Teaching-learning firewall configuration using a visual modeling web based tool: the SP2Model and its application to Computer Science course

Helton Molina Sapia, Rogério Eduardo Garcia,
Celso Olivete Júnior, Danillo Roberto Pereira
Departamento de Matemática e Computação
Faculdade de Ciências e Tecnologia
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Presidente Prudente - SP - Brazil
Email: helton.sapia@gmail.com; {rogerio, olivete}@fct.unesp.br;
dpereira@ic.unicamp.br

Kleber Manrique Trevisani
Instituto Federal de Educação
Ciência e Tecnologia de São Paulo
Presidente Epitácio - SP - Brazil
Email: kleber@ifsp.edu.br

***Abstract*—The traffic between computer networks must be controlled to prevent unauthorized access. Firewall is responsible for filtering data packets between networks, applying rules to select data packets that can get in/out to access the network. The firewall must be configured accordingly network access policy. Therefore, the students have to be trained to acquire skills not only to understand network access policy, but also to translate it into firewall native language. The use of a graphical representation (high level) to model the network access policy consist in a resource to facilitate understanding and to minimize defects on firewall native language. For that, we have proposed an extension to Security Policy Modeling Language (SPML), the SPML2, which aims to create a visual representation of the network access policy using graphical notation. Also, in this paper we present SP2Model, a web based tool to support SPML2 network accesses policy modeling and its translation into firewall native language. Using the SP2Model, the student can model the network access policy in a graphical notation and, then, can generate the set of rules in firewall native language. The use of SP2Model has facilitated the teaching-learning process, compared to traditional approaches. We evaluated SP2Model (and consequently the SPML extension) through an experiment following the GQM paradigm (Goal / Question / Metric) comparing the traditional approach and the use of SPML2 in operational security education. We performed two experimental sessions and we present the results, as well as a discussion about the tool, its use and the trade-off on using it.**

***Index Terms*—Learning Firewall Configuration; Practical Activities; Teaching Methodology; Visual Modeling.**

I. Introduction

Firewalls are used to protect internal networks. A crucial step to accomplish a trustworthy protection is the configuration of the firewall by network administrator according to the security policy. Generally, the security policy is expressed in a document through natural language, and determines which internal network objects can be accessed from the external network and which external network objects can be accessed from the internal network [1].

Security policy translation into native firewall rules is a complex task and liable to mistake. The network administrator must comprehend correctly the security policy and translate it into a set of rules using the firewall native language. The emergence of new network services and the modifications of in security policy cause changes in the firewall rules. The modification of firewall rules must be error free (for example, do not release packet that should be blocked or block packet that should be released) [2].

Wool [3], [4] examined firewalls organizations with different technologies, from organizations with own department to implement and maintain the security policy that used free solutions. He observed that all firewalls were misconfigured and, therefore, all networks were vulnerable.

This scenario has evidenced the importance of teaching-learning the course of computer networks. Emphasizing firewalls configuration that perform the controls according to the security policy.

One of the authors of this study has worked as professor of computer networks and teaches the perimeter protection course over 13 semesters. He has observed that students have difficulties to produce the configuration in firewall native language. The main difficulties are (i) understand the security policy relation with the network objects (internals and externals) and (ii) set the firewall, in according with the security policy, producing the configuration in the firewall native language.

The use of an abstraction capable of representing the internal knowledge of experts in a graphic visual form – as concept maps – facilitates the sharing of information and hence the process of teaching and learning [5].

The use of visual presentation to model the security policy and its translation for firewall native rules might be useful to student of perimeter protection supporting his/her studies and, consequently, facilitate the understanding the configuration defined in the security policy.

Trevisani and Garcia [6] proposed a visual model approach

for security policy modeling, Security Policy Modeling Language – SPML. The SPML uses diagrams showing interconnected components to represent graphically the functionality of a firewall. The SPML was originally represented by two diagrams, one for the filtering rules and another for translation rules. However, the symbol that represents the firewall is the same that represents the network address translation, causing an ambiguity. Also, the SPML does not allow the blocking of packets.

In this paper we present two rounds of experiment using SPML2, an evolution of SPML, as well as its results and a discussion about it. The purpose of the experiment is to have an evidence on using a visual approach to support teaching-learning of firewall configuration, complementing the teacher evaluation.

The remainder of this is organized as follows. In Section II presents the context required and the firewall configuration processes. Related works are briefly presented in Section VI. The description of the SPML2 approach is performed in Section III. Section V shows the experimental setup, the results and a discussion. Finally, Section VII presents conclusions and further work.

II. Firewall configuration

In order to protect the communication between computer networks, the firewall needs to address goals [7], such as:

- All traffic addressed to the protected network or originated in the protected network must be delivered to the firewall.
- Only authorized traffic previously defined by the local network access policy, can be delivered to your destination.
- The firewall itself must be immune to unauthorized access.

The forward or drop action of a packet is defined by rules that use the information contained in the packet header of the network layer (protocol field, source IP address and destination IP address) and information contained in the packet header of transport layer (source address and destination respectively inform the socket port number associated with the source process and the socket port number associated with the target process) [8]. Each rule can be associated with a network interface. A firewall positioned between two networks will have two network interfaces, each interface should connect to a different network. When the packet is delivered to firewall interface, the firewall checks if data headers matches with any rule. Once matched, the action defined by the rule is executed. If the packet does not match then a default action is performed. There are two possible policies for the default action:

- Discard: the package that is not expressly permitted is discarded and;
- Forward: the package that is not expressly prohibited is forwarded.

To implement the control of communications between network computers, the computer science student needs to understand the network access policy to then translate that policy in rules of firewall language. The student must make sure there are no errors and to validate the rules in native language firewall. As noted in the Figure 1, the procedure has three major phases: understand, translate and validate.

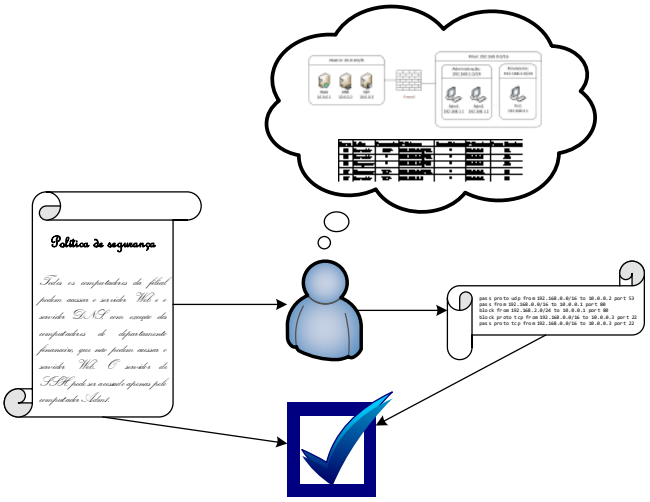


Fig. 1. Implementation of communication controls.

We have observed that Computer Science students have difficulty to associate network equipment with rules that define actions expressed in network access policy. Also, they have difficulty to understand the traffic direction on interfaces through which packet must pass or be blocked.

For example, lets consider that only the computer used as administrator located in a subsidiary can access the SSH server located in DMZ network. In this scenario, it is usual the difficulty to understand that the policy require two rules: (i) a rule to release incoming packets on interface that connects the administrator computer with the firewall; and (ii) another rule to release outgoing packets on interface that connects the firewall with the SSH server. Those rules need specify that the origin packet is the administrator computer and the destination is the SSH server.

During the process of teaching and learning we observed that when the network access policy was also presented graphically, before being written in firewall rules, students had fewer questions about the rules. The students create rules with fewer errors in the direction of traffic, the associated interface and the header information.

III. SPML2 approach

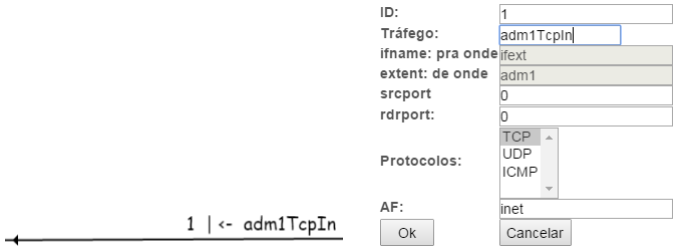
The SPML2 is an evolution of Security Policy Modeling Language (SPML) with ambiguity free avoiding the lack of packet traffic blocking. It is composed by components able to represent the security policy for the network communications. Such components are used to represent the firewall, the external entities and the packets traffic authorized by the security policy.

Each component is a firewall feature depicted in a graphical representation along with its attributes. There are attributes

not visible in graphical notation. The SPML2 components are organized in four folders, they are: (i) firewall components, (ii) translation components, (iii) filtering, and (iv) external components. The translation components and filtering determine the action should be applied to packets traffic. The external components to the firewall (external entities) might may be the source or the destination of traffic. External entities can not connect with each other, they necessarily need to connect to firewall. To visually express the security policy, the SPML2 components are described as follows.

(a) (b) (c)

Fig. 2. Firewall components of the SPML2, (a) Firewall, (b) Form Firewall and (c) Form Interface.



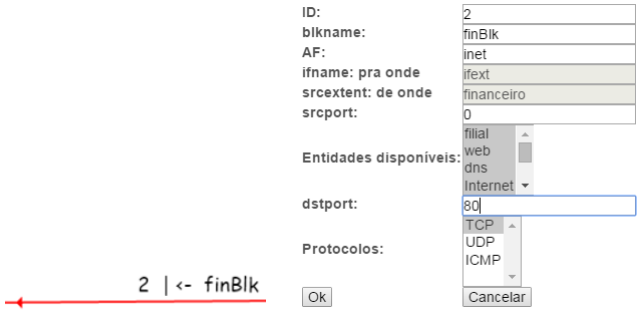
(a) (b)

Fig. 3. SPML2 filtering components: (a) Incoming traffic release and (b) Form incoming traffic release.

(a) (b)

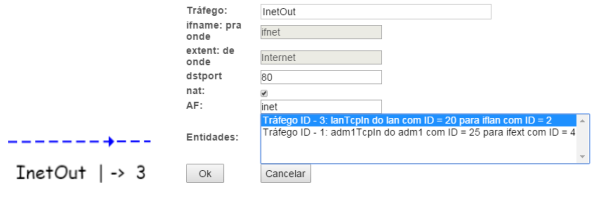
Fig. 4. SPML2 firewall filtering components: (a) Outgoing traffic release and (b) Form outgoing traffic release.

1) The circle is used to represent the couple hardware/-software that performs firewall functions, as shown in Figure 2.(a). The *firewall* component has the following attributes: *name* and *default policy*, see Figure 2.(b). Default policy indicates the action (*pass* or *block*) to be adopted for packets that were not explicitly defined in the security policy.



(a) (b)

Fig. 5. Filtering components of firewall of the SPML2, (a) Block traffic and (b) Form block traffic.



(a) (b)

Fig. 6. SPML2 translation components: (a) Traffic translation and (b) Form traffic translation.

2) A firewall can have one or more network interfaces, allowing external components connection. An external component can not connect directly to the firewall, requiring the interface component. The *interface* is graphically represented by a rectangle, as depicted in Figure 2.(a), and has the attributes: *interface name*, *device name*, *IP address*, *mask* and *firewall name*, as shown in Figure 2.(c).
3) The traffic filtering components are represented by an arrowed line (Figure 3.(a), Figure 4.(a) and Figure 5.(a)). The color indicates if the packet can pass through or be blocked in the interface: black indicates *incoming traffic*, blue indicates *outgoing traffic* and red indicates *blocked traffic*. In addition, the arrow connected to interface indicates an *incoming traffic* and connected to external entity indicates an *outgoing traffic*. The *incoming traffic* attributes are: *traffic id*, *traffic name*, *associated interface*, *associated external entity*, *source port*, *port redirect*[1], *protocol* and *version of the IP address* as shown in Figure 3.(b). The *traffic id* must be unique for each incoming traffic. It is used to bind an *incoming traffic* to one or more *outgoing traffic*. The *outgoing traffic* attributes are: *traffic name*, *associated interface*, *associated external entity*, *destination port*, *network address translation*[2] and one or more *traffic id* as shown in Figure 4.(b). The *traffic id* must have been defined by an *incoming traffic*. The *blocked packets* attributes are: *traffic id*, *block name*, *version of the*

[1]Applicable only to traffic redirection.
[2]Applicable only to network traffic translation.

(a) (b)

Fig. 7. SPML2 redirect traffic components: (a) Redirect traffic and (b) Form redirect traffic.

IP address, *associated interface*, *external entity source associated*, *source port*, *external entity destination associated*, *destination port* and *protocol*[3] as shown in Figure 5.(b).

4) *Translation* component is used to represent the traffic of packets that pass through the interface and require translation (port or address). It is represented by a dashed arrowed line (Figure 6.(a) and Figure 7.(a)). Black indicates a *port redirect* and blue indicates a *network address translation*. A packet of *outgoing traffic* will be translated when *network address translation* attribute is marked, as depicted in Figure 6.(b). A packet of *incoming traffic* will be redirected when *port redirect* is presented as depicted in Figure 7.(b).

(a) (b)

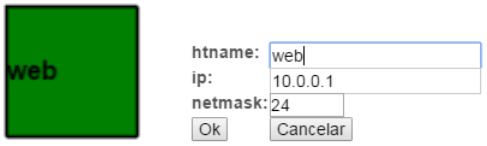
Fig. 8. External entities of the SPML2, (a) Unknown network and (b) Form unknown network.

(a) (b)

Fig. 9. External entities of the SPML2, (a) Network and (b) Form network.

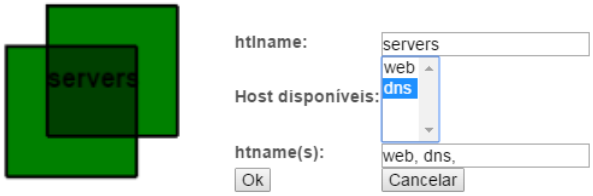
5) The *unknown network* component (Figure 8.(a)) is used to set the Internet or any network that does not have IP prefix defined, its unique attribute is *name* as can be seen in Figure 8.(b).
6) The *network* component is a TCP/IP network, graphically represented by a square with rounded corners (Figure 9.(a)). The network attributes are *name*, *network prefix* and *subnet mask* as can be seen in Figure 9.(b).

[3]The components *block* and *pass* are explained on item 3.



(a) (b)

Fig. 10. External entities of the SPML2, (a) Host and (b) Form host.



(a) (b)

Fig. 11. External entities of the SPML2, (a) Host list and (b) Form host list.

7) *Host* is the component used for any device capable of receiving an IP address. It is represented by a square, see Figure 10.(a), and its attributes are *name*, *IP address* and *subnet mask* (Figure 10.(b)).
8) *Host list* is a group of hosts on the same security policy traffic. It is represented graphically by two overlapped squares (Figure 11.(a)). Its attributes are *name* and two or more *hosts* previously defined on the host component, as depicted in Figure 11.(b).

A. Scenario example SPML2

To illustrate how security policy is mapped using the SPML2, we used a scenario using a fictitious organization, as illustrated in Figure 12. In that scenario, computers from

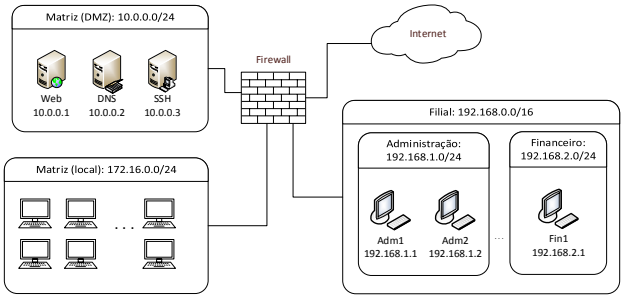


Fig. 12. Fictitious network used as scenario.

different departments can access services on servers in the company headquarters. The headquarters local network computers require address translation to access the Internet. The organization security policy is set to allow all affiliate computers access the *Web* server and the *DNS* server, with exception of the finance department computer, which can not access the *Web* server. The *SSH* server can be accessed only by *Adm1*

computer. The corresponding SPML2 visual presentation is shown in Figure 13.

B. The SP2Model – a web based tool

The rationale to use SPML2 is: first, a graphical modeling of security policy is drawn; then, the set of rules is automatically in firewall native language. To support the SPML2, we developed a tool named SP2Model (basically, we used HTML5 and JavaScript). SP2Model is a web-based tool that allows creating visual models of security policy using SPML2 (all figures depicting SPML2 components and their attributes are instantaneous from SP2Model). To illustrate its operation, the security policy using the scenario previously described (Figure 12) was modeled as depicted in Figure 13.

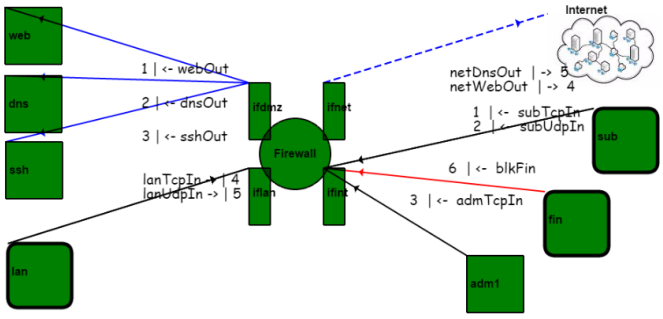


Fig. 13. Security Policy modeled in SPML2.

Using the graphical representation elaborated, it is possible to generate the set of rules into firewall native language. The modeled security policy expressed in firewall native language generated by SM2Model is shown in Listing 1.

Listing 1. Filtering rules in firewall native language – Packet Filter syntax

```
#DefaultPolicy
block in all
#Traffic ID 0
pass in on em1 inet proto tcp \
      from 192.168.0.0/16 to 10.0.0.1 port 80
pass out on em0 inet proto tcp \
      from 192.168.0.0/16 to 10.0.0.1 port 80
#Traffic ID 1
pass in on em1 inet proto udp \
      from 192.168.0.0/16 to  10.0.0.2 port 53
pass out on em0 inet proto udp \
      from 192.168.0.0/16 to 10.0.0.2 port 53
#Traffic ID 2
pass in on em1 inet proto tcp \
      from 192.168.1.1 to 10.0.0.3 port 22
pass out on em0 inet proto tcp \
      from 192.168.1.1 to 10.0.0.3 port 22
#Traffic ID 3
pass in on em2 inet proto tcp \
      from 172.16.0.0/24 to any port 80
pass out on em3 inet proto tcp \
      from 172.16.0.0/24 to any port 80 nat-to (em3)
#Traffic ID 4
pass in on em2 inet proto udp \
      from 172.16.0.0/24 to any port 53
pass out on em3 inet proto udp \
      from 172.16.0.0/24 to any port 53 nat-to (em3)
#Traffic ID 5
block in on em1 proto tcp \
      from 192.168.2.0/24 to 10.0.0.1 port 80
```

IV. Teaching Firewall Configuration: The Evaluation

In order to evaluate the use of SPML2 and its implementation in SP2Model by students, we conducted an experiment comparing firewall native language and SPML2 usage using a group of 15 computer science students. The experiment aims to compare security policy mapping (textual) to rules in native language firewall using visual approach (SPML2) model and traditional approach (Ad Hoc). For that, we consider the efficiency and the effectiveness by the group of undergraduate students.

The experiment was defined using the Goal/Question/Metric paradigm [9] and the template of Wohlin et al. [10]. For comparison purpose, we considered the following activities:

- Traditional configuration: using a network access policy, the students needed to write the set of firewall rules directly in firewall native language.
- SPML2-based Configuration: using a network access policy and SP2Model tool, the students needed to create the corresponding SPML2 diagram and, then, generated the set of rules in firewall native language.

The experiment was conducted focusing on the quality of produced rules, considering its correctness and time spent in preparation. The experiment was conducted in the network lab of São Paulo State University (UNESP), Campus at Presidente Prudente,

Two different access network policies (named Policy A and Policy B) were used during the experiment. Policies were prepared considering the inconsistencies between release/block packages implemented at the firewall for the authorized/unauthorized traffic provided by the network access policy [11]. The number of rules in the native firewall language necessary to implement each policy is shown in Table I.

TABLE I
Number of rules in native firewall language.

Policy	Number of rules
Policy A	31
Policy B	47

The experiment included 15 undergraduate Computer Science students, at ninth semester. All students have concluded the courses Computer Networks I and Computer Networks II. It is interesting to note that their participated as voluntary. The profile that characterizes most of the participants is:

- Never set up a firewall (11 participants).
- Never performed the firewall rules mapping from a network access policy (13 participants).
- Never studied modeling techniques for mapping the network access policy (12 participants).
- All students have theoretical background.

The students were divided into two groups (Group 1 and Group 2) according to arrival order. This criteria was used to randomization and, consequently, to avoid biased groups. The number of students per group is shown in Table II.

TABLE II
NUMBER OF STUDENTS PER GROUP.

Group	Number of students
Group 1	8
Group 2	7

The main goal is to analyze whether the visual approach (SPML2) is more effective than the traditional approach (Ad Hoc) during the teaching-learning process of firewalls configuration. Therefore, the null hypothesis was formulated to determine the significance of the adopted approach (Ad Hoc or SPML2) to the effectiveness Efe (the ability of students to translate the network access policy to set of rules in native firewall language) and to the efficiency Efi (the ability of students convert network access policy to set of rules in native firewall language in less time). Table III presents the formalization of assumptions considered.

TABLE III
HYPOTHESES INVESTIGATED IN THE EXPERIMENT.

Effect	Hypothesis type	Hypothesis formulated
Efe	Null hypothesis	$H_0^1 : Efe_{SPML2} \leq Efe_{AdHoc}$
	Alternative hypothesis	$H_a^1 : Efe_{SPML2} > Efe_{AdHoc}$
Efi	Null hypothesis	$H_0^2 : Efi_{SPML2} \leq Efi_{AdHoc}$
	Alternative hypothesis	$H_a^2 : Efi_{SPML2} > Efi_{AdHoc}$

The independent variable is the traffic controlled by the network access policy. The network access policy defines the authorized traffic set t_AP and unauthorized traffic set $t_{NA}P$. The union of authorized traffic set and unauthorized traffic set defined by the security policy provides the traffic of the network access policy $tP = t_AP \cup t_{NA}P$.

The dependent variables are traffic controlled correctly by the firewall and time to map into firewall rules. Settings of a firewall define which traffic set must be released t_LF and which traffic set must be blocked t_BF – the union of traffic set released and blocked by the firewall results in the traffic set controlled by the firewall $tF = t_LF \cup t_BF$.

The time spent is measured in minutes and denotes the time necessary to map the network access policy rules to native firewall language through the traditional approach (Ad Hoc) and visual approach (SPML2). Each participant must produce the firewall native language rules that implement the network access policy, and shorter time denotes that students were more efficient (an evidence that they have correctly comprehended the translation process).

The effectiveness score is measured using the traffic set provided by network access policy correctly defined in traffic set controlled by the firewall $Efe = tP \cap tF$, and the efficiency is measured by time in minutes spent TS for the preparation of firewall rules divided by the number of correct rules produced $Efi = TS/Efe$. Correct rules produced is not zero for all participants.

We conducted two experimentation sessions. So that, during the first round the Group 1 mapped the policy A using the traditional approach (Ad Hoc) and Group 2 mapped policy A using SPML2. In the second round, we used the policy B: Group 2 mapped out the policy B using the traditional approach (Ad Hoc) while Group 1 mapped the policy B using SPML2. The crossing groups allow to avoid the influence of policies on performing the required tasks. Table IV shows the organization of the experiment sessions.

TABLE IV
ORGANIZATION OF THE SESSIONS OF THE EXPERIMENT

Session	Policy	Traditional approach	SPML2 approach
Session 1	Policy A	Group 1	Group 2
Session 2	Policy B	Group 2	Group 1

TABLE V
ACTIVITIES FOR IMPLEMENTATION OF THE EXPERIMENT.

Day	Time	Task		
1	15 min	Presentation of the experiment		
	5 min	Signing the consent form		
	10 min	Fill the participant's profile questionnaire		
	90 min	Training: Firewalls configuration, Packet Filter and SPML2		
2	10 min	Organization of participants		
			Ad Hoc	SPML2
	120 min	Policy A implementation	Group 1	Group 2
3	120 min	Policy B implementation	Group 2	Group 1

The activities are presented in Table V, indicating the time spent for each activity. As depicted, the experiment tasks carried out over three days. On the first day we prepare the experiment: it was explained about the tasks to be performed, and then each participant signed a consent form and completed a questionnaire aimed to characterize their profile; also, the two training sessions were held in configuring firewalls, one using the traditional approach to write rules in native language, another using SPML2 to create the SPML model using SP2Model. The training material was designed to familiarize participants with the tasks they should perform. On the second day, it was held the first section of the experiment: the firewall rules relating to policy A were produced, group 1 used the traditional approach and group 2 used the SPML2. On the third day, it was held the second section of the experiment: the firewall rules relating to policy B were produced, group 1 used the SPML2 approach and group 2 used the traditional approach. The sections of the experiment occurred in network laboratory. The computers were configured with the necessary tools and files necessary to the study. As participants arrived, they were assigned to groups 1 and 2 (first to group 1, second to group 2, third group 1, etc), so as to keep them in balance.

One should take into consideration that the SPML2 approach requires all firewall components and all external entities to be configured prior to the drafting of traffic and locks.

V. RESULTS AND DISCUSSION

In this section we discuss the results obtained using the experimental setup described in the Section IV. In order to

allow a statistical evaluation, we performed the computation of the Wilcoxon test [12] over the average accuracy and time.

The average accuracy obtained by Ad Hoc and SPML2 methodology are presented in Table VI. The Figures 14 and 15 present the accuracy scatter and the accuracy box plots, respectively. The SPML2 obtained the best values for the two groups for the configuration firewall task considering Wilcoxon test. For both groups the average accuracy of the SPML2 were at least two times greater than the traditional Ad Hoc. In the Figures 16 and 17 we can observe that broadly the accuracy and the time spent of the SPML2 is significantly better than Ad Hoc.

TABLE VI
THE AVERAGE ACCURACY OF THE AD HOC AND SPML2 CONSIDERING THE EXPERIMENTAL SETUP. THE BEST VALUES ARE IN BOLD.

	Accuracy	
	Ad Hoc	**SPML2**
Group 1	17.34%	**50.27%**
Group 2	7.60%	**31.80%**

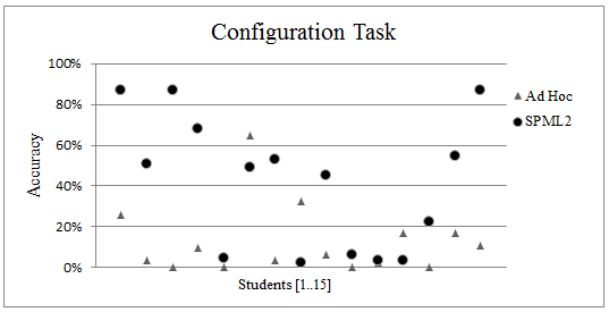


Fig. 14. Accuracy scatter plot.

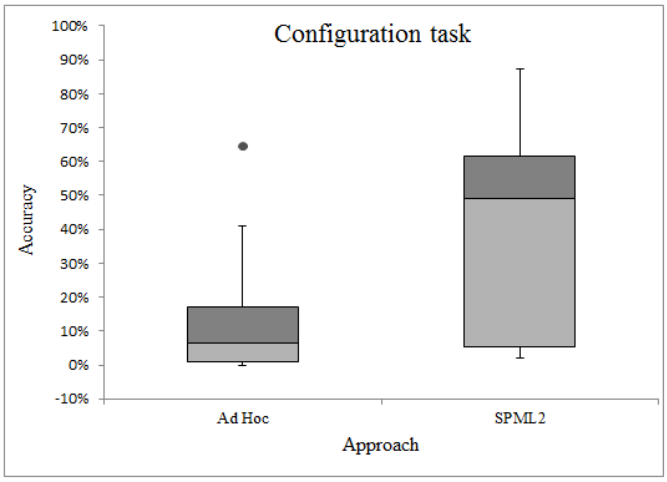


Fig. 15. Accuracy box plot.

The average time spent for the configuration task considering Ad Hoc and SPML2 methodology are presented in Table VII. The Figures 16 and 17 present the time scatter and the time box plots, respectively. The SPML2 obtained the best value in one out two groups for the configuration firewall task. One may observe on Figure 16 that just two students spent more time in the SPML2 than Ad Hoc approach; however the time difference is very larger, what influences significantly the average.

For Figures 16 and 17 the missing values denote the students that not drafted none firewall rule correctly.

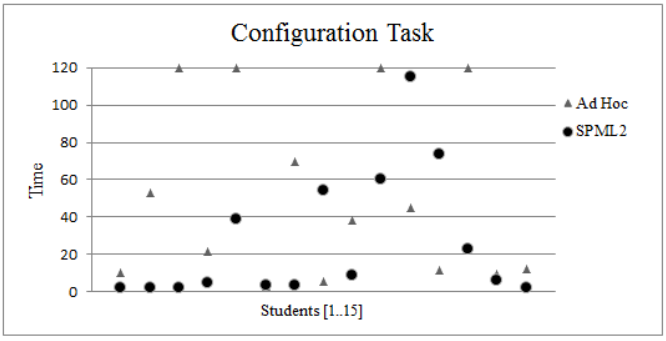


Fig. 16. Time scatter plot.

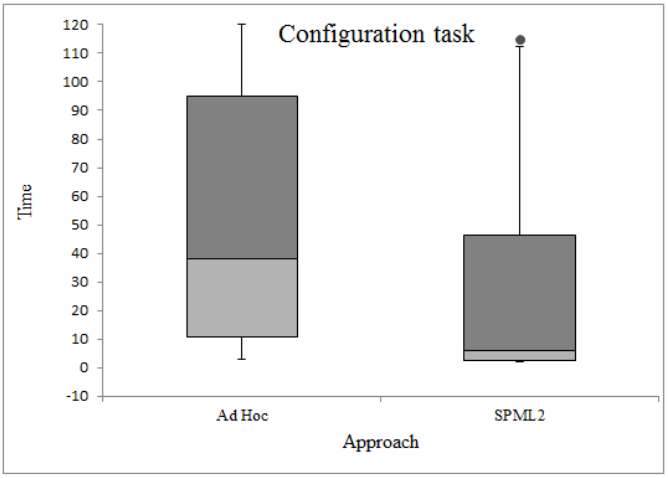


Fig. 17. Time box plot.

TABLE VII
THE AVERAGE TIME SPENT BY THE AD HOC AND SPML2 CONSIDERING THE EXPERIMENTAL SETUP. THE LOWEST TIME VALUES ARE IN BOLD.

	Time[minutes]	
	Ad Hoc	**SPML2**
Group 1	27.13	**11.51**
Group 2	**17.69**	41.8

VI. RELATED WORK

There are different techniques to support firewall configuration process that can be used in teach-learning process. Most provides assistance in validation of firewall configuration [13], [14]. Also, there are research works focusing on creating automated solutions to detect configuration inconsistencies firewalls through formal verification [15], [16].

The use of visual representations have facilitated the process of teaching and learning for students from various fields of knowledge [5].

Studies have concluded that "multisensory" learning approach provides improvement in learning process, facilitating the flow of information to the learners' brains in whole and simultaneous [17].

Mind Mapping is a technique that facilitates the students to learn the necessary information about determined context. It increase an individual's capacity to learn [18].

Masmann et al. [19] pointed out that firewalls configuration is a complex task. They argue that to express the firewall rules using visualization techniques makes the comprehension of easy rules. In their research they present a visual representation of rules using Sunburst visualization. That visual representation do not show the direction of packets and do not present a full picture of the firewall.

The SPML [6] used the visual approach to modeling the firewall configuration. The SPML2 is an evolution of the SPML: problems of ambiguity and visualization were resolved; SPML2 provides validation of configuration and delivery rules in native language firewall.

The SPML2 uses an abstraction to provide to Computer Science student a holistic vision of firewall behavior, exploring the "Multi-Sensory Learning" using the visual approach to facilitate the Teaching-learning Process.

VII. Conclusion and further works

In this paper we present the use of SPML2, an extension of Security Policy Modeling Language [6], as a tool to support the teaching-learning process of firewall configurations. We present an evaluation using controlled experiment, its organization and results.

We point out two important advantages. The first, we proposed a new approach for teaching-learning process for firewall configuration using a visual representation. In general, the related works focused on validating rules of firewall. This paper present a holistic vision of firewall configuration, what increase the visibility of entire process to student. Second, the results demonstrate that SPML2 facilitates the comprehension process and translation of firewall rules. The accuracy obtained by SPML2 was significantly better than Ad Hoc. As presented in Section V, despite the average time of the SPML2 be bigger than Ad Hoc for Group 2, just two out 15 students spent more time using SPML2.

As mentioned, one of the authors has taught perimeter protection course over 13 semesters. According his evaluation, the teaching-learning process was facilitated by using SPML2 (and SP2Model). Also, SPML2 has simplified to express the security policy. The experiment was conducted to corroborate such empirical evaluation.

Regarding to further works, we aim at comparing the SPML2 and Ad Hoc methodology applied to maintenance firewall task, considering that the firewall rules maintenance is more difficult than an initial configuration.

Acknowledgment

We would like to thank all students who participated in the activities, for their dedication, interest and given replies to survey.

References

[1] B. Forouzan and F. Mosharraf, *Computer Networks: A Top-down Approach*, ser. Connect, Learn, Succeed. McGraw-Hill, 2012. [Online]. Available: https://books.google.com.br/books?id=ieKVkgAACAAJ

[2] A. Bandara, A. Kakas, E. Lupu, and A. Russo, "Using argumentation logic for firewall configuration management," in *Integrated Network Management, 2009. IM '09. IFIP/IEEE International Symposium on*, 2009, pp. 180–187.

[3] A. Wool, "A quantitative study of firewall configuration errors," *Computer*, vol. 37, no. 6, pp. 62–67, 2004.

[4] ——, "Trends in firewall configuration errors: Measuring the holes in swiss cheese," *Internet Computing, IEEE*, vol. 14, no. 4, pp. 58–65, 2010.

[5] M. L. Espinosa, N. M. Sanchez, Z. G. Valdivia, and R. B. Perez, "Concept maps combined with case-based reasoning in order to elaborate intelligent teaching/learning systems," in *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*, Oct 2007, pp. 205–210.

[6] K. M. Trevisani and R. E. Garcia, "Spml: A visual approach for modeling firewall configurations," *MODSEC08: International Conference on Model Driven Engineering Languages and Systems*, 2008.

[7] W. Stallings, *Cryptography and Network Security: Principles and Practice*. Prentice Hall, 2011. [Online]. Available: https://books.google.com.br/books?id=3alWQwAACAAJ

[8] B. Forouzan and S. Fegan, *Data Communications and Networking*, ser. McGraw-Hill Forouzan networking series. McGraw-Hill Higher Education, 2007. [Online]. Available: https://books.google.com.br/books?id=bwUNZvJbEeQC

[9] V. R. Basili, G. Caldiera, and H. D. Rombach, "Goal question metric paradigm," University of Maryland, College Park, MD, USA, Tech. Rep., 1994. [Online]. Available: https://cgis.cs.umd.edu/~basili/publications/technical/T89.pdf

[10] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*, ser. Computer Science. Springer, 2012. [Online]. Available: https://books.google.com.br/books?id=QPVsM1_U8nkC

[11] L. Yuan, H. Chen, J. Mai, C.-N. Chuah, Z. Su, and P. Mohapatra, "Fireman: a toolkit for firewall modeling and analysis," in *Security and Privacy, 2006 IEEE Symposium on*, 2006, pp. 15 pp.–213.

[12] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[13] A. El-Atawy, T. Samak, Z. Wali, and E. Al-Shaer, "An automated framework for validating firewall policy enforcement," in *Policies for Distributed Systems and Networks, 2007. POLICY '07. Eighth IEEE International Workshop on*, June 2007, pp. 151–160.

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

[15] A. Gawanmeh, "Automatic verification of security policies in firewalls with dynamic rule sequence," in *Information Technology: New Generations (ITNG), 11th Int. Conference on*, April 2014, pp. 279–284.

[16] M. Moussa, H. Ould-Slimane, H. Boucheneb, and S. Chamberland, "A formal framework for verifying inter-firewalls consistency," in *Computers and Communication (ISCC), IEEE Symp. on*, June 2014, pp. 1–7.

[17] P. Chan and G. Krishnaswamy, "Do educational software systems provide satisfactory learning opportunities for 'multi-sensory learning' methodology?" in *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '11. New York, NY, USA: ACM, 2011, pp. 358–358. [Online]. Available: http://doi.acm.org/10.1145/1999747.1999872

[18] L. Davis, "Mindmapx," *SIGCSE Bull.*, vol. 37, no. 3, pp. 405–405, Jun. 2005. [Online]. Available: http://doi.acm.org/10.1145/1151954.1067609

[19] F. Mansmann, T. Göbel, and W. Cheswick, "Visual analysis of complex firewall configurations," in *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, ser. VizSec '12. New York, NY, USA: ACM, 2012, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/2379690.2379691