Network-Security-Policy Analysis

Christian Pitscheider

Dip. di Automatica e Informatica
Politecnico di Torino
Torino, Italy
christian.pitscheider@polito.it

***Abstract*—Computer network security is the first line of defence to accomplish information assurance. The computer network is at risk without a well-designed and flawless implemented network security policy. The main problem is that network administrators are not able to verify the network security policy. Although further research has been carried out, it mainly concerns small specific parts of the overall problem. This paper presents different approaches from literature and highlights how they are correlated and can operate together. This work summarizes the solutions proposed in literature, points out their advantages, disadvantages and limitations. To conclude, it proposes solutions for future research in this area.**

Keywords–Security Policy; Analysis; Reachability; Policy comparison.

I. INTRODUCTION

More and more computer networks are connected to the Internet and remote sites are becoming more frequent. As a result, computer networks have become a very complex structure that is hard to manage. As some studies have shown, firewall configuration errors are quite frequent [1][2]. The studies point out that network administrators do not have a good insight of the network and its configurations. Dedicated tools and procedures are needed to support the daily work of network administrators.

In literature, different approaches exist to help network administrators in their daily workflows. In general, the approaches can be divided into two distinct categories: policy analysis and policy generation. Policy analysis focuses on existing and deployed configurations, while policy generation focuses on automatically generating new configurations.

This survey gives a brief overview of policy generation but its main focus is on policy analysis, and in particular on three distinct policy analysis categories, namely, Conflict analysis, Reachability analysis, and Policy comparison.

The main limitation of the papers concerning policy analysis is that they focus on one single type of security control and cannot be applied to a complex computer network. For example, Al-Shear proposes a solution to perform conflict analysis of firewall policies [3]; however, this solution is not able to model Network Address Translation (NAT) / Network Address and Port Translation (NAPT) devices; therefore, computer networks that include NAT/NAPT devices cannot be analysed with this solution.

Another limitation is that the solutions are not compatible with each other. A model of a security control used in one approach cannot be reused in another. This means that a lot of research time is lost on modelling various security controls for each approach. For example, the reachability analysis model in [4] can also handle NAT devices; but, since the analysis model of this solution is not compatible with the one in [3], the model of NAT/NAPT devices cannot be reused and a new model must be defined to support this type of security control.

This paper first gives an extended overview of research carried out in this field and highlights the advantages, disadvantages, and limitations. Based on this analysis, we show that future research in this area should be concentrated on a unified analysis model. We also discuss what features this model should include and why such a model is desirable.

The rest of the paper is organized as follows. Section II presents the theoretical background. Section III presents a typical workflow that the network administrators may use to configure firewalls. Section IV presents the research carried out on different types of policy analysis techniques. Section V presents the summary of what is missing in the literature and how to fill the gap. Section VI concludes and summarizes the paper.

II. TECHNICAL BACKGROUND

A. Network Security Policy

A *network security policy* is a special kind of policy that focuses on security aspects of a computer network. Network security policies can be written in different formats and at different levels of abstraction. On the one hand, very abstract high-level policies exist which are written in natural language, that express network-wide security goals. On the other hand, concrete configuration of single security controls are written in a device-specific configuration language. High-level policies are easy to write and understand by humans but difficult to elaborate on machines; concrete configurations which are difficult to read and write for humans are easily interpreted by machines.

B. Security controls

Security controls are appliances or software modules of appliances within a computer network. They implement the functionalities needed to enforce a network security policy. Security controls can control the network traffic by blocking certain packets or modifying it by changing header information of certain packets. As an example, packet filters, stateful firewalls, and application-level firewalls are used to control

 ISBN: 978-1-61208-378-0

the traffic, whereas IPsec gateways, Virtual Private Network (VPN) terminators, and NAT/NAPT devices are able to modify the traffic.

C. Policy Analysis

Each of the three main policy analysis types focuses on a part of the analysis process, but they have overlapping functions and common steps to reach their goal.

Conflict analysis searches for possible errors within a single or a set of security policies. It searches for potential semantic errors within correlated policy rules. Conflict analysis can also be used to identify possible policy optimizations. Conflict analysis can be applied to a single policy (Intra-Policy analysis) or to set of policies of interconnected security controls (Inter-Policy analysis).

Reachability analysis evaluates allowed communications within a computer network. Furthermore, it can determine if a certain host can reach a service or a set of services. In general, reachability analysis is performed online by using tools such as "ping" or "traceroute". By using an accurate representation of the network and its security policies, reachability analysis can also be performed offline, during the design phase.

Policy comparison compares two or more network security policies and represents the differences between them in an intuitive way. Network security policies involved may include single concrete security control configurations, sets of configurations, and high-level policies of an entire network. One of the best use-cases of policy comparison is to verify that a desired network security policy is implemented correctly by comparing the designed high-level policy with the concrete network configuration.

III. Network Administrator work-flows

Research efforts in this field can be divided into two main groups: policy generation that proposes a complete new approach to policy definition, and policy analysis that tries to give additional support to already deployed systems.

The policy generation approach forces network administrators to completely redefine their workflow and to use less expressive configuration interfaces. Policy generation has the disadvantage that administrators cannot rely anymore on their previous work experience and have to trust a black box policy generation tool. Probably the bigger disadvantage is that already deployed systems cannot be integrated seamlessly, instead they have to be reconfigured from scratch by mean of policy generation tools.

The policy analysis approaches, works on already deployed devices, thus it has the advantage that administrators can continue their usual work and use policy analysis support only during complex tasks. Deployed systems remain unchanged and under the complete control of network administrators.

A. Policy generation workflows

The policy generation approach consists of three main parts: a high-level security policy, a model of the network topology and a policy refinement tool. The network administrator specifies the desired network security policy using a high-level language and abstractly represents the target network topology. The high-level security policy and the network representation are the input for the policy refinement tool. The transformation process implemented by the tool produces the device-specific configuration files.

The advantages of such approaches are limited by the expressiveness of the high-level policy, the number of supported device types and device manufactures, and the optimization that the transformation process introduces to the final configuration.

B. Policy analysis workflows

The application of policy analysis solutions proposed in literature follow specific workflows. Conflict analysis searches for potential errors within a configuration. Reachability analysis allows the administrators to query if specific properties of the configuration are true. Last but not least, policy comparison helps network administrators to identify differences between policies.

For a complete workflow from the design phase to implementation, testing and maintaining a network policy, all three analysis approaches must be applied. First, during the design phase of a policy, network administrators express the desired network security policy in a high-level language. Since at the moment there are no enterprise grade transformation tools to transform high-level policies into device specific configurations, administrators have to create the configurations by hand. The next step is to use a conflict analysis tool to identify potential errors, performance issues and rules which are never applied. After having reduced potential errors and performance issues, administrators may use a reachability analysis tool to verify that the key aspects of the desired policy are applied correctly. The last step is to use a policy comparison tool to compare the desired network security policy with the newly created one.

Other activities can be performed after the configurations have been deployed. Administrators may want to troubleshoot a connection problem using the reachability analysis tool. Having pinned down the connection problem to a missing firewall rule, the administrator wants to verify that a modification of this firewall configuration does not introduce conflicts and that only the desired change is applied. First, he uses the conflict analysis tool and afterwards he uses the policy comparison tool to compare the original with the modified configuration.

IV. State of the art

A. Policy refinement

In literature, different approaches exist towards automatic policy generation. Even though they show a great potential, the research is still ongoing and has not been adopted widely. Only a few enterprise grade products exist which have implemented such features and the adoption rate is fairly low. Algosec, the leader in network security management, has only a few thousand costumers. According to one of their surveys [5], only 13.4% of network operators use a centralized policy management whereas 74.8% do not use any type of automated tools. The survey considers as centralized policy management any type of policy analysis and policy refinement.

 ISBN: 978-1-61208-378-0

Bartal et al. propose a solution named Firmato [6]. It was one of the first solution proposals in this area and supports only packet filter firewalls. It is based on an entity-relationship model of the security policy and of the network topology. The entity-relationship model is compiled and translated into firewall specific configuration files. The prototype was used to manage a real network containing a single firewall with 50 rules.

Verma et al. [7] used a similar approach; the authors present a firewall analysis and configuration engine named FACE. It takes as input the network topology and a global security policy written in a high level language. FACE has two advantages over Firmato: firstly it can also analyse the firewall configurations created and secondly it configures only one secure path between source and destination instead of inserting ACCEPT rules on every possible path.

Garcia-Alfaro et al. [8] proposed MIRAGE, a management tool for the analysis and deployment of configuration policies. It is based on the same principles as Firmato [6] and FACE [7], but it is also capable of configuring intrusion detection systems IDS and VPN routers. MIRAGE can also perform policy analysis on already deployed configurations.

Casado et al. [9] take a different approach; they proposed a solution named SANE. Instead of generating concrete configurations for already deployed firewalls, it proposed a new architecture where the network contains a central server which controls all decisions made by the network devices.

B. Conflict analysis

In literature, conflict analysis is mainly applied only to single types of security controls and there is no complete solution that incorporates all types of security controls. Research is mainly concentrated on Intra- and Inter-policy analysis of packet filter and IPsec configurations.

The conflict analysis of policy was first introduced by Al-Shaer and Hamed [3]. They presented a classification scheme for packet filter rule relations, based on which they defined the four types of intra-policy rule conflicts (shadowing, correlation, generalization and redundancy). Two rules are shadowed when they enforce different actions and both rules match the same packets. Two rules are correlated when they enforce different actions and both rules have some matching packets in common. A rule is a generalization of a second rule when they enforce different actions and the second rule matches the same packets as the first one but not vice versa. Two rules are redundant when they enforce the same action and match the same packets.

Al-Shaer et al. introduced an extension of the intra-policy classification analysis, called inter-policy rule conflicts, in the extension [10][11] of the first paper. Inter-policy analysis evaluates rule relations between serially-connected packet filters. Al-Shaer et al. define five new intra-policy conflicts (shadowing, spuriousness, redundancy, correlation and irrelevance). Two rules from two different firewalls are shadowed when they match the same packets and the rule from the first firewall blocks a packet that is permitted by the second rule. Two rules from two different firewalls are spurious when they match the same packets and the rule from the first firewall permits the packet which is blocked by the second rule. Two rules from two different firewalls are redundant when they match the same packets and both rules block the packet. Two rules from two different firewalls are correlated when they have some matching packets in common and enforce different actions. A rule is classified as irrelevant if there is no possible traffic which can be matched by the rule, for example the source and destination address belong to the same zone.

Based on the work of Al-Shaer et al., other researchers proposed alternative models and classification schemas. These works prove that Al-Shaers classification scheme is valid and can be applied to real world scenarios. The main limitation of all these approaches is that they cannot handle other security controls but packet filters. Notable examples are: Firecrocodile [12] and FIREMAN [13]. Firecrocodile [12], proposed by Lehmann et al., was the first approach to help network administrators to correctly configure PIX firewalls. The tool builds a model which represents the PIX configuration file and performs the analysis on it. In addition to conflict analysis they verify also the configuration file for policy violations. Its main limitation is that it can analyse only intra-policy packet filtering rules of Cisco PIX configurations. FIREMAN [13], proposed by Yuan et al., uses binary decision diagrams (BDDs) to represent packet filtering policies. In addition to a intra-policy analysis, it also verifies that an end-to-end policy is correctly implement by the filtering configurations. The model is designed for packet filters only and does not support any other type of security control.

Garcia-Alfaro et al. [14] propose the integration of network intrusion detection systems (NIDS). The model can detect both intra- and inter-policy packet filter rule conflicts. The main improvement over Al-Shaer's model is that it can also handle NIDS, and not only packet filters. The tool can also verify which security controls are on the path of a given packet based on its source and destination address. Another feature of this model is that it can rewrite a policy in its positive or negative form. The positive form of a policy contains only ALLOW rules whereas the negative form contains only DENY rules. This work has been later integrated into the MIRAGE tool [8].

Abbes et al. [15] suggest a different approach to this topic by using an inference system to detect intra-policy conflicts. They use the inference system to construct a tree representation of the policy. The construction process is efficient and optimized for memory consumption. The inference contains a condition which stops the construction of a specific branch when no conflict can be found. The resulting classification tree contains potential rule conflicts in its leaves. The disadvantage of this approach is that it is not able to check for inter-policy conflicts, furthermore it is not capable of handling security policies such as IPsec/VPN.

Only recently stateful firewalls have been integrated into analysis models. One of the few examples is presented in [16] and [17]. Cuppens and Garcia-Alfaro [16] propose a solution for intra-policy analysis of stateful firewalls. With the introduction of stateful firewalls they also present new types of conflicts classes (intra-state and inter-state rule conflicts). Intra-state rule conflicts occur only between stateful rules and beside the known conflicts from the stateless analysis, they include two new conflict types. The first new conflict arises when the firewall blocks packets during the three-way

 ISBN: 978-1-61208-378-0

handshake. The second new conflict arises when the firewall blocks packets during the connection termination. Inter-state rule conflicts occur between stateful and stateless rules when application layer protocols establish multiple connections and at least one of this connections is blocked, an example of such a protocol is FTP. The proposed algorithmic solution to handle and eliminate such types of conflicts is based on a general automata describing the stateful rules. This initial work has been completed and formalized in [17]. Although the introduction of stateful firewall into the analysis process was a very important step, both solutions are still missing the inter-policy analysis.

Basile et al. [18] present an new analysis model based on the work of Al-Shaer. The authors introduce a new formal model for policy specification, named Geometrical Model, it is based on a set of rules, a default action and an ad hoc resolution strategy. The presented model can identify all types of intra-policy conflicts defined by Al-Shear. Furthermore, the authors present two new conflict types: general redundancy anomaly and the general shadowing anomaly. The general redundancy anomaly occurs when a rule is redundant to the union of multiple rules. The general shadowing anomaly occurs when a rule is shadowed by the union of multiple rules.

Basile et al. [19] present, based on their Geometrical Model, a extension which can perform conflict analysis of application-level firewall configurations. The extended model can identify all policy anomalies introduced in their previous work. The main contribution of this work is the conflict analysis of firewall rules including regular expressions. The model transforms the regular expressions into deterministic automata and calculates rule intersection based on them.

Fu et al. [20] present a first approach for IPsec policy conflict detection. The analysis is performed on a set of policy implementations written in a high-level language and the policy conflicts are identified by verifying the implemented policies against a desired one. Fu et al. define a conflict when the policy implementations do not satisfy the requirements of the desired policy. A simple example of such a policy conflict is when the desired policy specifies that node A must have an encrypted channel with host B, but the policy implementations do not instantiate an encrypted channel from A to B. In addition to conflict detection, the proposed solutions includes also conflict resolution. The conflict resolution process tries to find alternative policy implementations in order to satisfy the desired policy.

Al-Shaer [21] formalizes the classification scheme of [20]. The proposed model not only incorporates the encryption capabilities of IPsec, but also its packet filter capabilities. The work can be seen as the extension of its packet filter classification proposed by Al-Shaer et al. [11]. In particular he identified two new IPsec conflicts (overlapping-session and multi-transform conflict), both types are valid for inter and intra-policy analysis. Nested session conflicts occur when multiple IPsec session are established from the same source to different remote hosts and the traffic is delivered to the farther host before the nearer one. Multi-transform conflicts occur when traffic protection is applied to already encapsulated IPsec traffic and the second protection is weaker than the first one. Al-Shaer presents in [22] a complete taxonomy of policy conflicts concerning packet-filter and IPsec configurations. This is the only approach who tries to perform conflict analysis of two different security controls.

Li et al. [23] present a similar detection classification model for IPsec security policy conflicts. The model takes in consideration intra- and inter-policy conflicts but is not compatible with the packet filter rule classification model presented by Al-Shaer. Instead of the conflicts defined by Al-Shaer they present a new alternative one. The new classification scheme is essentially the same but has the advantage that its definition is clearer and therefore easier to implement.

Niksefat and Sabaei [24] present a improved version of Al-Shaer's [21] solution. The new detection algorithm can identify all IPsec conflicts defined by Al-Shaer but does not support filtering conflicts. The solution uses a Binary Decision Diagram (BDD) to represent IPsec policies. The main improvement over Al-Shaer's solutions is the performance of the implementation. Beside the improved efficiency in the implementation this approach can also resolve the detected conflicts.

C. Reachability analysis

Reachability analysis can be performed both online and offline. Online reachability analysis is performed on a deployed system by injecting test packets and verifying on different points of the network that those packets are present. Offline reachability analysis is performed on a model of the system without direct interaction with a real network.

Online reachability analysis in general is performed by using tools such as *ping*, *traceroute*, and *tcpdump*. There are only a few publications regarding this topic. The general approach taken in literature is to insert a traffic generator and a traffic analyser into the network. The most promising work is presented by El-Atawy et al. [25] and Al-Shaer et al. [26], they propose a traffic generator which analyses first the security policy and based on this analysis, the most relevant packets are generated. The limitation of this two approaches is that they can be applied to single firewalls only.

Offline reachability analysis has the advantage that the system to be analysed does not need to be deployed. This means that it can be used during the design and maintenance tasks. Furthermore, it can also verify reachability on alternative paths, and therefore test fault-tolerance properties of the systems.

Mayer, Wool, and Ziskind [27] present a firewall analysis engine called Fang. It is the first approach towards offline reachability analysis of computer networks containing only packet filters. The proposed solution takes as input the network topology and the configuration files of the deployed packet filters. A user interface to perform reachability queries is provided and the queries are evaluated by the tool. In the extended versions of the paper [28] and [29] the query interface has been improved and the most relevant queries are generated automatically by the tool.

Xie et al. based there reachability analysis on graph theory and dynamic programming [30]. The solution is able to calculate the upper and lower bound of reachability. The upper bound defines that there is at least one possible path for reachability and the lower bound defines that all possible paths allow reachability. The model can be used to represent

 ISBN: 978-1-61208-378-0

static NAT, routing and filtering rules based on the destination addresses, but it does not take into account the existence of connectionless and connection-oriented protocols. Although the correctness of the model is given, it is purely theoretical and lacks experimental results. Bandhakavi et al. [31] present an extension to Xie's work to overcome limitations. They use a more general model to describe firewalls, packet filtering and transformation rules, thus adding the possibility to handle policies that depend on source addresses and filtering states.

Khakpour and Liu [4] present a reachability analysis tool called Quarnet. Quarnet supports connectionless (stateless router/firewall and static NAPT) and connection-oriented transport protocols (stateful router/firewall and dynamic NAPT). The paper presents a model for calculating network reachability metrics and also includes a performance analysis. The solution is based on an internal representation of the network on which reachability queries are executed. The authors first calculate a Firewall Decision Diagram (FDD) to represent the global policy and afterwards compute two matrices which contain the effective reachability information needed. Although the single reachability queries are very fast to compute, it takes quite a long time to compute the internal representation of the network.

Another theoretical approach used to compute the network-wide reachability, has been proposed by Sveda et al. [32]. This approach uses traditional graph-based algorithms, such as Floyd-Marshall, whereas [30] and [31] require ad-hoc techniques to mimic routing protocols. To calculate the reachability of the network the authors use the encoding problem into SAT instance solved by automatized solvers. They describe how to represent both routing and filtering devices, but do not mention how to express packet transformation rules.

Kazemian et al. [33] present a generalization of Xie's work [30] based on "Header space" information of packets. Their algorithm is compatible with filtering, routing, and transformation technologies. However, this approach is limited to packet filters and cannot be used for filtering and security devices which work at a higher level of the ISO/OSI stack.

D. Policy comparison

Fu et al. [20] present a solution proposal to verify the correct implementation of IPsec policies. The algorithm presented takes as input high-level security policies describing an implementation and compares it with a desired end-to-end policy. Even though the algorithm is able to compare a desired policy with its implementation, it cannot been used to compare a modified policy with its original version. Furthermore, it only supports IPsec policies and does not support routing or other transformation policies. This approach is more directed towards conflict analysis then policy comparison.

Liu et al. [34] and [35] propose to reduce configuration errors by forcing network administrators to write two separate concrete configurations and to compare them afterwards. The two configurations are converted into two FDDs and the comparison is performed onto the two FDDs. The comparison algorithm merges the two FDDs and verifies that the action, contained in the leaves of the tree, is the same at each point. Possible conflicts found in the two FDDs must be corrected manually by the administrators and without any correlation to the original configurations. This approach can be generalized and the two input policies may be seen as the original and the modified policy.

Yin and Bhuvaneswaran [36] represent correlations between rules as spatial relations and show how this special relations can be used to evaluate the impact of rule changes on the policy. Filtering policies are represented by the so-called SIERRA tree. A SIERRA tree is similar to a FDD, each level of the tree represents a dimension of the special division. The impact analysis can only be performed on single changes, such as adding one rule, removing or replacing it. The performance of the algorithm is very poor since to calculate the difference between two policies containing 30 rules takes already several seconds.

Liu et al. have published two papers on change-impact analysis of firewall policies [37][38], his algorithm is based on a FDD and supports the classic 5-tuple filtering rules. Overlapping rules are eliminated during the creation of the FDD and as a result the FDD represents a filtering policy without overlapping rules. The algorithm is designed to support four basic operations on firewall policies: rule deletion, rule insertion, rule modification, and rule swap. The output of the algorithm presents an accurate impact of a proposed change. Furthermore, the algorithm is also capable of correlating the impact of a policy change with a high-level security requirement. Although the authors claim that the algorithm is practical, neither of the two papers does a performance evaluation of the presented algorithms.

Liu et al. [39] present a firewall verification tool which takes as input a firewall policy and a given property. The tool verifies that the policy satisfies the given property. The tool is mainly useful for offline firewall debugging and troubleshooting. The algorithm first converts the firewall policy into a FDD and the verification process is performed on the FDD. The verification process checks that all leafs, which are correlated to the given property, enforce the desired action. A implementation of the tool has been tested for performance and shows excellent results. This solution is limited to one single firewall and cannot verify the correct implementation of a complete network.

Youssef et al. [40] propose a formal and automatic verification method based on a inference system. The solutions certify that a firewall configuration is sound and respect completely to a security policy. In case that the configuration is not sound and complete, the method provide the user with information tosolve the issues. This paper only supports packet filter firewalls; however in an extended version [41], Youssef et al. propose a formal and automatic method to check also statefull firewall configurations.

E. Summary

Table I summarizes the capabilities of the different approaches. Each row stands for one approach identified by its citation number. The three analysis categories (conflict analysis, reachability analysis and policy comparison) are separated by horizontal lines. Each column stands for a specific capability; the first four columns identifies the type of analysis (intra-policy conflict analysis, inter-Policy conflict analysis, reachability analysis, and policy comparison) and the last

seven columns identify the supported security control (packet filter firewall, stateful firewall, application-level firewall, NIDS, IPsec/VPN, NAT/NAPT, and routing).

TABLE I. SUMMARY

	Intra-Policy	Inter-Policy	Reachability	Comparison	Packet Filter	Stateful FW	Application	NIDS	IPsec/VPN	NAT/NAPT	Routing
[3]	⊗				⊗						
[10] [11]	⊗	⊗			⊗						
[12]	⊗				⊗						
[13]	⊗	⊗			⊗						
[14]	⊗	⊗			⊗			⊗			
[15]	⊗				⊗						
[16] [17]	⊗				⊗	⊗					
[18]	⊗				⊗						
[19]	⊗				⊗		⊗				
[20]	⊗			⊗					⊗		
[21] [22]	⊗	⊗			⊗				⊗		
[23]	⊗	⊗			⊗				⊗		
[24]	⊗	⊗							⊗		
[4]			⊗		⊗	⊗				⊗	
[27]			⊗		⊗						
[28] [29]			⊗		⊗						
[30]			⊗		⊗					⊗	⊗
[31]			⊗		⊗	⊗				⊗	
[32]			⊗		⊗					⊗	
[33]			⊗		⊗					⊗	⊗
[34] [35]				⊗	⊗						
[36]	⊗			⊗	⊗						
[37] [38]				⊗	⊗						
[39]				⊗	⊗						
[40]				⊗	⊗						
[41]				⊗	⊗	⊗					

By comparing the different approaches, the two major limitations are evident. Firstly, the majority of papers is concentrated only on packet filters and ignore other security controls. Secondly, the papers mainly focus only on one of the three analysis types (conflict analysis, reachability analysis and policy comparison), and only a few try to combine different approaches into one single model.

V. FUTURE RESEARCH

As it becomes clear from the analysis of research carried out so far, there is a lack of interoperability among the various models. This has three major disadvantages. Firstly, a security control modelled for one research approach cannot be reused in another one. Secondly, the execution time spent to instantiate a model is repeated for each and every analysis performed on network security policies. Thirdly, it is nearly impossible to make a performance comparison of the different approaches since they use different test scenarios or do not present a performance evaluation at all.

By combining all the proposed analysis techniques into one single extensible model, all of these disadvantages are eliminated and a proper analysis framework is created for future research. Firstly, after a security control has been modelled, evaluated and implemented it can be used by all types of analysis techniques. Secondly, when a network administrator wants to perform different types of analysis, he has to insert the required information and instantiate the model just ones. Thirdly, by having just one model, new algorithms can be evaluated by comparing them directly to each other.

To accomplish this goal, the new model should have some distinctive features, such as well-defined input formats, a flexible structure, and extendible bindings. Furthermore, the new model may include tests-scenarios to evaluate the performance of new algorithms.

The new model has to take as input the network topology, and the network security policies written in different formats and for different security controls. For example, it could take as input the global network security policy written in a technology-independent formal language and the complete network structure with all its concrete configurations. The model can then perform a policy comparison between the two input formats and verify that the implementation follows the desired network security policy. As a further step, network administrators can verify reachability of critical components or perform a conflict analysis for better understanding.

The new model has to be flexible to accommodate all types of security controls and network topologies. In order to support all types of computer networks, the model should be able to compose different security controls in different order. Security controls should be modelled so that they are completely independent form network topology.

The new model has to be extensible for new types of security controls. In order to be prepared for future security controls, the model has to be able to include new ones without significant changes to the model itself.

VI. CONCLUSION

Need for better tools to support network administrators is evident, from the number of publications regarding this topic. Although publications are very promising, they are only at the beginning. The analysis of articles has shown that the research is concentrated on quite small sub-problems and there exists no global solution to the problem.

By combining all research approaches into one single model, the impact grows in two dimensions: first, the number of possible analysis types, and second, the number of supported security controls. This leads to a model that can perform different policy analysis and, at the same time, covers a wider range of security controls. This approach leads to two improvements: firstly, reduced research effort and secondly, reduced execution time.

The research effort is mainly reduced because security controls have to be modelled only once and afterwards they can be used for different policy analysis. The execution time is reduced mainly because the model is shared by various policy analysis and its creation has to be performed only once.

ACKNOWLEDGMENT

The research described in this paper is part of the SECURED project, co-funded by the European Commission (FP7 grant agreement no. 611458).

REFERENCES

[1] A. Wool, “Firewall Configuration Errors Revisited,” CoRR, vol. abs/0911.1240, 2009, pp. 103–122.

[2] W. Avishai, “Trends in Firewall Configuration Errors: Measuring the Holes in Swiss Cheese,” IEEE Internet Computing, vol. 14, no. 4, July 2010, pp. 58–65.

[3] E. Al-Shaer and H. Hamed, “Firewall Policy Advisor for anomaly discovery and rule editing,” in IFIP/IEEE 8th Int. Symposium on Integrated Network Management, Colorado Springs, CO, March 24–28 2003, pp. 17–30.

[4] A. Khakpour and A. Liu, “Quarnet: A tool for quantifying static network reachability,” IEEE/ACM Trans. Netw., vol. 21, no. 2, February 2009, pp. 551 – 565.

[5] AlgoSec Inc., “Examining the dangers of complexity in network security environments: Algosec survey insights,” 2012.

[6] Y. Bartal, A. Mayer, K. Nissim, and A. Wool, “Firmato: A novel firewall management toolkit,” ACM Transactions on Computer Systems, vol. 22, no. 4, November 2004, pp. 381–420.

[7] P. Verma and A. Prakash, “FACE: A Firewall Analysis and Configuration Engine,” in 2005 Symposium on Applications and the Internet, Trento, Italy, January 31 – February 4 2005, pp. 74–81.

[8] J. Garcia-Alfaro, F. Cuppens, N. Cuppens-Boulahia, and P. Stere, “MIRAGE: a management tool for the analysis and deployment of network security policies,” in SETOP 2010: 3rd International Workshop, Athens, Greece, September 23 2011, pp. 203–215.

[9] M. Casado, T. Garfinkel, and A. Akella, “SANE: A protection architecture for enterprise networks,” in USENIX-SS06: USENIX Security Symposium, Vancouver, Canada, July 31 – August 4 2006, pp. 137–151.

[10] E. Al-Shaer and H. Hamed, “Discovery of policy anomalies in distributed firewalls,” in INFOCOM 2004, Hong Kong, Cina, March 7–11 2004, pp. 2605–2616.

[11] E. Al-Shaer, H. Hamed, R. Boutaba, and M. Hasan, “Conflict classification and analysis of distributed firewall policies,” IEEE Journal on Selected Areas in Communications, vol. 23, no. 10, October 2005, pp. 2069–2084.

[12] N. Lehmann, R. Schwarz, and J. Keller, “FIRECROCODILE: A Checker for Static Firewall Configurations,” in SAM06: International Conference on Security & Management, Las Vegas, NV, June 26–29 2006, pp. 193–199.

[13] L. Yuan, H. Chen, J. Mai, and C.-n. Chuah, “FIREMAN: A Toolkit for FIREwall Modeling and ANalysis,” in IEEE Symposium on Security and Privacy, Berkeley/Oakland, CA, May 21–24 2006, pp. 199–213.

[14] J. Garcia-Alfaro, N. Boulahia-Cuppens, and F. Cuppens, “Complete analysis of configuration rules to guarantee reliable network security policies,” International Journal of Information Security, vol. 7, no. 2, April 2007, pp. 103–122.

[15] T. Abbes, A. Bouhoula, and M. Rusinowitch, “An inference system for detecting firewall filtering rules anomalies,” in SAC08: ACM symposium on Applied computing, Fortaleza, Brazil, March 16–20 2008, pp. 2122–2128.

[16] F. Cuppens, “Handling Stateful Firewall Anomalies,” in SEC2012: Information Security and Privacy Conference, Heraklion, Greece, June 4–6 2012, pp. 174–186.

[17] J. Garcia-Alfaro, F. Cuppens, N. Cuppens-Boulahia, S. Martinez, and J. Cabot, “Management of stateful firewall misconfiguration,” Computers & Security, vol. 39, no. A, November 2013, pp. 64–85.

[18] C. Basile, A. Cappadonia, and A. Lioy, “Network-Level Access Control Policy Analysis and Transformation,” IEEE/ACM Transactions on Networking, vol. 20, no. 4, August 2012, pp. 985–998.

[19] C. Basile and A. Lioy, “Analysis of Application-Layer Filtering Policies With Application to HTTP,” IEEE/ACM Transactions on Networking, vol. PP, no. 99, December 2013, pp. 1–1.

[20] Z. Fu, S. F. Wu, H. Huang, K. Loh, F. Gong, I. Baldine, and C. Xu, “IPSec/VPN Security Policy: Correctness, Conflict Detection, and Resolution,” in International Workshop, POLICY 2001, Bristol, UK, January 29–31 2001, pp. 39–56.

[21] E. Al-Shaer, H. Hamed, and W. Marrero, “Modeling and Verification of IPSec and VPN Security Policies,” in 13th IEEE Int. Conference on Network Protocols, Boston, MA, November 6–9 2005, pp. 259–278.

[22] E. Al-Shaer and H. Hamed, “Taxonomy of conflicts in network security policies,” IEEE Communications Magazine, vol. 44, no. 3, March 2006, pp. 134–141.

[23] Z. Li, X. Cui, and L. Chen, “Analysis And Classification of IPSec Security Policy Conflicts,” in FCST06: Japan-China Joint Workshop on Frontier of Computer Science and Technology, Fukushimna, Japan, November 17–18 2006, pp. 83–88.

[24] S. Niksefat and M. Sabaei, “Efficient Algorithms for Dynamic Detection and Resolution of IPSec/VPN Security Policy Conflicts,” in 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, WA, April 20–23 2010, pp. 737–744.

[25] A. El-Atawy, T. Samak, Z. Wali, E. Al-Shaer, F. Lin, C. Pham, and S. Li, “An Automated Framework for Validating Firewall Policy Enforcement,” in POLICY07 : 8th IEEE International Workshop on Policies for Distributed Systems and Networks, Bologna, Italy, June 13–15 2007, pp. 151–160.

[26] E. Al-Shaer, A. El-Atawy, and T. Samak, “Automated pseudo-live testing of firewall configuration enforcement,” IEEE Journal on Selected Areas in Communications, vol. 27, no. 3, April 2009, pp. 302–314.

[27] a. Mayer, a. Wool, and E. Ziskind, “Fang: a firewall analysis engine,” in 2000 IEEE Symposium on Security and Privacy, Berkeley, CA, May 14–17 2000, pp. 177–187.

[28] A. Wool, “Architecting the Lumeta Firewall Analyzer,” in 10th USENIX Security Symposium, Washington, DC, August 13–17 2001, pp. 85–97.

[29] A. Mayer, A. Wool, and E. Ziskind, “Offline firewall analysis,” International Journal of Information Security, vol. 5, no. 3, July 2006, pp. 125–144.

[30] G. Xie, D. Maltz, A. Greenberg, G. Hjalmtysson, and J. Rexford, “On static reachability analysis of IP networks,” in INFOCOM2005: 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, March 13–17 2005, pp. 2170–2183.

[31] S. Bandhakavi, S. Bhatt, C. Okita, and P. Rao, “Analyzing end-to-end network reachability,” in IM09: IFIP/IEEE Int. Symposium on Integrated Network Management, Long Island, NY, June 1–5 2009, pp. 585–590.

[32] M. Sveda, O. Rysavy, and G. D. Silva, “Static Analysis of Routing and Firewall Policy Configurations,” in ICETE10: 7th International Joint Conference, Athens, Greece, 2012, pp. 39–53.

[33] P. Kazemian, G. Varghese, and N. McKeown, “Header space analysis: Static checking for networks,” in NSDI12: 9th USENIX conference on Networked Systems Design and Implementation, San Jose, CA, April 25–27 2012, pp. 9–9.

[34] A. Liu and M. Gouda, “Diverse Firewall Design,” in Int. Conference on Dependable Systems and Networks, Florence, Italy, June 28 – July 1 2004, pp. 595–604.

[35] A. Liu and M. Gouda, “Diverse Firewall Design,” IEEE Transactions on Parallel and Distributed Systems, vol. 19, no. 9, September 2008, pp. 1237–1251.

[36] Y. Yin and R. Bhuvaneswaran, “Inferring the Impact of Firewall Policy Changes by Analyzing Spatial Relations between Packet Filters,” in ICCT06: Int. Conference on Communication Technology, Guilin, China, November 27–30 2006, pp. 1–6.

[37] A. Liu, “Change-impact analysis of firewall policies,” in 12th European Symposium On Research In Computer Security, Dresden, Germany, September 24–26 2007, pp. 155–170.

[38] A. Liu, “Firewall policy change-impact analysis,” ACM Transactions on Internet Technology, vol. 11, no. 4, March 2012, pp. 1–24.

[39] A. Liu, “Formal Verification of Firewall Policies,” in 2008 IEEE International Conference on Communications, Beijing, China, May 19–23 2008, pp. 1494–1498.

[40] N. Ben Youssef, A. Bouhoula, and F. Jacquemard, “Automatic verification of conformance of firewall configurations to security policies,” in 2009 IEEE Symposium on Computers and Communications, Sousse, Tunisia, July 5–8 2009, pp. 526–531.

[41] N. B. Youssef and A. Bouhoula, “Dealing with Stateful Firewall Checking,” in DICTAP2011, Dijon, France, June 21–23 2011, pp. 493–507.

 ISBN: 978-1-61208-378-0