An efficient method for ontology-based multi-vendor firewall misconfiguration detection: A real-case study

Ruben F. Cordova[1], Armando L. Marcovich[2], Cesar A. Santivanez[1]
[1]*Advanced Networks Research Lab* (*GIRA*), *Pontifical Catholic University of Peru* (*PUCP*) Lima, Peru
[2]*Inndatsys Consulting*. Research while at *Centauri Technologies Corporation*, Panama City, Panama
ruben.cordova@pucp.pe, amarcovich@inndatsys.com, csantivanez@pucp.pe

Abstract—**Large enterprises employ a variety of firewalls, possibly from different vendors each with its own rule syntax. Furthermore, enterprise policy may be mapped to hundreds of rules on each device. Manual configuration of a large set of rules is a complex process that may result in misconfigurations and the resulting in security vulnerabilities. A promising alternative is the use of semantic web technologies (an ontology combined with a query language or reasoner) to detect firewall misconfigurations. However, a poorly designed ontology may result in excessive memory consumption and processing load, rendering the method ineffective. In this paper, we present an efficient ontology design for detecting misconfigurations on firewall rules, that attempts to reduce the computing resources needed to validate the firewall rules of the companys policies. The design was tested on a real-world scenario of an enterprise with equipment from 3 different vendors: Fortinet, Cisco ASA, and Checkpoint. Our solution was able to detect over a hundred misconfigured rules. Finally, an evaluation of the impact of the chosen combination of ontology, query language, and reasoner on the computational cost is also presented.**

Index Terms—**Firewall management, misconfigured rules, anomaly detection, semantic web, ontology.**

I. Introduction

Firewalls play a key role on enterprise networks security, protecting it from attacks. Entities like the National Institute of Standards and Technology (NIST) have published some guidelines for appropriate security policy planning and implementation [1]. Therefore, configured firewall rule-set must be consistent with enterprise security policy as well as with best practices. Managing these rules is a manual, complex, time-consuming and error-prone task that could result in a misconfigured firewall (i.e. blocking valid enterprise traffic).

In the past 20 years, several algorithms have been proposed to model firewall rules and automate the detection of anomalies in firewall rule-set. Also, there have been efforts to use Semantic Web technologies to check database consistency, contrasting database entries with enterprise policy [2]. This paper proposes to leverage and combine these efforts to use query-based Semantic Web technologies for detecting firewall misconfigurations/anomalies.

An anomaly is defined as a potential conflict between firewall rules [3]. There are at least 4 types of anomalies:

- Shadowing: rule not executed because packets match a preceding rule
- Redundancy: specific rule has the same action and matches the same packets than a general rule
- Generalization: preceding rule is an exception (different action) of a general rule
- Correlation: rule matches some packets other rule (and vice-versa)

We extended one of the most well-known algorithms for detecting anomalies in the configuration of firewalls [3] to use Semantic Web technologies. Since a poorly designed ontology may incur on computational resources, the efficient use of them was a key design goal for the semantic analyzer. We propose a vendor-neutral ontology to model firewall rules and the use of computationally-effcient SPARQL queries to find what anomalies may exist in firewall configuration.

The main contributions of this paper are (i) it provides an ontology able to analyze multi-vendor firewalls, (ii) proposes the use of computationally efficient SPARQL queries to collect information from the ontology, and (iii) compute and compare the computational cost of our proposed query retrieval method against the use of a reasoner.

The rest of this paper is organized as follows: in section (II) we discuss the existing methods for examining firewall rules; in section (III) we formulate the problem based on how to analyze firewall configuration with different syntax and how to evaluate computational cost of using semantic tools; in section (IV) we show the taxonomy of the proposed ontology used to detect misconfigurations; in section (V) we present the conditions where the tests were performed and the results obtained; finally in section (VI) we present our conclusions.

II. II. State of the art

There have been many efforts to design algorithms to automate the detection of misconfiguration in firewalls. In [3] Al-Shaer et al. defined the anomalies that may exist in a firewall rule set based on the relations between rule fields (protocol, source and destination address and port, and action). They proposed an algorithm for anomaly discovery based on these relations, and in [4] they extended these

This work was supported by grant IE-2015 from Senacyt in Panama.

978-1-5386-8372-9/18/$31.00 ©2018 IEEE

definitions and analysis to distributed firewalls. In [5] Abedin et al. proposed some modifications to the anomaly definitions and presented an algorithm that not only detect but resolve anomalies by reordering and splitting rules. These works tested the correctness of the algorithms with a generic firewall like Netfilter, but no other firewall vendors were considered.

Fitzgerald et al. introduced in [6] the usage of description logic through employing different Semantic Web technologies (ontologies, SWRL rules and reasoners) for detecting anomalies, specifically shadowing, in firewall rules. As they remarked, using description language enables the network administrators to provision firewall configuration in a reliable and human-convenient way. They did not consider the impact in resource consumption of inefficiently defining an ontology for large firewall rule-set and using a reasoner to infer knowledge. They did not consider the use of queries instead of a reasoner.

With the advent of stateful firewalls, there were some efforts ([7] and [8]) to extend the analysis of anomalies for stateless firewalls to stateful ones. Additionally, in [9] and [10] the authors extended the analysis of misconfigurations to include various distributed security network devices (firewalls, using Binary Decision Diagrams, and routers with Access Control Lists, using Boolean satisfiability). In [11] and [12], the authors proposed new methods to detect and solve anomalies. They presented scalability and performance evaluations related to the number of rules in the firewall, but they did not take into account the rule definition structure. We consider this last criterion to be important because each vendor has its own syntax for configuring firewalls (i.e. address group).

To the best of our knowledge, this work is the first in proposing a standard ontology for multi-vendor firewalls and employing computationally efficient queries for information retrieval, evaluating its computational cost.

III. Problem formulation

Large enterprises may present scenarios with firewalls from different vendors. Unfortunately, there is no common syntax for defining their rule configurations. Currently, there is a gap for standard modeling of firewall information in a multi-vendor environment. Recently, modeling data using Semantic Web technologies has gained a lot of attention due to its ability to describe and provide meaning to data. With this, computers gain the ability to understand it, infer new information from it, and even make decisions based on it. There exist 2 possibilities for retrieving information with semantics: using queries or a reasoner.

- A query language, such as SPARQL, allows to formulate questions to the ontology: it is only possible to retrieve information explicit in the ontology
- A reasoner allows to derive new information from the existing data

Depending on the kind of information one is expecting to retrieve, one should decide whether querying the system is sufficient or if it is necessary to use a reasoner. Even though this decision seems to be trivial, choosing one with an inefficient ontology could result in a high resource consumption.

IV. Ontology Description

In order to create the ontology, we will first describe our algorithm for detecting anomalies on firewalls. We based our work in [3], from which we take the definition of possible relations and anomalies, and the anomaly discovery algorithm.

We employed Semantic Web technologies to model the firewall rules and retrieve the information about the existing misconfigurations. We use RDF to model firewall data (in RDF/XML format), RDFS and OWL for creating the relations (vocabulary and its meanings) between the elements that compose a firewall rule. Figure 1 presents the taxonomy of the ontology regarding the components of firewall rules. Blue links refer to the modeling of firewall rules, while green links refer to the decomposition of rules into subrules (purple links are employed by both). Notice that we had to define a class named “SubRule” because a rule may be defined with many address and service groups. In order to properly define the intersections to detect anomalies, we had to decompose each rule into subrules. We also modeled the relations between firewall rules components in the anomaly discovery algorithm as object properties.

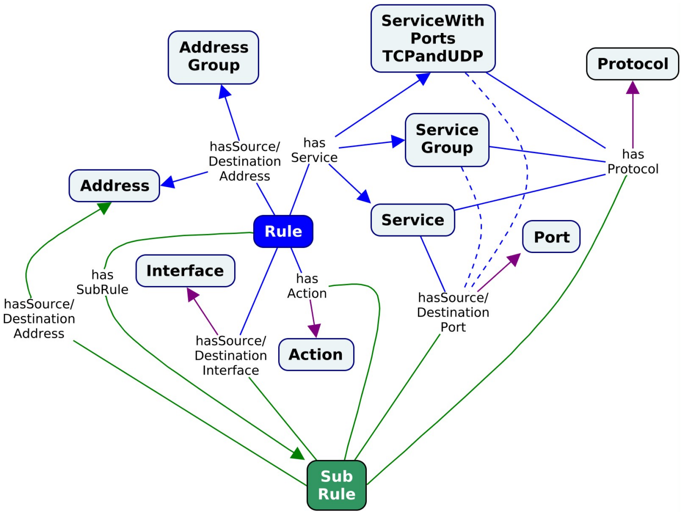
Address Group
ServiceWith Ports TCPandUDP
Protocol
hasSource/ Destination Address
has Service
Service Group
has Protocol
Address
Service
Rule
Port
has SubRule
Interface
has Action
hasSource/ Destination Port
hasSource/ Destination Address
hasSource/ Destination Interface
Action
Sub Rule

Fig. 1. Taxonomy of proposed ontology.

To get the information of existing anomalies, we use SPARQL as the query language to look for patterns that identify anomalies, as shown in Table I. We chose it over reasoning because of its efficiency to retrieve anomaly information without loss. For instance, consider that a firewall ontology with SubRules 1A and 1B (decomposition of Rule 1), both of type SubRule. A reasoner, besides detecting the anomaly defined using SWRL rules, would infer that these SubRules are also of type Rule. Instead, we could query the ontology to get the inmediate class each SubRule belongs to. Furthermore, an intelligent choice of the query (regarding statements and their order) impact in the running time.

V. Evaluation and results

We present the results of 3 experiments. First, we verify the correctness of the semantic analyzer using a configuration

TABLE I
EXAMPLES OF ANOMALY DETECTION QUERYS

Anomaly	***SPARQL Query***
Shadowing	[Rx type Rule][Ry type Rule][Rx hasRuleNum Nx][Ry hasRuleNum Ny][Sx type SubRule][Sy type SubRule][Sx sameProto Sy] [Sx sameOrIncSrc Sy][Sx sameOrIncDst Sy][Sx hasAction Ax][Sy hasAction Ay][Ax differentFrom Ay][Nx gtrThan Ny]
Redundancy	[Rx type Rule][Ry type Rule][Rx hasRuleNum Nx][Ry hasRuleNum Ny][Sx type SubRule][Sy type SubRule][Sx sameProto Sy] [Sx sameOrIncSrc Sy][Sx sameOrIncDst Sy][Sx hasAction Ax][Sy hasAction Ay][Ax sameAs Ay][Nx gtrThan Ny]
Generalization	[Rx type Rule][Ry type Rule][Rx hasRuleNum Nx][Ry hasRuleNum Ny][Sx type SubRule][Sy type SubRule][Sx sameProto Sy] [Sx sameOrIncSrc Sy][Sx sameOrIncDst Sy][Sx hasAction Ax][Sy hasAction Ay][Ax differentFrom Ay][Nx lessThan Ny]
Correlation	[Rx type Rule][Ry type Rule][Rx hasRuleNum Nx][Ry hasRuleNum Ny][Sx type SubRule][Sy type SubRule][Sx sameProto Sy] [Sx sameOrIncSrc Sy][Sy sameOrIncDst Sx][Sx hasAction Ax][Sy hasAction Ay][Ax differentFrom Ay][Nx gtrThan Ny]

file with known anomalies. Then, we verify that our solution is vendor-independent in a Panamanian company employing firewalls from 3 different vendors using configuration files provided by Centauri Technologies Corporation. Finally, we verify that our query-based solution is resource efficient by comparing its running time and memory consumption against the use of a reasoner to detect anomalies in Fortinet firewalls.

A. *Correctness of semantic firewall analyzer*

To validate that our algorithm correctly detects anomalies, we employed the test rule-set proposed in [3]. We adapted the Netfilter rules to each vendor syntax (Fortinet, Cisco and Checkpoint) and used them as input to the proposed ontology, obtaining the same presented anomalies, except by the following: some correlated rules (i.e. 2 and 3) should not be detected because, by definition, correlation only occurs when both rules have different actions, and some generalization anomalies not listed there (i.e. rule 9 is a generalization of rules 2, 3, 6, 7, 8) that our system was able to find.

B. *Testing in a multivendor environment*

We used our system to detect anomalies in other commercial firewalls using queries. These were a Cisco ASA and Checkpoint firewalls. Table II shows the results of resource consumption. If we focus only in Checkpoint rules, there were 101 defined rules. From these, when extending them into subrules, there are 4025. The anomaly detection results were 818 redundant, 3185 generalized and 22 correlated rules.

TABLE II
REASONING VS. QUERYING FOR ANOMALY DISCOVERY

No. of Rules	**Test Results**				
	Vendor	***Executed time***		***Memory consumption***	
		Reasoning	***Querying***	***Reasoning***	***Querying***
22	Fortinet	1h42m	3m7s	∼ 8.9 GB	6 GB
37		Did not finish	1h30m	> 15 GB	∼ 7.5 GB
13	Cisco ASA	-	5.75s	-	300 MB
101	Check point	-	3h53m	-	2.4 GB

C. *Comparing resource consumption against a reasoner*

It is shown in Table II that when using the Pellet reasoner with Fortinet configuration files, the system spent near 2 hours to analyzing 22 firewall rules, and it did not complete the analysis for another of 37 rules because the RAM memory was almost full. However, when using SPARQL queries the time execution drastically reduces. This is why we proposed to use queries to detect anomalies in firewall rules.

VI. CONCLUSIONS

Firewall management is a complex, error-prone task because of the number of rules to consider. In this paper we used Semantic Web technologies to model firewall rules and extend one of the most used algorithms for anomaly discovery. In order to employ this semantic analyzer with real data, first we validated the consistency of our algorithm. After that, we tested the semantic analyzer with real firewall configuration files from 3 vendors. It was shown that when poorly defining the ontology and choosing a semantic web tool to retrieve anomalies, the computational cost of using the proposed method can be very elevated. also, we could identify 3 different types of anomalies within the rule-set, with redundancy and generalization having a high rate of occurrence.

REFERENCES

[1] K. Scaforne and P. Hoffman, (2009, Sep.), Guidelines on Firewalls and Firewall Policy, National Institue of Standards and Technology, [Online], Available: https://www.nist.gov/publications/guidelines-firewalls -and-firewall-policy

[2] Centauri Technologies Corporation, Consultora e I+D, [Online], Available: http://www.centauritech.com/consultoria-e-id

[3] E. Al-Shaer and H. Hamed, "Firewall policy advisor for anomaly discovery and rule editing," Integrated Network Management VIII, Springer, Boston, MA, 2003. 17-30.

[4] E. Al-Shaer and H. Hamed, "Discovery of policy anomalies in distributed firewalls," INFOCOM 2004.

[5] M. Abedin et al., "Detection and resolution of anomalies in firewall policy rules," IFIP Annual Conference on Data and Applications Security and Privacy, Springer, Berlin, Heidelberg, 2006.

[6] W. Fitzgerald, S. Foley, and M. O'Foghlu, "Confident firewall policy configuration management using description logic," 12th Nordic Workshop on Secure IT Systems, 2007.

[7] L. Buttyn, G. Pk, and T. V. Thong, "Consistency verification of stateful firewalls is not harder than the stateless case," Infocommunications Journal 64.1 (2009): 2-8.

[8] J. Garcia-Alfaro et al., "Management of stateful firewall misconfiguration," Computers & Security 39 (2013): 64-85.

[9] L. Yuan et al. "Fireman: A toolkit for firewall modeling and analysis," Security and Privacy, 2006 IEEE Symposium on. IEEE, 2006.

[10] S. Maity, P. Bera, S. K. Ghosh, and E. Al-Shaer, "Formal integrated network security analysis tool: formal query-based network security configuration analysis," IET Networks 4.2 (2014): 137-147.

[11] B. Khorchani, S. Hall, and R. Villemaire, "Firewall anomaly detection with a model checker for visibility logic," Network Operations and Management Symposium (NOMS), IEEE, 2012.

[12] A. Saadaoui, N. B. Souayeh, and A. Bouhoula, "Formal approach for managing firewall misconfigurations." Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference.