewall rules, but not considering ongoing connections states. Our proposed language mainly differs at the semantic level. To the best of our knowledge, we are the only work proposing a denotational semantics for stateful firewalls and defining a semantic equivalence over them. Finally, our language is integrated in Cloud Calculus [21] to specify distributed firewalls.

3. Cloud deployment case study

A typical cloud deployment of three-tier web applications consists of: web, application, and database. We consider two data centers (DW) and (DE), hosting muti-tenants VMs as depicted in Fig. 1. Network topologies are two-level trees (edge and core tiers) for both data centers. Between two given zones, there exists a firewall path that enforces the access control between them.

We suppose that the VMs are grouped within logical security groups (i.e. web, app, db) as shown in Fig. 1. In order to secure its instances, a firewall policy is specified for each VM group as follows: (1) Web group: allow any to connect on ports 80 (HTTP) and 443 (HTTPS), (2) App group: allow only web group to connect to port 8000, (3) Db group: allow only application group to connect to port 3306, (4) Default (for all): allow corporation network (CorpN) on port 22 (SSH) for Secure Shell. We consider the case where Vi_4 has to be migrated from DW to DE to be collocated with the group app2. An excerpt of the firewall rules before and after migration are summarized in Table 1.

An excerpt of related state tables is shown in Fig. 1. For instance, IP1 could be a user that is actually connected (established TCP connection) to the web service instance Vi_4 through port 443. Since the stateful firewall $sF1$ connects their respective zones, a record of the connection state is kept in the state table showing the current state of the connection (estab) and the involved parties (IP1 and Vi_4) with their respective ports. Adding to the migration of Vi_4-related connection states, some firewalls have to be reconfigured after migration.

4. Distributed firewall specification language

A summary of the notation used in this paper is available online [22]. This section presents our language that captures the syntax and semantics of distributed firewalls. It is