MLSvisual: A Visualization Tool for Teaching Access Control Using Multi-Level Security

Man Wang
Department of Computer Science
Michigan Technological University
Houghton, MI
manw@mtu.edu

Steve Carr
Department of Computer Science
Western Michigan University
Kalamazoo, MI
steve.carr@wmich.edu

Jean Mayo,
Ching-Kuang Shene,
Chaoli Wang
Department of Computer Science
Michigan Technological University
Houghton, MI
{jmayo,shene,chaoliw}@mtu.edu

ABSTRACT

Information security continues to be a pressing issue for industry and government. Perhaps the two most fundamental mechanisms for controlling access to information are cryptography and access control systems. This paper presents MLSvisual, a tool that helps students learn the multi-level (Bell-LaPadula) access control model. MLSvisual allows students to create, explore, and modify an MLS policy through a graphical visualization system. A query system can be used by students to test their understanding of a given policy. Instructors can utilize a test function in the tool to assign an exercise or quiz, with answers sent to them via email. We also present the results of an evaluation of MLSvisual within a senior-level course on information security. This evaluation received positive feedback and showed that MLSviusal helped the understanding of the Bell-LaPadula model and enhanced the course. We believe that this user-level tool will help instructors to teach this material more effectively, and make teaching this material more practical in resource-constrained environments.

Categories and Subject Descriptors

k.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer science education, information systems education*

General Terms

Security, Access control model

Keywords

Security, visualization

ITiCSE'14, June 21–25, 2014, Uppsala, Sweden.

http://dx.doi.org/10.1145/2591708.2591730.

1. INTRODUCTION

Application of the principle of least privilege requires that a process be given access to only those resources necessary for it to complete its task. On modern systems, a very tight application of this principle can lead to a large (tens of thousands of rules) and complex access control policy that is challenging both to create and maintain.

This problem has been partially addressed through improved access control technology. Access control systems have evolved significantly over the last decade. A large part of the effort has been implementation of sophisticated security models, such as Multi-Level Security (MLS) [1, 2], Role-Based Access Control (RBAC) [7] and Type Enforcement (TE) [3]. These models abstract modern, common patterns of information access, and hence simplify policy development and administration.

Visualization has been applied to some access control models. Schweitzer, Collins, and Baird developed a visualization system to enable active learning about the Harrison, Ruzzo, Ullman and Take-Grant models of access control [11]. Hallyn and Kearns developed DTEEdit and DTEView for graphical analysis of DTE specifications [6]. DTEEdit and DTEView do not have pedagogical goals. Visualization and animation have also been applied in many areas of security education [4, 5, 8, 9, 10, 11, 12, 13, 14]. MLS is a fundamental access control model. To our knowledge, no visualization tool has been developed to help the teaching and learning of the model. This paper describes MLSvisual which aims to enhance the pedagogy of the MLS model. It allows students to create, modify, and analyze policies graphically. It also allows import and export of a human-readable text-based policy. To present and help explore the details, three graphical representations are used to illustrate a policy and an additional query subsystem is provided to answer some fundamental questions. Instructors may use a test module that requires students to answer questions about policies and then sends the answers via email. The system runs at the user-level and is not tied to the underlying file system. It currently supports Linux and MacOS. MLSvisual was tested in a senior-level course on computer security. The evaluation indicated that MLSviusal helped the understanding of the Bell-LaPadula model and enhanced the course.

The remainder of this paper is organized as follows: Section 2 provides the background of the computer security

course where MLSviusal was evaluated, Section 3 presents our tool, Section 4 has a detailed study of our findings from student evaluation, and Section 5 has our conclusions.

2. COURSE INFORMATION

MLSvisual was used in a computer security course, CS4471 Computer Security, that was offered in the Department of Computer Science at Michigan Technological University. It is a senior level course that gives a basic introduction to topics in computer security. The access control component covers the Bell-LaPadula (BLP), Domain Type Enforcement (DTE), and Role-Based Access Control (RBAC) models. The course also covers secure coding in C, cryptography, key management, authentication, malicious logic, and intrusion detection.

Most students are computer science majors who take the course as an elective. The class in which the evaluation was conducted included seventeen CS majors, three Computer Systems Science majors (who are required to take the course), three Software Engineering majors, seven Computer Engineering majors, and three students from other majors including Electrical Engineering, Math and Communications, Cultures and Media.

Students were given paper and pencil exercises on the BLP model as part of the regular course homework. For this first use of MLSvisual, students were additionally given an extra credit assignment that required use of the tool. The problem was to evaluate a simple policy via a series of questions and then complete a test using the `Test` module. After the students had submitted their solutions to the extra credit assignment problem, the instructor distributed a survey to the class. Completion of the survey was voluntary.

3. SOFTWARE DESCRIPTION

MLSvisual is designed to facilitate the teaching as well as self-learning of Multi-Level Security using visualization. It implements the Bell-LaPadula model, where security levels are assigned to subjects (users and processes) and objects (files and directories). A security level (L, C) consists of a clearance L and a subset C of a comprehensive set of categories. A clearance such as "Secret" or "Top secret" represents the sensitivity of a subject or an object, and a set of categories indicates descriptive attributes such as "Documentary" or "States". Security levels are compared using the dominates relation $\geq$, where $(L_1, C_1) \geq (L_2, C_2)$ if and only if $L_1 \geq L_2$ (L_1 has a sensitivity higher than or equivalent to L_2, e.g., "Top Secret" $\geq$ "Secret") and $C_2 \subseteq C_1$. A subject can only read objects it dominates, and can only write to objects that dominate the subject. The dominates relation on the security levels forms a directed graph $G(V, E)$ where V contains a node for each security level (L, C) and there is an edge from the node for (L_1, C_1) to the node for (L_2, C_2) if and only if (L_2, C_2) dominates (L_1, C_1). Self-loops and edges inferred by transitivity that appear in a full digraph of the partial order are omitted to reduce clutter. Since two security levels can not dominate each other, $G(V, E)$ is a directed acyclic graph with the starting node dominating all the other nodes in V.

Figure 1 shows the user interface and visualization illustrating the relationship among security levels. The part with a white background is the drawing canvas. The red frame has all color coded clearances, and the blue frame has all categories indexed by numbers. Users may specify the clearance and categories of a security level by checking the buttons in these frames, and use the `Add one node` operation to draw the node for the security level on the canvas. The node will have the clearance color and the indices of its categories as its label.

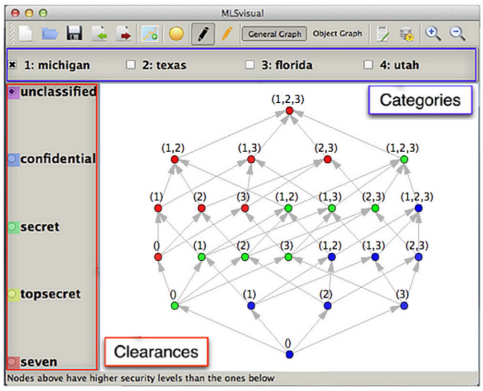


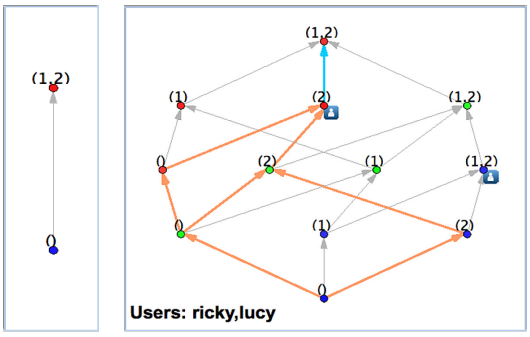
Figure 1: Main Window

MLSvisual supports the import and export of specification files (.mls) and visualization files (.mlsvis). A specification file contains a human-readable, text-based specification of components in an MLS policy, where clearances, categories, and the security levels of users and objects are designated. A visualization file stores the graph on the drawing canvas. A session may begin with loading a file of either type or building clearances and categories from scratch. The visualization focuses on the interpretation of the relationship among security levels and read and write permissions that subjects have towards objects. It includes three graphs: the `Whole graph`, the `General graph` and the `Object graph`. The `Whole graph` provides an automatically generated overview of the dominates relation among all security levels. The `General graph` allows users to gradually reveal the relations among security levels of interest. They may add nodes one by one or choose two existing nodes as end nodes and reveal the dominates relation between them using the `Generate graph` operation. The `Object graph` shows the security level assignment to objects. MLSvisual also has two modes: the `Analysis mode` and `Edit mode`. The `Analysis mode` allows the users to analyze properties of the policy they imported and helps the users better understand the BLP model. The `Edit mode` supports editing clearances and categories as well as the security level assignments to subjects and objects. This mode allows users to modify a policy and aims to help them design policies that fulfill specific security requirements. In addition, `Specification` and `Exercise` modules are provided for further exploration and design of MLS policies.

3.1 Visualization

The `Whole graph` $G_w(V_w, E_w)$ shows the directed graph described above of all security levels (Figure 4 (a)), where V_w contains all security levels and E_w contains the directed edges that represent the dominates relation. It starts with the node that dominates all the other nodes. The `General graph` $G_g(V_g, E_g)$ helps the users focus on the security levels and subjects of interest. (Figure 2). V_g is a subset of V_w, and E_g has the edges for the dominates relation among elements

in V_g. The `Whole graph` and the `General graph` together provide both overall and partial views of the relationship among security levels so that a full understanding of a policy becomes easier.



(a) Before (b) After

Figure 2: General Graph Before And After Generate Graph Operation

The `Object graph` shows the security level assignment of objects. This graph has a number of concentric circles with the center being the root directory. The circles with increasing radii represent directories of increasing directory depth. The nodes in the graph are objects and the edges represent the membership of the directory (Figure 3). Each node is a rectangle with two colors. The left color indicates its clearance and the right one shows the category based on the color-category correspondence in the legend.

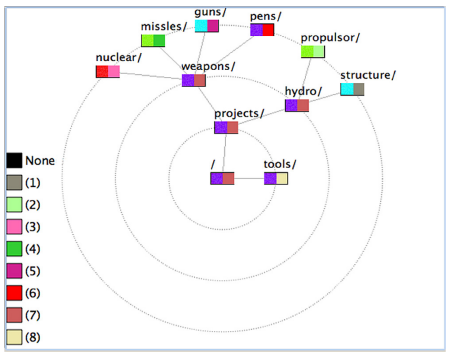


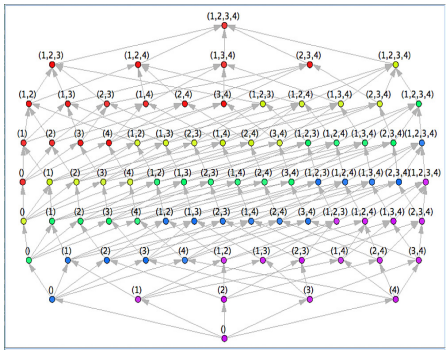
Figure 3: Object Graph

3.2 Analysis Mode

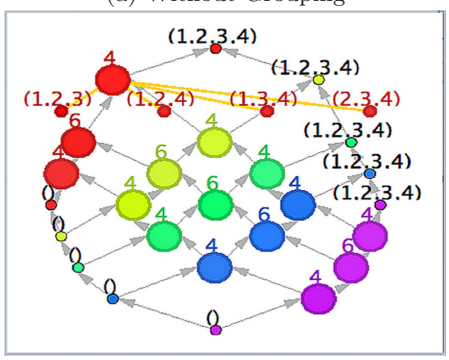
The `Analysis mode` is to facilitate the understanding of the relationship among security levels and permissions of subjects in the imported policy using the three graphs.

The `Whole graph` shows the relationship for all security levels. While a fine-grained policy may contain complicated and meaningful relationships, it also generates a cluttered graph. Hence, we use node grouping to reduce the clutter. If there are more than 15 nodes at one depth level in the graph, an expandable group node will replace the nodes with the same clearance at the same depth level and be labeled as the number of contained nodes. This group node can be replaced by the contained nodes when needed. Figure 4 (a) shows a `Whole graph` without grouping, while Figure 4 (b) has the same graph with grouping.

The `General graph` shows the relationship among some nodes in which users are interested. There are two methods to draw the relationship: adding security level nodes one at a time and using the `Generate graph` operation. The first method draws an edge directly between nodes when they are related under the dominates relation. When a new node is added, the graph is updated. This helps students investigate when one node is reachable from another. Using the second method, a user designates two nodes and the tool generates the full directed graph between them. Nodes along all paths from the lower node to the upper node, as well as the edges between them, are generated. This is useful when investigating the reachability of the two given nodes, the possible paths and the involved security levels. It also avoids the overwhelming and repetitive operations of adding nodes. However, the first method still has its value when users are not interested in the detailed paths between nodes and prefer just knowing the reachability. Figure 2 (a) shows two nodes and their relationship. Applying the `Generate graph` operation to the same pair of nodes, Figure 2 (b) shows all the security levels in between. An icon by a node indicates there is a subject assigned to this security level. When the node (e.g., the red node with label (2) in Figure 2 (b)) is moused over, the subject's name appears at the lower left corner of the canvas, and its permissions to other security levels are shown in highlighted paths. These subjects can write to the nodes along the blue paths and can read the nodes along orange ones. By referring to the security levels of specific objects in the `Object graph`, it is easy to tell the permissions that subjects have towards objects.



(a) Without Grouping



(b) With Grouping

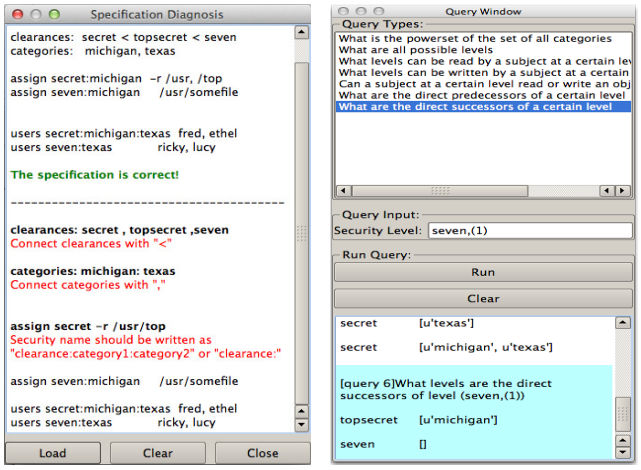
Figure 4: Whole Graph

3.3 Edit Mode

MLSvisual starts with the `Edit mode` to create a policy. It can also be used to modify an existing policy. A policy contains four components: clearances, categories, security levels of users and security levels of objects. This mode

provides four editing operations: `add/delete clearance`, `add/delete category`, `assign directory` (assigning security levels to objects) and `assign users` (assigning security levels to users in the operating system). One can move from the `Edit mode` to `Analysis mode` in the same session in order to evaluate policy changes.

3.4 Specification and Exercise Modules



(a) Specification Diagnosis (b) Query

Figure 5: Specification and Exercise Modules

Two more modules, `Specification` and `Exercise`, are provided to help the users understand the specification of an MLS policy and the BLP model. The `Specification` module has `Specification window` and `Specification diagnosis` components. The `Specification window` component generates a specification of the policy under consideration and is useful when a policy is being created graphically or the imported one is modified. The specification can be used as a guidance for writing correct specification files. The `Specification diagnosis` component is used to check the syntax of a `specification` file loaded in this module. If it is correct, confirmation of correctness will show up as the last line in green along with the original specification content in a pop-up window. Otherwise, information on how to correct the errors will be given under each problematic line (Figure 5 (a)).

The `Exercise` module consists of two components for self-evaluation: `Query` and `Test`. The `Query` component has seven questions (Figure 5(b)) to help the exploration of MLS policies. It provides answers to some frequently asked questions such as what are all sets of categories, what are the possible security levels and whether a specific subject has read or write permission to an object. The `Test` component provides a way to evaluate the understanding of clearance, category, relationships and permissions through 13 questions on policies in various scales. Users have to choose an answer to proceed to the next question. This can be used for in-class exercises or quizzes. Instructors will receive a student's answer, a grade on each question and overall grade via email. This component currently has an example set of questions covering the core aspects of the BLP model. Instructors may populate the test with their own questions by modifying an input text file.

4. EVALUATION

The MLSvisual evaluation consists of two components, 17 rating questions (Table 1) and 9 write-in comments. The first 14 questions (Q1-Q14) study the effects of MLSvisual. The choices are: 1:strongly disagree, 2:disagree, 3:neutral, 4:agree, and 5:strongly agree. Questions Q15, Q16 and Q17 study the use of MLSvisual. The choices for Q15 are 1:less than 5 mins, 2:5-10 mins, 3:10-15 mins, 4:15-30 mins and 5:more than 30 mins. The choices for Q16 are 1:once, 2:1-3 times, 3:3-5 times, 4:5-10 times and 5:more than 10 times. The choices for Q17 are 1:less than 5 mins, 2:5-15 mins, 3:15-30 mins, 4:30-60 mins and 5:more than 1 hour. We collected 22 valid forms. The distribution of majors is as follows: 10 in Computer Science, 6 in Computer Engineering, 3 in Computer Systems Science, 1 in Software Engineering, and 2 undeclared.

Table 1: Survey Questions

Q1	MLSvisual helped better understand BLP model
Q2	MLSvisual was helpful for my self-study
Q3	General graph's analysis mode showed the relationship between different security levels clearly
Q4	General graph's edit mode allowed easy creation and modification to policies
Q5	Object graph depicted files' security levels in a straightforward way
Q6	Whole graph helped better understand of policies
Q7	Representation and layout eased use of the tool
Q8	Colors helped understand BLP's information flow
Q9	Permissions of security levels are clearly depicted
Q10	The tool helped realize BLP's limitations
Q11	The tool helped learn Principle of Tranquility
Q12	Feel prepared to design policy after using the tool
Q13	The tool helped understand what wasn't understood
Q14	MLSvisual enhanced the course
Q15	How long did it take you to understand the BLP model by using the tool
Q16	How many times did you use the tool
Q17	How long did you use the tool in total

4.1 General Discussion

Table 2 shows the mean and standard deviation of each question. Feedback from participants was positive with an overall mean of 3.77 and standard deviation of 0.73. Q3 and Q8 received the highest scores of 4.2 and 4.3 with standard deviation 0.8 and 0.6, respectively. This indicates that the `General graph` showed the relationship among security levels clearly and that the use of colors helped students understand the BLP model. Q5 and Q11 received the lowest score 3.0. Q5 investigates whether the security levels of objects are straightforward in the `Object graph`. The low score may be because the `Object graph` and `General graph` were supposed to be used together. However, even if the security level assignment to the objects is visually presented, students probably treated them as separate and independent components, and hence Q5 received a neutral rating. Q11 received 3.0 because there is no direct visual presentation of this principle. Students have to edit a policy in several iterations to get hands-on experience of whether the strong or weak tranquility principle should be preserved. The `Edit mode` is designed for this purpose. The other questions re-

ceived scores around 4.0. Hence, the general response to the tool was positive and participants considered that the tool helped them understand the concepts and enhanced the course. Of the three usage questions (Q15-Q17), Q17 had an average of 3.6, which indicated that students used the tool for 15 to 30 minutes. The average of Q15 was 2.9 which means that it took around 10-15 minutes for students to understand the BLP model using MLSvisual. The average of Q16 was 1.5 showing that students used the tool once or twice. Table 3 has the distribution of answers to these three questions. Q15 had 9%, 23% and 41% of students select Choice 1, Choice 2 and Choice 3, respectively. Thus, 73% of all students required less than 15 minutes to understand the BLP model. Since no student selected Choice 5, all of them understood the BLP model within 30 minutes. The answer distribution of Q16 indicated that 50% of all students used it only once while the rest used MLSvisual twice. For Q17, 87% of all students selected among Choice 1 to Choice 4, which means that 87% of all students spent less than one hour using the tool.

Table 2: Mean (μ) and Standard Deviation (σ)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
μ	4.0	3.8	4.2	4.1	3.0	3.7	3.7	4.3	3.8
σ	0.6	0.7	0.8	0.7	0.9	0.6	0.7	0.6	0.9

	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17
μ	4.0	3.0	3.6	3.6	3.9	2.9	1.5	3.6
σ	0.7	0.9	0.6	0.8	0.5	0.9	0.5	1.0

Table 3: Usage Distribution

	Choice1	Choice2	Choice3	Choice4	Choice5
Q15	9%	23%	41%	27%	0
Q16	50%	50%	0	0	0
Q17	5%	9%	23%	50%	13%

We also looked at the correlations between each pair of questions from Q1 to Q14. The ratings of each question are loosely positively related with the highest correlations 0.65 for (Q3, Q10) and 0.64 for (Q7, Q8). The correlation between Q3 and Q10 suggested that those who considered the `Analysis mode` showed the relationship among security levels clearly (Q3) also tended to believe that MLSvisual helped them realize the BLP's limitations (Q10). For Q7 and Q8, those who considered the representation and layout made the use of MLSvisual easy (Q7) also might consider the color scheme helped them understand the information flow of the BLP model (Q8). There are some other pairs having correlations around 0.55. The correlations between (Q3, Q4) was 0.56, indicating that students who liked the `Analysis mode` of the `General graph` (Q3) also rated the `Edit mode` of the `General graph` (Q4) higher. The correlation 0.55 of (Q6, Q10) suggested that students who rated the `Whole graph` (Q6) higher might find it easier to realize the limitations of BLP model (Q10). The correlations between (Q1, Q13) and (Q2, Q13) were 0.52 and 0.55, respectively. This suggested that many students who felt that MLSvisual helped them understand what was not understood also tended to consider the tool helped self-study and a better understanding of the BLP model.

4.2 Statistical Analysis

We used MANOVA and ANOVA to investigate if the use of the tool may affect student ratings. The level of significance is $\alpha = 0.05$. The null hypothesis for this study is: the time spent on understanding the BLP model (Q15), the number of times using this tool (Q16), and the total time spent on this tool (Q17) did not affect the answers to the 14 questions (Q1-Q14). Based on the answers to Q15, we divided students into 3 groups. Group 1 had students who spent less than 10 minutes to understand the model. Group 2 spent 10 to 15 minutes, and group 3 spent more than 15 minutes. The p-value of a MANOVA Wilk's lambda test was 0.525, suggesting that there was no significant difference among these groups. To verify the result, we also used ANOVA to perform individual test against Q15, and found that Q5 vs. Q13 had the smallest p-values 0.051. Since it is still larger than the level of significance, we can not reject the null hypothesis.

Students were divided into two groups according to their responses to Q16. The first group had 11 students who used the tool only once. The second group had the other 11 students who used the tool twice. The p-value of a MANOVA Wilk's lambda test was 0.677, which indicated that the null hypothesis can not be rejected. ANOVA tests against Q16 showed that Q1 and Q12 had the two smallest p-values 0.062 and 0.070, respectively. Since they are still greater than the significance level, the null hypothesis can not be rejected.

For Q17, we divided students into 2 groups. The first group included 8 students who used the tool for less than 30 minutes while the other group of 14 students spent more than 30 minutes. The MANOVA Wilk's lambda test had a p-value of 0.332, and the null hypothesis can not be rejected. ANOVA tests against Q17 showed that the p-value for Q13 (0.0046) was the only one less than the significance level. The null hypothesis was rejected. Therefore, students who spent less than 30 minutes and the students who spent more than 30 minutes responded to Q13 differently. This happened because students used the tool after learning the BLP model in class. The parts they did not understand before were some challenging aspects. The different responses showed that many students were able to understand the challenging parts after spending enough time on the tool. Based on the findings, we have sufficient evidence to claim that the time students spent on the tool affects whether they could understand the parts that they did not understand before. But, in general, the use of the tool does not affect student rating when all questions are considered at the same time.

4.3 Student Comments

The set of 9 write-in questions was designed to gather suggestions from students for future improvement. The aspects we investigated are: whether the graph presentation is helpful, the `Specification diagnosis` module, the `Test module`, the use of colors and user interface, features to add and the software installation issues.

Student feedback was quite positive to the graph presentation. Some students said "*It clearly illustrated the lattice formed by the policy, and helped me see the relationship between levels*", "*The graph was very nice and definitely helped me understand the BLP model better*", "*The graph showed useful information with button to auto-generate*", and "*It worked perfectly as I imagined*". Therefore, we believe that the graph presentation did help students understand the BLP model better.

The comments on the `Specification diagnosis` model

were generally positive. Students mentioned that "*It was definitely useful*" and "*It was a nice addition to the visual*". However, some students mentioned that they were not sure whether they had used the module. This is understandable since the extra credit assignment did not include the use of this module.

The `Test` module received positive feedback. Students mentioned that "*I was impressed by how well the software handled examples*" and "*The most populated object graph was nice*". A suggestion "*It would be better if there were answers to the questions at the end of the test*" was also mentioned. Since instructors usually use the module as a quiz, the questions can be answered on demand in class.

All students were satisfied with the use of colors and the user interface. A student suggested that "*Queries should default to a pop-out window*". Most of them did not think about additional features; however, one student indicated that "*Maybe a quick run down on the model and particular specification*". No software installation problem was reported.

Students also provided some general comments for further improvement. They suggested adding tooltip to all buttons, having the larger default window size, and providing a version for 64-bit Linux since some of their systems were not 32-bit compatible and needed some packages installed before use.

In summary, we believe that MLSvisual effectively helped self-learning and in-class teaching of the MLS policies and BLP model. With the suggestions from the students, we will improve MLSvisual in the near future.

5. CONCLUSIONS

This paper discusses a visualization tool MLSvisual to facilitate the teaching and self-learning of the MLS access control model. Instructors may use the tool in class and read in policies while explaining the concepts and properties. Students who are interested in learning the model on their own or exploring the model further after class may use the tool to understand the model better. Students may also learn the design of an MLS policy and perform self-evaluations.

The evaluation showed that MLSvisual was helpful. In the grouping analysis, MANOVA tests found no difference in rating against student's use of the tool considering all questions at the same time while ANOVA tests showed that the time students spent on the tool affected whether they were able to understand the parts that challenged them before. As suggested in the feedback, we will improve the tool as follows: (1) include visual presentation of the principle of strong and weak tranquility, (2) provide a `Practice` component with answers to questions, and (3) add tooltip to the user interface.

MLSvisual is a part of larger development of security visualization tools supported by the National Science Foundation. Besides MLSvisual, DTEvisual for Domain Type Enforcement access control model has been developed. Visualization tools for Role-Based Access Control model and a large visualization framework for the combination and communication of all the visualization tools will be available in the future. The tool, user guide and demo video are accessible at the following link:

`http://acv.cs.mtu.edu/mlsvisual.html`

6. REFERENCES

[1] D. E. Bell and L. J. La Padula. Secure computer systems: Mathematical foundations. Technical Report MTR-2547, Vol 1, The MITRE Corporation, Bedford, MA, Nov. 1973.

[2] K. J. Biba. Integrity considerations for secure computer systems. MTR-3153, Rev. 1, The MITRE Corporation, Bedford, MA, Apr. 1977.

[3] W. E. Boebert and R. Y. Kain. A practical alternative to hierarchical integrity policies. In *Proceedings of National Computer Security Conference*, pages 18–27, Oct. 1985.

[4] J. R. Crandall, S. L. Gerhart, and J. G. Hogle. Driving home the buffer overflow problem: A training module for programmers and managers. In *Proceedings of National Colloquium for Information Systems Security Education*, June 2002.

[5] D. Ebeling and R. Santos. Public key infrastructure visualization. *The Journal of Computing Sciences in Colleges*, 23(1):247–254, Oct. 2007.

[6] S. Hallyn and P. Kearns. Tools to administer domain and type enforcement. In *Proceedings of USENIX Conference on System Administration*, pages 151–156, Dec. 2001.

[7] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 20(2):38–47, 1996.

[8] D. Schweitzer and L. Baird. The design and use of interactive visualization applets for teaching ciphers. In *Proceedings of IEEE Workshop on Information Assurance*, pages 69–75, 2006.

[9] D. Schweitzer, L. Baird, M. Collins, W. Brown, and M. Sherman. Grasp: A visualization tool for teaching security protocols. In *Proceedings of National Colloquium for Information Systems Security Education*, pages 75–81, 2006.

[10] D. Schweitzer and W. Brown. Using visualization to teach security. *The Journal of Computing Sciences in Colleges*, 24(5):143–150, 2009.

[11] D. Schweitzer, M. Collins, and L. Baird. A visual approach to teaching formal access models in security. In *Proceedings of National Colloquium for Information Systems Security Education*, 2007.

[12] J. Tao, J. Ma, M. Keranen, J. Mayo, and C.-K. Shene. ECvisual: A Visualization Tool for Elliptic Curve Based Ciphers. In *Proceedings of ACM Technical Symposium on Computer Science Education*, pages 571–576, 2012.

[13] J. Tao, J. Ma, J. Mayo, C.-K. Shene, and M. Keranen. DESvisual: A Visualization Tool for the DES Cipher. *The Journal of Computing Sciences in Colleges*, 27(1):81–89, 2011.

[14] X. Yuan, Y. Qadah, J. Xu, H. Yu, R. Archer, and B. Chu. An animated learning tool for kerberos authentication architecture. *The Journal of Computing Sciences in Colleges*, 22(6):147–155, 2007.

Acknowledgements

This work was supported in part by the National Science Foundation under grants DUE-1140512, DUE-1245310 and IIS-1319363.