Short Paper: Automatic Configuration for an Optimal Channel Protection in Virtualized Networks

Short Paper: Automatic Configuration for an Optimal Channel Protection in Virtualized Networks

Daniele Bringhenti
Politecnico di Torino, Dip. Automatica e Informatica
Torino, Italy
daniele.bringhenti@polito.it

Guido Marchetto
Politecnico di Torino, Dip. Automatica e Informatica
Torino, Italy
guido.marchetto@polito.it

Riccardo Sisto
Politecnico di Torino, Dip. Automatica e Informatica
Torino, Italy
riccardo.sisto@polito.it

Fulvio Valenza
Politecnico di Torino, Dip. Automatica e Informatica
Torino, Italy
fulvio.valenza@polito.it

Abstract

Data confidentiality, integrity and authentication are security properties which are often enforced with the generation of secure channels, such as Virtual Private Networks, over unreliable network infrastructures. Traditionally, the configuration of the systems responsible of encryption operations is performed manually. However, the advent of software-based paradigms, such as Software-Defined Networking and Network Functions Virtualization, has introduced new arms races. In particular, even though network management has become more flexible, the increased complexity of virtual networks is making manual operations unfeasible and leading to errors which open the path to a large number of cyber attacks. A possible solution consists in reaching a trade-off between flexibility and complexity, by automatizing the configuration of the channel protection systems through policy refinement. In view of these considerations, this paper proposes a preliminary study for an innovative methodology to automatically allocate and configure channel protection systems in virtualized networks. The proposed approach would be based on the formulation of a MaxSMT problem and it would be the first to combine automation, formal verification and optimality in a single technique.

CCS Concepts • **Security and privacy** → *Formal security models*; *Security protocols*; *Privacy-preserving protocols*; • **Networks** → *Security protocols*; *Network management*.

Keywords channel protection, network security optimization, network functions virtualization, automation

ACM Reference Format:
Daniele Bringhenti, Guido Marchetto, Riccardo Sisto, and Fulvio Valenza. . Short Paper: Automatic Configuration for an Optimal Channel Protection in Virtualized Networks. In . ACM, New York, NY, USA, 8 pages.

, ,
.

1 Introduction and Motivation

The introduction of virtualization paradigms, such as *Network Functions Virtualization* (NFV) and *Software-Defined Networking* (SDN), has risen novel arms races for network cybersecurity. On one side, these paradigms increased flexibility in network management: for instance, virtual functions can be rapidly turned on and set up, with respect to what a hardware device was used to require in the past. On the other side, they determined an increase of the networks' size and complexity, with a serious impact on the protection of the data transmitted on the networks.

More specifically, the security properties that have been impacted are the ones representing the security model called CIA triad (i.e., confidentiality, integrity and authentication), whose aim is to avoid cyberattacks capable of intercepting or altering the conveyed information. These properties are often enforced by the creation of secure channels. Some of the most common solutions are IPSec-based *Virtual Private Networks* (VPNs), TLS and SSH tunnels, or channels created with protocols such as PPTP and L2TP. These channels are created using shared or untrusted transmission infrastructures, but the data transmitted on them respect the CIA requirements because the *Channel Protection Systems* (CPSs, that are the devices at the channel's borders) apply security mechanisms such as the computation of MACs or digital signatures before sending the data in the channel. Albeit conceptually easy, these techniques are nevertheless becoming more troublesome in virtual computer networks. In such a dynamic environment, it is arduous to keep the full control on the all channels created for communication protection and to ensure that they have been properly configured.

The main problem behind the difficulty of these operations is that both the decisions where the CPSs should be placed and how they should be configured are personally taken by human beings. The fallibility of a manual security configuration is inevitable in complex networks where an administrator struggles to have a complete overview, thus opening the path to cyber attacks [16]. This claim finds proof

in the most recent Verizon's Data Breach Investigation Report [24], according to which misconfiguration of security functions due to manual operations represents the most exploited error category for data breaches.

In this arms race, nevertheless, a trade-off between advantages and drawbacks of network virtualization can be achieved. Even though virtualization has introduced complexity in the management of channel protection, on the other hand its dynamism could be efficaciously exploited to avoid human errors in the security configuration. In fact, a feasible solution to the problem would be to automatize the security configuration for the channel protection, replacing human operations. However, limited research has been carried out about this prospect.

In light of these considerations, this paper proposes a preliminary study of an automated methodology for establishing the allocation scheme of CPSs in a virtualized network and computing their full configuration. The outcome would be reached by means of a policy refinement operation, where the security requirements expressed by the user are refined into the configuration rules for each CPS. Formal verification based on a correctness-by-construction approach and optimality criteria such as minimization of allocated functions and of configured rules would be fulfilled by formulating the problem as *Maximum Satisfiability Modulo Theories* (MaxSMT).

Therefore, a novel contribution of this paper is that this formulation would lead to a combination of three main features – i.e., automation, formal verification and optimality – in a single technique, an achievement that has still not be reached for channel protection, to the best of our knowledge. Another contribution is the feasibility of this methodology for virtual networks, as a means of reaching a trade-off between their complexity and dynamism.

The remainder of this paper is structured as follows. Section 2 describes other works related to the automatic security configuration, with a specific focus on channel protection. Section 3 states the challenged problem. Section 4 illustrates the approach which is proposed to overcome the limitations of other methodologies, with a focus on the optimization objectives which are pursued. Finally, Section 5 briefly draws conclusions and describes future works in this research activity.

2 Related Works

Automation for network security configuration has been currently mostly investigated for firewalls. The main reasons are that on one side firewalls are more used than channel protection systems in networks even though they enforce different security properties, on the other side the channel protection is much more difficult to be dealt with. The first works ([2], [25], [13]) proposed firewall management methodologies which could be exclusively applied to networks where any security function was a hardware appliance. Later, formal verification techniques have been exploited to provide correctness assurance after the automated computation of firewall configurations ([12], [5], [17], [22], [1]). Then, after the advent of softwarization in networking, this research path has found new relevance ([20], [8], [9], [6]) and currently has become an important research trend in network security.

Research about the automatic configuration for channel protection, instead, has been quite more limited and it has exclusively focused on IPSec-based VPNs. The milestone in this area is represented by [10], which proposes three different algorithms for the creation of VPN tunnels to fulfill the user requirements: 1) a direct approach, where for each request a separate tunnel is created; 2) a bundle approach, where the traffic flows interested by the requirements are grouped and for each set a single tunnel is generated, providing completeness at expense of speed; 3) a combined approach, as trade-off of the previous two. A fourth strategy has been later illustrated by the same authors in [27]: in this so-called ordered-split approach, not only a correct solution is computed, but the optimal one is chosen with regards to minimizing the number of required VPN tunnels. In these two works, however, the environment that has been considered is intra-domain. An extension, described in [26], is represented by a negotiation protocol through which gateways of different Autonomous Systems can negotiate the automatic generation of the tunnels in an inter-domain scenario. Other relevant approaches are presented in [7], where a heuristic algorithm is used to generate the tunnel starting from the longest one in an iterative way, and [21], whose additional features with respect to other works are high scalability, agility and robustness.

Some research papers ([14] [11] [4]) propose an automatic configuration of a full security architecture, which can cover multiple security properties at the same time, including the CIA triad. However, their analysis on systems specifically focused for channel protection is limited with respect to other function types. Besides, most of the problems arising in virtualized networks and with an impact on channel protection are not addressed.

In conclusion, from the analysis of the related works, it emerges that almost no automated technique proposed for channel protection is either enriched by formal verification, or with optimality criteria (with the exception of [27]); moreover, most of them are not specifically designed for virtualized networks. Consequently, the combination of features which would characterize our methodology represents a central novelty in literature.

3 Problem Statement

The configuration of security devices oriented to channel protection, in this paper called *Channel Protection Systems*

(CPSs), is fundamental to protect critical assets from disclosures that would lead to privacy or intellectual property violation or, in a worse case, huge monetary loss. These security requirements and goals are often formulated by means of *Channel Protection Policies* (CPPs), in the growing trend of policy-based management for network security. In particular, each CPP represents the user-specified policy describing how channel protection must be enforced. An example is the following: "All the traffic generated by hosts in the subnetwork 124.56.10.0/24 and headed for the web server 88.40.12.2 must be protected with confidentiality and integrity. These security properties must be enforced by applying AES-256-GCM and SHA-256 and must be present when the traffic crosses any node not belonging to the networks either 124.56.10.0/24 or 88.40.0.0/16". This example clarifies that a CPP does not specify which technology must be used for channel protection, or the number of CPSs and tunnels which must be created to enforce it.

The enforcement of the CPPs requires the administrator to perform two tasks:

- choosing the technology or protocol (e.g. IPSec, TLS, SSH) to use for channel protection, and deciding where to allocate the required CPSs in the network;
- writing the configuration rules for each deployed CPS, in order to establish when a channel must be effectively created.

Traditionally, these tasks are performed manually by network administrators. If the complexity of these operations was already high when performed on traditional hardware-based networks, the advent of virtualization further increased it [19]. A manual refinement of CPPs is consequently getting unfeasible and unbearable for human beings. The remainder of this section will address and deeply explain this statement. In particular, Subsection 3.1 describes the problems which arise in virtual environments for a manual configuration of channel protection. Then, Subsection 3.2 illustrates the negative consequences of those issues. Finally, Subsection 3.3 introduces our proposal to overcome the problems.

3.1 Problems for a manual configuration

When a secure channel must be created to enforce the CIA security properties for a kind of traffic, the administrator must choose the technological implementation that is most suitable for the specific situation. With the advent of virtualization, the landscape of available solutions has enormously expanded [3]. New types of CPSs can be easily implemented as software programs, instead of being built as hardware boxes, and they can work on the basis of newly developed algorithms or protocols. In light of this scenario, a manual choice of the best solution, or even of the correct solution, is not trivial. The administrator should identify the security capabilities that are expressed in all the CPPs and then select a suitable solution. This task is indeed burdensome and requires non-negligible effort.

Additionally, virtual networks are highly dynamic and ever-changing environments. A configuration that was set up for channel protection in traditional networks might endure for a long period of time before requiring any change. Instead, the dynamism brought over by virtualization demands continuous reconfiguration of the security systems. Possible reasons might be the high frequency by which an NFV controller changes the IP addresses and ports of the deployed virtual services, or the mitigation of incoming attacks detected thanks to innovative softwarized systems based on artificial intelligence algorithms. This is a property that virtualized systems inherited from dynamic topologies, which change accordingly the evolving context. In such an environment, the rapidity required in the reaction might result in accidental misconfiguration errors, with a severe impact on the security of the network service.

Another problem deriving from the dynamism which characterizes virtual environments is that the complexity of the networks where protection channel must be enforced is constantly increasing as well. The dimension of virtual networks is increasingly getting bigger, more complex functions are created (e.g., stateful middleboxes), new kinds of cyber-attacks (e.g., side channel attacks across virtual machines [19]) are starting to exploit the vulnerabilities shown by virtualization. All these factors must be accounted for the configuration of channel protection. Nonetheless, it is clear that a human being would have great difficulties in achieving a configuration that is correct with respect of all the external influencing elements.

3.2 Consequences of the problems

The illustrated problems might lead to undesired consequences when they afflict the two main tasks, which an administrator is in change of, for the configuration of a secure channel: allocation of the security systems and computation of their rules.

Firstly, an incorrect or sub-optimized decision about where the CPSs should be allocated in a *Service Graph* (SG), logical description of a virtualized network topology, can lead to unwanted consequences from different points of view. On one side, if a redundant number of virtual functions are deployed to enforce the CPPs, available physical resources (e.g. CPU usage, random access memory, hard disk memory) dramatically decrease, hardening the management of other function types. Moreover, another negative consequence would be that the overhead largely impacts on the amount of traffic and service type (i.e., encryption, hashing, digital signature) that must be managed by the CPSs. On the other side, depending on where the systems are positioned (i.e., where the security algorithms are applied), the set of network nodes where the traffic flow passes without protection is different.

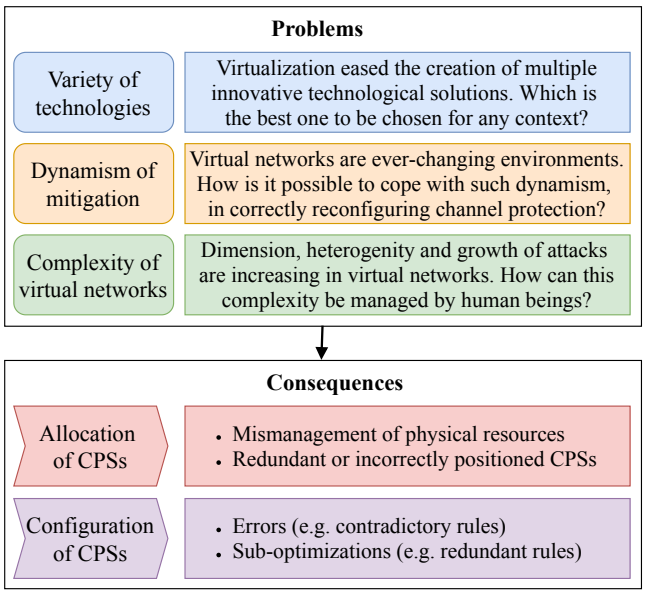

Figure 1. Main problems and their consequences

If a channel is created too far from the source, for example, it is possible that in the path towards the border system the traffic could have been already inspected or modified.

Then, assuming a correct allocation scheme has been established, the rules composing the configuration of the CPSs must be computed. In this operation, however, the probability of introducing anomalies is directly proportional to the size and complexity of the network [15]. An anomaly is, in particular, an error, a sub-optimization or a conflict arising in configuring a single gateway (intra-gateway anomaly) or multiple ones (inter-gateway anomaly). Several anomaly examples can be cited: a rule is shadowed if it is never applied, is redundant if another rule exists and enforces the same properties on the same traffic kind. An empirical study [23] recently showed that out of 30 administrators the percentage of them that created at least one anomaly (i.e., a conflict in the rules configured in the CPSs) in channel protection policies is 93.33%, which is astonishingly high. If only administrators with high level of expertise are considered, the percentage goes down only to 90% because, even though they make a limited number of errors, their decisions are still prone to sub-optimizations. These experimental results are motivated by the intrinsic complexity of the CPPs and the heterogeneity of implementation technologies or security protocols through which secure channels can be created.

An infographic schema, summarizing the problems arising for a manual channel protection in virtual networks and their consequences, is illustrated in Figure 1. This overview underlines how a manual configuration for channel protection is not feasible anymore, and research is needed to pursue innovative ways to deal with the problems that emerged with the virtualization of networks.

3.3 How to challenge the problems

Virtualization has introduced not only complexity in the networking world, but also the agility and dynamism that is essential to face cyber attacks [18]. In this trade-off, a potential idea is to use the offered agility to automatize both the design of the allocation scheme for CPSs and the computation of their configuration rules. Following the main principles of policy-based management, a network administrator should only specify the policies describing the security requirements and goals to be enforced in the virtual service. The refinement of those policies into concrete security configurations would not be manually managed by human beings, but by automatic tools. Besides, if these tools are enhanced with formal verification techniques such as model checking or correctness-by-construction, then a formal guarantee for the correctness of the configuration problem can be provided - something that would be impossible if performed manually.

In light of all these considerations, automation is consequently a key feature to mitigate the difficulties that emerge when facing these two problems, because it can overcome all the limitations of a manual configuration and increase the confidence level in the designed security service. For this reason, the methodology that we propose and that is described in Section 4 aims to automatically compute both the optimal and formally correct allocation scheme and configuration of CPSs in virtual networks in order to fulfill the requested CPPs.

4 The Proposed Approach

The formal and optimized approach followed in the methodology proposed in this paper is based on the formulation of a MaxSMT problem. It represents an optimization-enhanced version of the SMT problem[1], characterized by two kinds of clauses. The hard constraints must be always fulfilled, whereas the soft ones do not strictly require satisfaction to achieve a correct solution. Instead, each soft clause is given a weight and the goal is to maximize the sum of the weights assigned to the satisfied soft clauses. *First-Order Logic* (FOL) has been used for the definition of the models on which the MaxSMT's constraints are based. This formulation additionally enables an optimized correctness-by-construction approach: some of the variables are left open, so that their correct and optimal values are established by the problem solver, without requiring a-posteriori either verification or optimization.

According to this formulation, the methodology we are proposing in this paper is shaped as the workflow shown in Figure 2. The optimization engine is the MaxSMT solver

[1]Differently from a traditional SAT problem, in an SMT problem not only the Boolean theory, but also additional theories such as integers, bit-vectors or strings can be used. These theories comes in handy for modeling networking properties such as IP addresses.

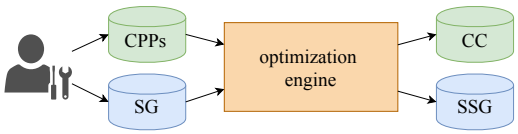

Figure 2. Workflow of the automated methodology

which computes the optimal result with respect to the required constraints. As illustrated, the two inputs introduced by a human being are a SG and a set of CPPs. After elaborating the received information, if at least a correct solution can be computed, the engine generates a *Security Service Graph* (SSG), that is the original SG enriched with the allocated CPSs, and the *CPS Configuration* (CC), inclusive of all the rules to configure on the respective devices. In case no correct solution can be achieved, a not-enforceability report is instead generated to motivate the problem unfeasibility (e.g., conflict in CPPs' formulation). This report can be read by a human being, who can accordingly modify the inputs so that a correct solution can be reached in a next run.

In Subsection 4.1, we describe how the output is computed by the MaxSMT solver in compliance with the hard constraints coming from the inputs representing the virtual topology and the CPPs' formulation. Then, in Subsection 4.2 the formalization of the optimization objectives as soft constraints is illustrated.

4.1 Constraint-based policy refinement

The result computation is performed by the engine with a *constraint-based policy refinement*. This operation is constraint-based because the decisions are bounded to the respect of some hard clauses, whereas it is a policy refinement because there is no a one-to-one mapping between the user-specified policies and the CPS configuration. The first motivation is that, since the security architecture designed for channel protection by this methodology is distributed, it is not needed that each policy is enforced by all the CPSs. Secondly, if the entire policy set was configured on each device, the achieved solution would be deeply unoptimized. In view of these considerations, in the following we will describe the hard constraints which the solver must fulfill to reach a formally correct solution.

Firstly, the SG is represented in the MaxSMT problem as a directed graph, where each vertex represents either a communication end point or a service function (e.g., middleboxes like load balancer or web cache, which cannot enforce security properties), while each link is the interconnection between a pair of vertices. The information provided by the elements in the SG is captured with the hard constraints. The behavior of each service function and the paths formed by the topology links shape how the packet flows crossing the network are managed and which paths they can follow to reach the destination. For example, a NAT can block an external communication if a state relative to a previous communication opened from the internal shadowed network does not exist: this possible event, which could lead to the unfeasibility of some problem instances, must be taken into account with a hard clause.

The CPPs instead describe which security properties must be enforced in the generation of the secure tunnels. In particular, each CPP is characterized by the following elements:

- the set of untrusted middleboxes, where safety is not guaranteed for the traffic flow to protect;
- the set of inspector middleboxes, where the crossing traffic must be plain, so that it can be analyzed by them (note that inspector middleboxes cannot be untrusted for a given CPP);
- the condition set representing the characteristics which enable the identification of the traffic flows to protect. Each condition is, in particular, defined as a tuple, where the first element is a packet field (e.g., source/destination IP address/port), the second is either the specific value the field can assume or the range where the field value falls into. This set includes conditions on the source IP address and on the destination IP address for the traffic flow to protect. Note that, given a specific condition set, multiple flows could satisfy these conditions: for example, if a condition with "IPSrc" as field and "124.12.2.0/24" as value is specified, all the packets with a source address in this range match the condition;
- the security properties to be applied on the matching flows. Each security property is characterized by a type (confidentiality, authentication or integrity), the algorithms to be applied for its enforcement (e.g., AES-128-CBC, HMAC-SHA-512), the required key length.

This formulation is thus general enough to be applied not only to IPSec-based VPNs, but also to other channel protection protocols working at different levels of the ISO/OSI stack.

The enforcement of the CPPs is applied through some hard constraints expressed for each CPP:

1. at least two CPSs must be allocated in the SG (the first before the list of untrusted middleboxes, the second after) for each flow starting from the CPP's source, headed for the CPP's destination and matching the CPP's conditions;
2. when the flows cross the untrusted middleboxes, they must be characterized by the security properties expressed by the CPP;
3. when the flows cross the inspector middleboxes, they must be composed by plain traffic without encryption features.

It is worth underlining that the presence of these hard constraints reduces the solution space. On one side, a correct solution for the configuration problem could be reached faster, because the solver must analyze a smaller set of valid

solutions to identify the best one. On the other side, a possible outcome might be the absence of a correct solution for the problem.

In the hard constraints used for modeling the SG and the CPPs, some variables are left open, i.e., without a predetermined value. At the same time, the output of some predicates applied to model elements is not constraint-bound, in accordance with the correctness-by-construction principle. This decision is explained by the fact that in both cases their value will be established by the problem solver and will be exclusively set in the output model. In particular, these open components of the model are related to the allocation of the CPSs on one side, to their configuration rules on the other.

First, about these components, each SG's link is a potential candidate position for the allocation of a CPS. This statement is formally expressed with the *allocate* predicate, applied to any link of the topology. The Boolean output of the predicate is not forced in the problem constraints, but it will be established by the solver as result. Then, if *allocate* is set true for a SG's link, then other variables related to the configuration of the CPS will be filled by the solver. These problem elements must be provided in advance, though. Each rule is then effectively configured only if needed. For this purpose, the $configure$ predicate states if the rule of a CPS on which it is applied is effectively configured. In positive case, the values of the rule fields are assigned to the open variables.

4.2 Optimization

The hard constraints, the open variables, and the predicates presented so far are not enough to reach an optimal solution, but only a correct one. Consequently, the formulation of some soft constraints is required to add optimization to the other two features (automation and formal verification), already provided by the hard clauses. We adopt the $Soft(c, w)$ notation for the representation of a soft clause: c represents the constraint, while w is the assigned weight.

The first optimization objective is to minimize the number of allocated CPSs in the SG in order to reduce resource consumption. This goal is achieved by some soft clauses requiring that, whenever possible in compliance with the satisfiability of the hard constraints, a CPS should not be installed. The formal representation of this class of soft clauses is shown in (1), which is valid for each SG's link, referred to with the l letter. It is evident that this clause cannot be satisfied for any SG's link; nevertheless, as long as the assigned weights are positive, the problem solver will attempt to satisfy the maximum number of these soft clauses.

$$Soft(\neg allocate(l), w_l) \tag{1}$$

The second optimization objective is, instead, to minimize the number of rules configured in the allocated CPSs to enforce all the CPPs. The purpose is to improve the efficiency of the security operations: if the rule set is redundant, the device must analyze more rules than what is strictly necessary, with a potentially significant impact on the efficiency. In view of this consideration, focusing on each SG's link where a CPS is tentatively allocated, a soft clause shown in (2) is defined for each possible rule, represented with the r letter. The best situation that this set of soft clause wishes for is that each rule is not configured. With this approach, only the soft constraints related to the rules needed for the effective enforcement of the CPPs are falsified by the solver. For these kind of rules, additionally, the solver must altogether establish the values of the rule fields, by assigning them to the open variables originally introduced in the input model.

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

Note that the first objective has higher priority than the second. Actually, minimizing the number of allocated CPSs can contribute to indirectly reduce the cardinality of their rule sets, since more rules could be aggregated in the same device. This constraint is respected by imposing that, for each SG's link, the weight assigned to the soft clause (1) is higher than the sum of the weights assigned to the related clauses (2), as represented in (3).

$$\sum_r (w_r) < w_l \tag{3}$$

5 Conclusion and Future Works

In this paper, we illustrated a preliminary study for a novel methodology to automatize the orchestration and configuration of a distributed security architecture for channel protection. The goal is to avoid the typical errors that a manual configuration is prone to, by reaching a trade-off between flexibility and complexity introduced by network virtualization. The proposed approach is based on a MaxSMT formulation. It is thus the first approach that, to the best of our knowledge, would combine automation, optimization and formal verification for the configuration of channel protection systems.

Currently, we are completing the formulation of the methodology described in this paper and we are implementing it as a Java framework, where the optimization engine of z3 theorem prover is used as MaxSMT solver. This implementation will enable the interaction with NFV and cloud orchestrators, like Open Source MANO or Kubernetes, so that the automatically computed virtual SSG can be deployed on the servers of the underlying physical infrastructure. As additional future work, we are planning to extensively test the framework in order to prove its feasibility in topologies representing computer networks of current generation. Moreover, a next extension would be the possibility to automatically allocate and configure other kinds of network security functions, such as intrusion detection systems and deep packet inspectors, thus enabling the orchestration of heterogeneous security services.

Acknowledgments

This work has been partially supported by the EU H2020 Projects ASTRID (Grant Agreement no. 786922) and CyberSec4Europe (Grant Agreement no. 830929).

References

[1] Arosha K. Bandara, Antonis C. Kakas, Emil C. Lupu, and Alessandra Russo. 2009. 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*. 180–187. https://doi.org/10.1109/INM.2009.5188808

[2] Yair Bartal, Alain J. Mayer, Kobbi Nissim, and Avishai Wool. 2004. *Firmato*: A novel firewall management toolkit. *ACM Trans. Comput. Syst.* 22, 4 (2004), 381–420. https://doi.org/10.1145/1035582.1035583

[3] Cataldo Basile, Daniele Canavese, Antonio Lioy, and Fulvio Valenza. 2014. 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 (Lecture Notes in Computer Science)*, Vol. 8924. Springer, 148–163. https://doi.org/10.1007/978-3-319-17127-2_10

[4] Cataldo Basile, Fulvio Valenza, Antonio Lioy, Diego R. Lopez, and Antonio Pastor Perales. 2019. Adding Support for Automatic Enforcement of Security Policies in NFV Networks. *IEEE/ACM Trans. Netw.* 27, 2 (2019), 707–720. https://doi.org/10.1109/TNET.2019.2895278

[5] Padmalochan Bera, Soumya Kanti Ghosh, and Pallab Dasgupta. 2010. Policy Based Security Analysis in Enterprise Networks: A Formal Approach. *IEEE Trans. Network and Service Management* 7, 4 (2010), 231–243. https://doi.org/10.1109/TNSM.2010.1012.0365

[6] Daniele Bringhenti, Guido Marchetto, Riccardo Sisto, Fulvio Valenza, and Jalolliddin Yusupov. 2020. 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, 1–7. https://doi.org/10.1109/NOMS47738.2020.9110402

[7] Chi-Lan Chang, Yun-Peng Chiu, and Chin-Laung Lei. 2005. 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*. 233–246. https://doi.org/10.1007/11562436_18

[8] Ahmed El-Hassany, Petar Tsankov, Laurent Vanbever, and Martin T. Vechev. 2017. Network-Wide Configuration Synthesis. In *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II*. 261–281. https://doi.org/10.1007/978-3-319-63390-9_14

[9] Ahmed El-Hassany, Petar Tsankov, Laurent Vanbever, and Martin T. Vechev. 2018. 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*. 579–594. https://www.usenix.org/conference/nsdi18/presentation/el-hassany

[10] Zhi Fu and Shyhtsun Felix Wu. 2001. 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*. 279–290.

[11] Joaquín García-Alfaro, Frédéric Cuppens, Nora Cuppens-Boulahia, and Stere Preda. 2010. 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*. 203–215. https://doi.org/10.1007/978-3-642-19348-4_15

[12] John Govaerts, Arosha K. Bandara, and Kevin Curran. 2008. A formal logic approach to firewall packet filtering analysis and generation. *Artif. Intell. Rev.* 29, 3-4 (2008), 223–248. https://doi.org/10.1007/s10462-009-9147-0

[13] Joshua D. Guttman. 1997. Filtering Postures: Local Enforcement for Global Policies. In *1997 IEEE Symposium on Security and Privacy, May 4-7, 1997, Oakland, CA, USA*. 120–129. https://doi.org/10.1109/SECPRI.1997.601327

[14] Joshua D. Guttman and Amy L. Herzog. 2005. Rigorous automated network security management. *Int. J. Inf. Sec.* 4, 1-2 (2005), 29–48. https://doi.org/10.1007/s10207-004-0052-x

[15] Hazem H. Hamed, Ehab S. Al-Shaer, and Will Marrero. 2005. Modeling and Verification of IPSec and VPN Security Policies. In *13th IEEE International Conference on Network Protocols (ICNP 2005), 6-9 November 2005, Boston, MA, USA*. 259–278. https://doi.org/10.1109/ICNP.2005.25

[16] Wolfgang John, Guido Marchetto, Felician Németh, Pontus Sköldström, Rebecca Steinert, Catalin Meirosu, Ioanna Papafili, and Kostas Pentikousis. 2017. Service Provider DevOps. *IEEE Commun. Mag.* 55, 1 (2017), 204–211. https://doi.org/10.1109/MCOM.2017.1500803CM

[17] Soumya Maity, Padmalochan Bera, and S. K. Ghosh. 2012. 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*. 314–318. https://doi.org/10.1109/ISED.2012.72

[18] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck, and Raouf Boutaba. 2016. Network Function Virtualization: State-of-the-Art and Research Challenges. *IEEE Communications Surveys and Tutorials* 18, 1 (2016), 236–262. https://doi.org/10.1109/COMST.2015.2477041

[19] Rajendra Patil and Chirag Modi. 2019. An Exhaustive Survey on Security Concerns and Solutions at Different Components of Virtualization. *ACM Comput. Surv.* 52, 1 (2019), 12:1–12:38. https://doi.org/10.1145/3287306

[20] Dinesha Ranathunga, Matthew Roughan, Phil Kernick, and Nick Falkner. 2016. 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*. 197–206. https://doi.org/10.5220/0005946201970206

[21] Michael Rossberg, Guenter Schaefer, and Thorsten Strufe. 2010. Distributed Automatic Configuration of Complex IPsec-Infrastructures. *J. Network Syst. Manage.* 18, 3 (2010), 300–326. https://doi.org/10.1007/s10922-010-9168-7

[22] Nicolas Stouls and Marie-Laure Potet. 2007. Security Policy Enforcement Through Refinement Process. In *7th International Conference of B Users, Besançon, France, January 17-19, 2007, Proceedings*. 216–231. https://doi.org/10.1007/11955757_18

[23] F. Valenza, C. Basile, D. Canavese, and A. Lioy. 2017. Classification and Analysis of Communication Protection Policy Anomalies. *IEEE/ACM Trans. Netw.* 25, 5 (Oct 2017), 2601–2614. https://doi.org/10.1109/TNET.2017.2708096

[24] Verizon. 2020. 2020 Data Breach Investigations Report.

[25] Pavan Verma and Atul Prakash. 2005. 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*. 74–81. https://doi.org/10.1109/SAINT.2005.28

[26] Yanyan Yang, Zhi (Judy) Fu, and Shyhtsun Felix Wu. 2003. BANDS: An Inter-domain Internet Security Policy Management System for IPSec/VPN. In *IFIP/IEEE Eighth International Symposium on Integrated Network Management (IM 2003), March 24-28, 2003, Colorado Springs, USA*. 231–244.

[27] Yanyan Yang, Charles U. Martel, and Shyhtsun Felix Wu. 2004. 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. 277–290. https: //doi.org/10.1109/NOMS.2004.1317665