Automating the configuration of firewalls and channel protection systems in virtual networks

Automating the configuration of firewalls and channel protection systems in virtual networks

Daniele Bringhenti, Riccardo Sisto, Fulvio Valenza
Dip. Automatica e Informatica, Politecnico di Torino, Torino, Italy, Emails: {first.last}@polito.it

***Abstract*—Network virtualization has revolutionized the traditional approaches for security configuration. If in the past error-prone and unoptimized manual operations were performed by human beings, nowadays automated methodologies are employed for establishing the configuration of virtual security functions that can enforce the requested security properties. However, these techniques can only perform the automatic configuration of a single function type at a time. This restriction may be excessively limiting, because the configuration of some functions may directly impact others, and they cannot be configured in sequence. In light of these considerations, the paper investigates the stated problem for the two most commonly used security functions, packet filtering firewalls and channel protection systems. It also proposes a preliminary approach to automatically perform their joint intent-based configuration, by defining the problem through a Maximum Satisfiability Modulo Theories formulation.**

***Index Terms*—security configuration, firewall, channel protection, network virtualization**

I. Introduction

The flexibility and agility introduced into networking by novel virtualization paradigms, such as *Network Functions Virtualization* (NFV) and *Software-Defined Networking* (SDN), have benefited the security configuration [1]. Automation has been leveraged, through intent-based techniques such as policy refinement [2] and collaborative attack mitigation [3], for establishing the configuration of virtual functions employed to enforce security properties (e.g., connectivity requirements, authentication, confidentiality). Automated tools have been progressively replacing the traditional manual interventions performed by human beings, which were error-prone and were becoming unfeasible due to the higher complexity of modern networks [4].

In literature, several automatic approaches have been proposed in recent years, but most of them focus on a specific type of function at a time, e.g., packet filtering firewalls for enforcing reachability and isolation properties [5] [6], or *Channel Protection Systems* (CPSs) for enforcing confidentiality, integrity, and authentication [7] [8]. Therefore, a simultaneous automatic configuration of multiple types of security functions is not performed. One may argue that it is enough to configure a single function type at a time, and then the sequence of different configuration operations would be correct. Nevertheless, such an approach might be both restrictive and error-prone. On the one hand, optimizations which would be enabled by a joint configuration are missing. On the other hand, the actions enforced by functions of a certain type may impact the others, if they are configured in sequence instead of simultaneously.

In light of these considerations, this paper addresses the problem of a joint automated configuration for two of the most used security function types, i.e., firewalls and CPSs such as VPN gateways. In doing so, it proposes two main contributions. Firstly, an analysis of the problem is illustrated, with the aim to motivate why configuring simultaneously firewalls and CPSs, in an automatic way, allows to reach a higher lever of security and higher confidence in the fact that their global configuration is correct. Secondly, a preliminary intent-based methodology for their joint configuration is presented. This methodology is based on a correctness-by-construction approach, where there is a formal guarantee that the computed configuration is compliant with the requested security requirements without the need of applying other a-posteriori formal verification techniques. This property is achieved through the formulation of the configuration problem as a *Maximum Satisfiability Modulo Theories* (MaxSMT) problem, which also allows the enforcement of optimization objectives, such as the minimization of functions allocated in the virtual service, or the minimization of their configuration rules.

The remainder of this paper is structured as follows. Section II discusses the related work, emphasizing the differences with the approach proposed in this paper. Section III states the problems deriving from a disjoint automatic configuration of firewalls and CPSs. Section IV lays the preliminary foundations for an approach where these functions are simultaneously configured. Finally, Section V briefly concludes the paper and prospects future works for further development of this approach.

II. Related Work

Packet filtering firewalls represent the security function type that has been mainly investigated in the literature about automatic security configuration. The research took the first steps in this area before the advent of network virtualization paradigms. However, the initial ideas that have been proposed ([9]–[11]) could only be applied to traditional hardware-based firewalls, and in most of the cases only for centralized border firewalls instead of distributed ones. Later, softwarization gave the required push for making steps forwards in this research line ([5], [6], [12]–[16]). In some cases, such as in [15] and in [6], formal verification techniques were used, following the trends of another group of studies related to firewall

Action	IPSrc	IPDst	pSrc	pDst	tProto
deny	192.174.1.∗	192.174.2.∗	∗	∗	∗
deny	192.174.2.∗	192.174.1.∗	∗	∗	∗
deny	192.174.1.∗	150.10.0.∗	∗	80	TCP
allow	192.174.1.∗	100.10.0.∗	∗	≠80	TCP
allow	192.174.1.∗	100.10.0.∗	∗	∗	UDP
allow	192.174.2.∗	100.10.0.∗	∗	∗	∗

TABLE I: Connectivity Policies

Action	FilteringConditions	protectionInfo	trustRequirements

encrypt	IPSrc = 192.174.1.∗
	IPDst = 144.14.2.∗
		AES-GCM-256	Untrusted nodes =
			{125.2.2.2}

encrypt	IPSrc = 122.33.33.3
	IPDst = 12.67.84.2
		3DES-CBC	Trusted nodes =
			{55.44.33.22}

authenticate	IPSrc = 192.174.1.1
	IPDst= 22.134.2.∗
		HMAC-SHA-256	Untrusted nodes =
			{125.2.2.2}

TABLE II: Channel Protection Policies

configuration ([17]–[19]). Even though all the ideas shared by these works have been fundamental for the field of automation for security configuration, all their efforts have been spent exclusively for firewalls, without considering possible CPSs that may require a configuration as well.

A similar argument applies to the research related to the automatic configuration of CPSs. The problem has been considered relevant since before network virtualization ([20]–[22]) and has continued to be researched after its advent ([7], [8]). In fact, in all these works the automatic configuration problem is not addressed for firewalls jointly with CPSs, and most of the approaches are not optimized or do not leverage formal verification techniques for providing correctness assurance. From this point of view, the progress has been slower than in the research line about firewall configuration, due to the much higher complexity of CPSs and larger variety of different solutions (e.g., IPSec-based VPN gateways, SSH tunnel terminators, etc.).

Only few studies ([23], [24]) have tried to address the configuration problem for multiple types of security functions at the same time. All of them are limited and do not address the most serious problems related to a joint automated configuration. [23] uses a theorem proving method, called B Method, to perform a top-down refinement of global policies into network security component configurations. However, it is not clear if the proposed model for that method is suitable for virtual networks, and how well the solution effectively performs. Instead, [24] presents a full workflow for a formally correct automated security orchestration and embedding in virtual networks. Nevertheless, not enough details are provided for understanding how intra-function conflicts may be avoided in automatically configuring multiple types of virtual functions simultaneously.

III. Problem Statement

Packet filtering firewalls and CPSs are security functions that are employed to enforce different types of *Network Security Policies* (NSPs), i.e., different kinds of security properties.

The former are used to enforce connectivity policies, which require to satisfy isolation or reachability properties. An isolation policy may require that a specific network traffic must be blocked before reaching its destination, while a reachability policy may require that a traffic must reach the destination without being discarded by any intermediate function. The traffic is identified in these NSPs by means of the values characterizing the fields of the IP 5-tuple, i.e., source and destination IP addresses, source and destination ports, and transport-level protocol. In fact, with respect to more advanced technologies such as web application firewalls or anti-spam filters, a packet filtering firewall can only take decisions on network information belonging to the layers 3 and 4 of the ISO/OSI stack. Some examples of connectivity policies are shown in TABLE I.

The latter are used to enforce channel protection policies, which express security requirements concerning the so-called CIA triad: confidentiality, authentication and integrity. For this reason, a channel protection policy should be characterized by the following elements:

- the action that a corresponding CPS must enforce (e.g., encryption or authentication);
- the filtering conditions which allow to establish on which packets the policy action must be applied (e.g., the fields of the IP 5-tuple);
- information about the protection algorithms that must be employed for enforcing the corresponding security property (e.g., AES-GCM-256 for confidentiality, HMAC-SHA-256 for authentication and integrity);
- trust requirements, expressing information about the security status of other network nodes. More specifically, some nodes may be classified by a channel protection policy as untrusted, if it is necessary that the traffic crosses those nodes with robust security protections, or trusted, if the traffic can cross them plain and be inspected by them as well (e.g., for intrusion detection purposes). Note that these trust requirements are valid only for the traffic identified by the condition of a single policy. Therefore, if a network node is considered untrusted for a traffic, it may be trusted for another one.

Some examples of channel protection policies are shown in TABLE II.

In light of these considerations, apparently connectivity and channel protection policies serve different purposes, and therefore it could be not easy to understand the reasons why a joint configuration of firewalls and CPSs to enforce these policies may be necessary. In fact, a possible argument would be that, if first all the firewalls are allocated and configured in the logical topology of a virtual network, and then the same operations are performed for CPSs (or vice versa), then the resulting outcome not only would be the same, but it would be totally adherent to the security specifications of the policies. However, this is not right, because in that case either sub-optimizations or conflicts, only resolvable through a joint configuration, would occur. These issues can be classified into three categories: deployment model, rule generation, and technology choice.

1) Deployment Model: On the one hand, CPSs might be used to protect the communications opened by a single device (customer-provisioned model), or the traffic generated by a large group of devices (provider-provisioned model). On the other hand, packet filtering functionalities might be enforced either with a single border firewall, if all the traffic of interest converges to a single network point (centralized model), or multiple firewall instances, for instance throughout the employment of personal firewalls (distributed model). If the configuration of a security function type is automatically performed before the other, the choices related to the deployment model may be unoptimized. The reason is that the deployment model adopted for a function type might constrain the other. For example, we can suppose that firstly a large number of CPSs are allocated in the virtual service throughout a customer-provisioned deployment model. Later, a deployment model is chosen for packet filter firewalls. At that point, some possible solutions might have become unfeasible. In particular, having a single centralized firewall might not be possible anymore, because it might not have access to the plain traffic generated by the end-point devices. Instead, if a joint configuration is performed, it is possible to reason about connectivity requirements while enforcing communication protection policies (and vice versa). At that point, the best trade-off between the optimal deployment models for the two function types can be really chosen.

2) Rule generation: If the deployment model is about the way function instances are allocated in the virtual network topology, the rule generation is the automatic operation which establishes how the rule sets of the functions are computed to enforce the security policies. The encryption rules configured on a CPS may impact the filtering rules of a firewall, and vice versa. For example, if a firewall is configured with a rule to block a specific traffic (initially plain), in case later a CPS is configured so as to encapsulate that traffic with a solution such as IPSec in tunnel mode for encrypting the IP and TCP headers of the internal packets, then the original firewall rule becomes incorrect, because defined over the fields of the plain traffic. This issue is more serious than a simple sub-optimization, because it is an inter-function conflict anomaly [25], i.e., a conflict arising among rules of different types of security functions. It is indeed true that not always conflicts would arise, but only a joint configuration would really allow to always define rules for a function type that are compliant and consistent with the rules established for the other type.

3) Technology and implementation choices: Nowadays there exists a huge variety of technological solutions available for channel protection, as some studies have reported [8]. This complexity adds up to the availability of different function implementations, both for firewalling and channel protection, since each implementation is a software program that can be created by any developer and can easily run on Virtual Machines or Docker containers. Typically, the best choice of which technology and implementation is the optimal one to be employed takes place in light of the chosen deployment model and of the generated rules. As a consequence, also this choice would be optimized and devoid of more serious issues only if it occurs after a joint configuration.

In light of the aforementioned problems, a joint configuration of firewalling and channel protection functions is a solution that is worth being investigated. Nonetheless, the complexity of implementing such a solution is high. In fact, for distributed functions, the term configuration involves two different aspects: definition of the allocation scheme (i.e., how the functions are allocated in the virtual topology) and generation of filtering/channel protection rules. Manual approaches are evidently unfeasible because they would be subject to human errors, and automation should be instead leveraged.

IV. The Proposed Approach

The problem of automatically configuring firewalls and CPSs jointly can be addressed in multiple ways. The preliminary approach that is proposed in this paper is not limited to reach a valid solution for the configuration problem, but aims two additional objectives. The first one is to provide formal assurance of the solution correctness with respect to the specified NSPs, whereas the second one is to optimize the solution so that the minimum number of function instances is allocated, and the minimum set of rules is computed for them.

These objectives are achieved throughout the formulation of a *Maximum Satisfiability Modulo Theories* (MaxSMT) problem. This formulation enriches the definition of the more traditional *Satisfiability* (SAT) problem from two points of view:

- a SAT problem is only characterized by formulas expressed within Boolean algebra, with the consequence that the expressiveness freedom is restricted. Instead, a MaxSMT problem allows to use other theories (e.g., string or integer theories), whence the acronym SMT. These theories are necessary to represent the behavior of complex network functions, such as network address translators and load balancers, and of the two security functions analyzed in this paper;
- a standard SAT/SMT problem can be composed only of hard constraints, i.e., it must exist an interpretation that makes them satisfied. In case no interpretation is found, it means the problem cannot be solved. Instead, a solution among all the possible ones is selected, even if it may be less optimized than others. Instead, in a MaxSMT formulation, there are also weighted soft constraints. These clauses do not strictly require satisfaction. However, since the solution for a MaxSMT problem aims to maximize the sum of the weights associated with the satisfied soft constraints, these clauses are satisfied as far as possible. The soft constraints thus represent optimization objectives, which are enforced only if possible.

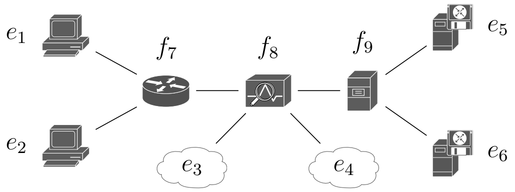

Fig. 1: Original Service Graph without firewalls

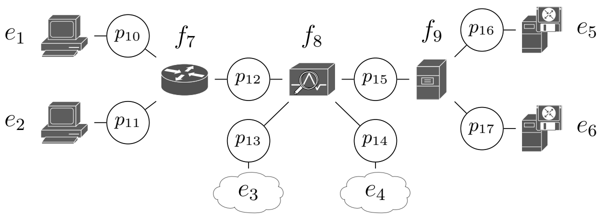

Fig. 2: Allocation Graph with Allocation Places

The remainder of this section is structured as follows. Subsection IV-A illustrates the inputs that are required for the definition of the MaxSMT problem. Instead, Subsection IV-B describes how hard and soft constraints are formulated to compose the MaxSMT problem.

A. *Inputs for the MaxSMT problem*

For the definition of the MaxSMT problem characterizing the proposed approach, only two inputs are required to be formulated by a human being, e.g., the security provider in charge of protecting a virtual network that has been already created or that should soon be deployed and activated.

The first input is the description of the *Service Graph* (SG) where firewalls and CPSs should be allocated. The SG represents the generalization of the *Service Function Chain* (SFC) concept, i.e., it shows how end points and intermediate middleboxes composing a virtual network are interconnected in the logical topology of the network. It is important to underline that in this SG no security function is present, but only functions which carry out networking operations such as network translation, logging or caching. This input is pre-processed so as to be transformed in another representation, which is called *Allocation Graph* (AG). In-between any pair of adjacent nodes of the SG, a placeholder virtual position, called *Allocation Place* (AP), is introduced in the corresponding AG. Each AP is a candidate position where a security function (i.e., a firewall or CPS instance) may be allocated. An example clarying how this pre-processing operation works is shown through Fig. 1 and Fig. 2, where the former represents a SG, while the latter depicts the AG derived from it.

The second input is composed of a set of NSPs, divided into two categories (i.e., connectivity policies and channel protection policies) according to the definition presented in Section III. All of them must be enforced in the AG derived from the input SG so that a correct configuration for the security service can be achieved. If one of them cannot be satisfied, then the configuration problem is not satisfiable.

B. *Formulation of the MaxSMT problem*

In the MaxSMT problem, formal models for network components, traffic flows and NSPs are employed. A modeling approach that has been recently published in literature [26] and was proved successful for the formal representation of virtual networks was adopted here as a starting point for the definition of these models. As long as since models are correct, MaxSMT guarantees that the computed solution, if any, is correct as well, through its intrinsic correctness-by-construction paradigm.

These formal models are composed of some variables and predicates that are left free, i.e., finding their interpretation is a task of the MaxSMT solver that is used to solve the configuration problem. These components represent the best way for formulating the allocation scheme and rule configuration for firewalls and CPSs, since they are not known a-priori and instead they are the output itself of the process. In light of this consideration, the definition of the allocation scheme is based on the *allocated* predicate, that can be applied to the APs of the AG topology. The value of this predicate, when applied to an AP, is established to be true if a firewall has been allocated in that AP in the final solution, to false otherwise. Similarly, for each possibly allocated security function, another predicate, called *configured*, is applied to each possible configuration rule, that in turn is composed of free variables. In this way, if the solver configures a rule for a firewall or a CSP, the *configured* predicate returns true when applied to that rule, and its variables are assigned with specific values by the solver itself (e.g., for the rule conditions of a firewall, the values of the IP 5-tuple are computed).

Then, the formal models are employed for the definition of both hard and soft clauses.

Hard clauses are used to express constraints that always require satisfaction. Therefore, they represent all the elements which may impact on the satisfaction of the NSPs and also on the generation of the rules. More specifically, the elements considered for the approach proposed in this paper are the following:

- the topological information of the virtual network, i.e., how the virtual functions and the APs introduced in the pre-processing phase of the SG are interconnected among them;
- the behavior of the service functions composing the input SG, because it may prohibit traffic flows to cross them, or it may redirect them on other paths thus making their original destination unreachable;
- the behavior of the security functions (i.e., firewalls and CPSs) that may be allocated in the AG. With respect to the already present service functions, the hard constraints modeling the behavior of the security functions are effectively evaluated by the solver only if the result of applying the *allocated* predicate on the AP where the security function may be placed is true. Otherwise, if it is

false, it means no security function has been put in that AP, which simply behaves as a forwarder;

- the packet transformation performed by middleboxes, because it may impact the rule configuration for the allocated firewalls or CPSs (e.g., if a firewall that must block a specific traffic identified by the condition of an input connectivity policy is established to be put after a NAT, then its filtering rule must be generated so as to consider that the NAT can potentially modify IP addresses);
- the enforcement of the connectivity and channel protection policies, expressing the security intents.

Instead, soft clauses are used to express optimization objectives. In this approach, two objectives have been defined, i.e., minimization of the allocated firewalls and CPSs on one side, minimization of the configured rules for the allocated security functions on the other side.

The first objective can be achieved by imposing that, for any AP a, the *allocated(a)* predicate should be false. Formula (1) expresses this concept, by using the notation $Soft(c, w)$, where c is the soft clause, whereas w is the weight assigned to the clause. It is clear that the situation where no security function is allocated is impossible. However, since the aim of a MaxSMT solver is to maximize the sum of the weights associated with satisfied soft clauses, then it will try to satisfy the maximum number of them (i.e., it will try to allocate the minimum number of functions), as long as all the hard constraints are fulfilled.

$$Soft(\neg allocated(a), w_a) \tag{1}$$

The second objective is similarly modeled. Formula (2) states that the optimal solution would be the one where no rule is configured in the allocated functions, i.e., where for each rule r the $configured(r)$ predicate is false. In turn, each rule r is composed of free variables, expressing the rule conditions and actions, whose values are determined by the solver at runtime only if the corresponding $configured(r)$ is true. Note that a rule does not always correspond to a single NSP. Since the process that is performed is policy refinement, it is possible that a rule may be enough to enforce multiple NSPs (e.g., if multiple hosts belonging to the same subnetwork must be blocked, a single rule may be configured on a packet filter), or that multiple rules are needed to enforce a single NSP (e.g., a traffic that must be encrypted and reach its destination would require a rule in a CPS working as channel opening, a rule in a CPS working as channel ending, and possibly some rules in firewalls in order to avoid blocking that traffic).

$$Soft(\neg configured(r), w_r) \tag{2}$$

After the solver is fed with all the hard and soft clauses, it searches for the optimal solution that satisfies all the hard constraints. From this solution, the values computed for the *allocated* predicate allows the user to understand in which places of the logical topology firewalls and CPSs should be allocated. Instead, from the values computed for the *configured* predicate and from the values of the free variables composing the rules the user can retrieve all the information required for configuring the deployed functions.

V. Conclusion and Future Works

This paper illustrates some preliminary ideas for a methodology which might automatically perform a joint configuration of the two most commonly used network security functions in virtual networks, i.e., packet filtering firewalls and channel protection systems. The need of such a methodology is motivated by the intrinsic relationships between the security properties these functions must enforce. Only when both of them are configured at the same time, it is possible to avoid any type of configuration anomaly, from simple sub-optimizations to critical inter-function conflicts.

Currently, a prototype framework implementing the approach illustrated in this paper is under development. For its implementation, a state-of-the-art MaxSMT solver developed by Microsoft Research, called Z3 [27], is used for the formulation and resolution of the MaxSMT problem representing the joint configuration problem. Then, the framework will be validated on use cases representing realistic production computer networks, with the aim to validate its efficacy and scalability with respect to the state of the art. Finally, if all these objectives are successfully achieved, a next future work that is planned is to integrate other types of network security functions (e.g., intrusion detection systems) in this joint automated methodology, so that a complete security service may be synthesized in a single shot.

References

[1] W. Yang and C. J. Fung, "A survey on security in network functions virtualization," in *IEEE NetSoft Conference and Workshops, NetSoft 2016, Seoul, South Korea, June 6-10, 2016*. IEEE, 2016, pp. 15–19.

[2] R. Boutaba and I. Aib, "Policy-based management: A historical perspective," *J. Netw. Syst. Manag.*, vol. 15, no. 4, pp. 447–480, 2007.

[3] J. Steinberger, A. Sperotto, H. Baier, and A. Pras, "Collaborative attack mitigation and response: A survey," in *IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, Ottawa, ON, Canada, 11-15 May, 2015*, R. Badonnel, J. Xiao, S. Ata, F. D. Turck, V. Groza, and C. R. P. dos Santos, Eds. IEEE, 2015, pp. 910–913.

[4] D. Bringhenti, F. Valenza, and C. Basile, "Toward cybersecurity personalization in smart homes," *IEEE Secur. Priv.*, vol. 20, no. 1, pp. 45–53.

[5] A. El-Hassany, P. Tsankov, L. Vanbever, and M. T. Vechev, "Netcomplete: Practical network-wide configuration synthesis with autocompletion," in *15th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2018, Renton, WA, USA, April 9-11, 2018*, 2018, pp. 579–594.

[6] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, "Automated firewall configuration in virtual networks," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 2, pp. 1559–1576, 2023.

[7] M. Rossberg, G. Schaefer, and T. Strufe, "Distributed automatic configuration of complex ipsec-infrastructures," *J. Network Syst. Manage.*, vol. 18, no. 3, pp. 300–326, 2010.

[8] D. Bringhenti, G. Marchetto, R. Sisto, and F. Valenza, "Short paper: Automatic configuration for an optimal channel protection in virtualized networks," in *Proceedings of the 2nd Workshop on Cyber-Security Arms Race, colocated with ACM CCS 2020*. ACM, 2020, p. 25–30.

[9] Y. Bartal, A. J. Mayer, K. Nissim, and A. Wool, "Firmato: A novel firewall management toolkit," in *1999 IEEE Symposium on Security and Privacy, Oakland, California, USA, May 9-12, 1999*, 1999, pp. 17–31.

[10] P. Verma and A. Prakash, "FACE: A firewall analysis and configuration engine," in *2005 IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2005), 31 January - 4 February 2005, Trento, Italy*, 2005, pp. 74–81.

[11] J. D. Guttman and A. L. Herzog, “Rigorous automated network security management,” *Int. J. Inf. Sec.*, vol. 4, no. 1-2, pp. 29–48, 2005.

[12] D. Ranathunga, M. Roughan, P. Kernick, and N. Falkner, “The mathematical foundations for mapping policies to network devices,” in *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016), Lisbon, Portugal, July 26-28, 2016*, 2016, pp. 197–206.

[13] A. El-Hassany, P. Tsankov, L. Vanbever, and M. T. Vechev, “Network-wide configuration synthesis,” in *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II*, 2017, pp. 261–281.

[14] E. Karafili, F. Valenza, Y. Chen, and E. C. Lupu, “Towards a framework for automatic firewalls configuration via argumentation reasoning,” in *NOMS 2020 - IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, April 20-24, 2020.* IEEE, 2020, pp. 1–4.

[15] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, “Automated optimal firewall orchestration and configuration in virtualized networks,” in *NOMS 2020 - IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, April 20-24, 2020.* IEEE, 2020, pp. 1–7.

[16] D. Bringhenti, J. Yusupov, A. M. Zarca, F. Valenza, R. Sisto, J. B. Bernabé, and A. F. Skarmeta, “Automatic, verifiable and optimized policy-based security enforcement for sdn-aware iot networks,” *Comput. Networks*, vol. 213, p. 109123, 2022.

[17] A. K. Bandara, A. C. Kakas, E. C. Lupu, and A. Russo, “Using argumentation logic for firewall configuration management,” in *11th IFIP/IEEE International Symposium on Integrated Network Management, Long Island, NY, USA, June 1-5, 2009*, 2009, pp. 180–187.

[18] P. Bera, S. K. Ghosh, and P. Dasgupta, “Policy based security analysis in enterprise networks: A formal approach,” *IEEE Trans. Network and Service Management*, vol. 7, no. 4, pp. 231–243, 2010.

[19] S. Maity, P. Bera, and S. K. Ghosh, “Policy based ACL configuration synthesis in enterprise networks: A formal approach,” in *International Symposium on Electronic System Design, ISEDs 2012, Kolkata, India, December 19-22, 2012*, 2012, pp. 314–318.

[20] Z. Fu and S. F. Wu, “Automatic generation of ipsec/vpn security policies in an intra-domain environment,” in *Operations & Management, 12th International Workshop on Distributed Systems, DSOM 2001, Nancy, France, October 15-17, 2001. Proceedings*, 2001, pp. 279–290.

[21] Y. Yang, C. U. Martel, and S. F. Wu, “On building the minimum number of tunnels: an ordered-split approach to manage ipsec/vpn policies,” in *IEEE/IFIP Network Operations and Management Symposium, Seoul, Korea, 19-23 April 2004*, 2004, pp. 277–290.

[22] C. Chang, Y. Chiu, and C. Lei, “Automatic generation of conflict-free ipsec policies,” in *Formal Techniques for Networked and Distributed Systems - FORTE 2005, 25th IFIP WG 6.1 International Conference, Taipei, Taiwan, October 2-5, 2005, Proceedings*, 2005, pp. 233–246.

[23] J. García-Alfaro, F. Cuppens, N. Cuppens-Boulahia, and S. Preda, “MIRAGE: A management tool for the analysis and deployment of network security policies,” in *Data Privacy Management and Autonomous Spontaneous Security - 5th International Workshop, DPM 2010 and 3rd International Workshop, SETOP 2010, Athens, Greece, September 23, 2010, Revised Selected Papers*, 2010, pp. 203–215.

[24] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, “Towards a fully automated and optimized network security functions orchestration,” in *2019 4th International Conference on Computing, Communications and Security (ICCCS), Rome, Italy, October 10-12, 2019*, 2019, pp. 1–7.

[25] C. Basile, D. Canavese, A. Lioy, and F. Valenza, “Inter-technology conflict analysis for communication protection policies,” in *Risks and Security of Internet and Systems - 9th International Conference, CRiSIS 2014, Trento, Italy, August 27-29, 2014, Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 8924. Springer, 2014, pp. 148–163.

[26] D. Bringhenti, G. Marchetto, R. Sisto, S. Spinoso, F. Valenza, and J. Yusupov, “Improving the formal verification of reachability policies in virtualized networks,” *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 1, pp. 713–728, 2021.

[27] L. M. de Moura and N. Bjørner, “Z3: an efficient SMT solver,” in *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, 2008, pp. 337–340.