RBACvisual: A Visualization Tool for Teaching Access Control using Role-based Access Control

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

Thomas Lake,
Steve Carr
Dept. of Computer Science
Western Michigan University
Kalamazoo, MI
{thomas.l.lake, steve.carr}@wmich.edu

Chaoli Wang
Dept. of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN
chaoli.wang@nd.edu

ABSTRACT

This paper presents RBACvisual, a user-level visualization tool designed to facilitate the study and teaching of the role-based access control (RBAC) model, which has been widely used in companies to restrict access to authorized users. RBACvisual provides two graphical abstractions of the underlying specification. Policies can be input and modified graphically or using text-based files. Students can use an embedded Query system to answer commonly asked questions and to test their understanding of a given policy. A Practice subsystem is also provided for instructors to assign quizzes to students; the answers can be sent to the instructor via email. We also present the results of an evaluation of RBACvisual within a senior-level course on information security. The student feedback was positive and indicated that RBACvisual helped students understand the model and enhanced the course.

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

1. INTRODUCTION

Within organizations and companies, it has always been critical, yet challenging to associate privileges and responsibilities with different positions. In the 1970s, computer applications were developed to implement access constraints according to job positions. The role-based access control models were simple and application-specific. The first general-purpose RBAC model was proposed by Ferraiolo and Kuhn [4] in 1992. Based on this model, Sandhu et al. [7] introduced an RBAC framework in 1996. Later, a U.S. national standard for RBAC was proposed and accepted in 2004. Now the model is widely used in modern industry. As the RBAC model gains more and more popularity, understanding the model and using it to design policies to fulfill security goals has become increasingly important.

Visualization has been applied to some access control models. Schweitzer et al. developed a visualization system to enable active learning about the HRU (Harrison, Ruzzo, Ullman) and Take-Grant models of access control [9]. 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 [1, 3, 8, 9, 10, 11]. This paper describes RBACvisual that aims to enhance the pedagogy of the RBAC model. It allows students to create, modify, and analyze policies graphically. Students can practice RBAC policy design without taking time to learn the details of a security specification language. RBACvisual can import and export human-readable text-based policies. Analysis is via three graphical representations of a policy or via a query subsystem. Instructors may use a test module that requires students to answer questions about policies and then sends student answers to the instructor by email. The system is not tied to the underlying operating system and currently runs under Linux and MacOS. RBACvisual was tested in a senior-level course on computer security.

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

RBACvisual was used in a Computer Security course, offered by the Department of Computer Science at Michigan Technological University. It is a senior-level course that

ITICSE'15, July 04-08, 2015, Vilnius, Lithuania.

http://dx.doi.org/10.1145/2729094.2742627.

gives a basic introduction to topics in computer security. The access control component covers the Role-Based Access Control (RBAC), Domain Type Enforcement (DTE), and Bell-LaPadula (BLP) models. The course also covers secure coding in C, cryptography, key management, authentication, malicious logic, and intrusion detection.

Most students were computer science majors who took the course as an elective. The class in which the evaluation was conducted included twenty-seven students.

Students were given paper and pencil exercises on the RBAC model as part of the regular course homework. For this first use of RBACvisual, students were additionally given extra credit if they used the tool to solve their assignment questions. The problem was to evaluate some simple policies via a series of questions. A quiz feature (described in Section 3.5) was used in the take-home final exam. A survey was distributed for students to participate in voluntarily.

3. SOFTWARE DESCRIPTION

Roles

	cust	dev	pres	qc	sales
cathy	X				
charles	X				
dave	X	X			
dot		X			
patty			X		
quinn				X	
sam					X

Objects

	/	/path	/path/to	/path/to/db	path/to/evidenc	/path/to/files	/path/to/tests
cust				x,r,w			
dev							r,w
pres					r,w		
qc							x
sales				x,r,w		-r | r,w	

Figure 1: User Interface (with Matrix View)

RBACvisual is a visualization tool designed to facilitate the study and teaching of the Role-Based Access Control (RBAC) model. It implements the RBAC model in Core and Hierarchical forms [5]. The basic concept of Core RBAC is that users as well as permissions to objects (files and directories) are directly assigned to roles based on their job functions. Therefore, users' membership to roles determines if they have access to objects in the system. Core RBAC allows the user-to-role assignment and role-to-object permission to be many-to-many. Based on Core RBAC, Hierarchical RBAC additionally supports a role hierarchy. A hierarchy is mathematically a partial order defining a seniority relation between roles, whereby senior roles acquire the permissions of their juniors, and junior roles acquire the user membership of their seniors [5]. We denote the senior role as r_1 and junior role as r_2. Let $U(r)$ be the set of users assigned to role r and $P(r)$ be the permissions of role r. We define the inheritance relation $>$ such that $r_1 > r_2$ if and only if $U(r_1) \subseteq U(r_2)$ and $P(r_1) \supseteq P(r_2)$.

RBACvisual supports two types of files: specification files (.rbac) and visualization files (.rbacvis). A specification file contains text that describes role inheritance, user-to-role assignments and role-to-object permissions. A visualization file stores the graphical information of the visualization, and implicitly the underlying specification, so that the same arrangement and layout can be retrieved later. The visualization focuses on the interpretation of the user-to-role and role-to-object relationship combined with the role hierarchy.

3.1 Visualization

Two different views, the Matrix View and the Hierarchy View, are available to examine a policy. Figure 1 has an example of the Matrix View. The top matrix is for the user-to-role assignment and the bottom matrix shows the role-to-object permissions.

Figure 2 shows the Hierarchy View, which consists of two parts. The Role Hierarchy Section with green background constructs a graph based on the role hierarchy. The Object Hierarchy Section with red background shows the hierarchy of objects in the file system. Green nodes represent roles, red nodes represent objects, and yellow nodes representing users are located around their role nodes. An edge is drawn from node r_1 to node r_2 when node r_1 inherits node r_2[1]. All inheritance relationships are extracted from the policy and are depicted by an edge. If the inheritance is not specified explicitly in the policy file, the edge line is dashed.

3.2 Analysis Mode

In the Matrix View in Figure 1, users, roles and objects are shown as headers of the tables. Clicking on a user (*i.e.*, dave) highlights the roles (*i.e.*, cust, dev, qc) this user occupies based on direct assignment and any defined role hierarchy and thus highlights the accessible objects. Likewise, clicking on an object highlights users and roles that have access to it. In the bottom table, the permissions of roles to objects are listed. The content follows the format "-r| permissionset"[2] when the permission applies to the objects underneath and is highlighted in yellow. When the permissions do not apply to objects underneath, the format will simply be "permissionset".

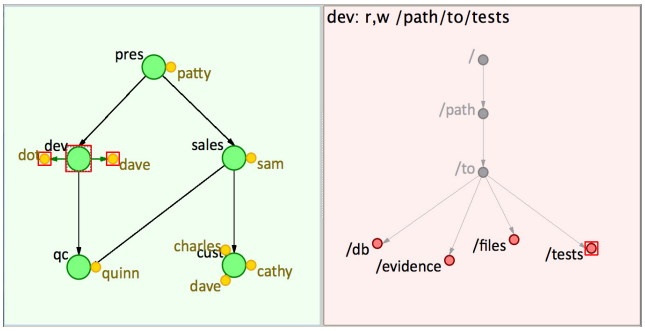


Figure 2: Role Node Highlight without Inheritance

As for the Hierarchy View, clicking on a node of interest will highlight the user and object nodes the role can access. Explicit read, write and execute permissions can be found in the top left corner of the Object Hierarchy Section. When a user node is clicked, the roles to which the user is assigned and the objects that can be accessed through those roles are highlighted. Clicking on an object node will highlight the roles and users that have access to the object.

[1]Edges inferred by transitivity in the role hierarchy are removed to reduce visual clutter.

[2]The permissionset can be any subset of r,w,x where r stands for read, w for write, x for execute. Thus, if granting read and write permissions, the permissionset should be "r, w".

Functions in the Highlight Nodes section in a toolbox (not shown) allow users to configure the highlight scheme of role nodes. Highlighting can be configured to include or exclude the role hierarchy. When the hierarchy is included, highlighting shows the users assigned to the role and the objects the role has access to from itself as well as through the inheritance relation. User nodes are visible by default but can be turned off to reduce the clutter in the graph.

Figure 2 shows an example of clicking on a role node. When the Without role hierarchy option is selected, each of the role nodes can be turned on and off by clicking on it. In Figure 2, role dev is turned on. Therefore, its user nodes dot and dave and object node /path/to/tests are highlighted by red frames while all other nodes are off (gray).

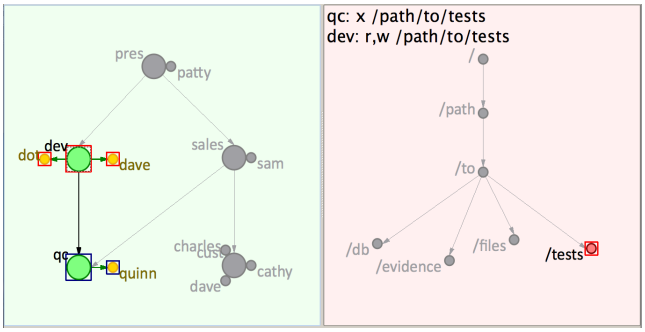


Figure 3: Role Node Highlight with Children

With the role hierarchy turned off, multiple nodes can also be turned on. All users and objects accessible by highlighted nodes will be highlighted. While highlighting a single node provides information directly from the policy specification, highlighting multiple nodes allows a study of the combined permissions of many roles.

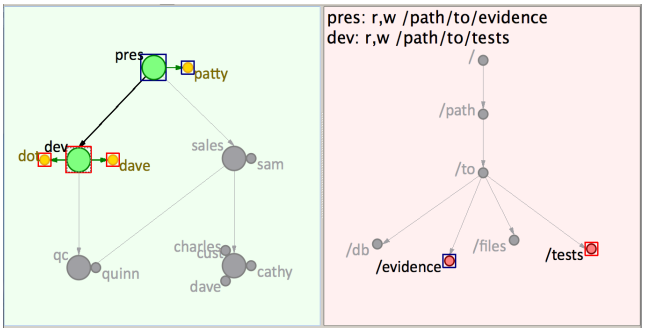
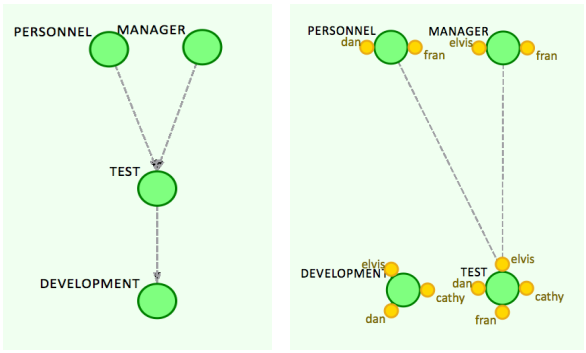


Figure 4: Role Node Highlight with Parents

It is also possible to configure the Highlight Nodes section in the toolbox so that role inheritance is involved. Choices of highlighting the children, parents, or both children and parent role nodes of the clicked role node are available. Different from the mode without hierarchy, this mode only allows one role node to be selected at a time. Along with the selected node, role nodes with the selected inheritance relation will be highlighted in blue frames. User nodes and object nodes will be highlighted in red frames if directly accessible from the clicked role or in blue frames if accessible from blue-framed roles. Figures 3 and 4 depict the nodes related to role dev. The highlighting in the left view shows that the child and parent role nodes of dev are qc and pres, respectively. The right view shows that dev and qc both have access to /path/to/tests with different permissions and pres additionally has access to /path/to/evidence. Likewise, when a user node is clicked, the roles it is assigned to and the objects accessible from those roles will be highlighted. When an object node is clicked, the roles and users that have access to the object are highlighted.

3.3 Edit Mode

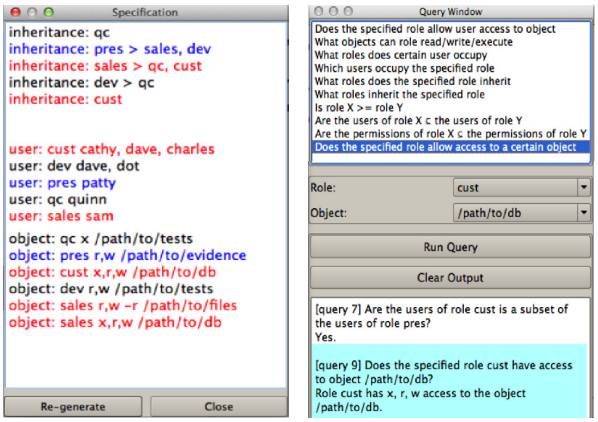


(a) Before Change (b) After Change

Figure 5: Edit mode

Both views allow building a policy from scratch and editing the policy graphically. In the Matrix View, the table cell values can be changed. In the Hierarchy View, a context menu (not shown) can be used for editing the properties of each node. The toolbox provides a dialog to modify, add, or delete any element of the user-to-role or permission-to-role assignment. Addition and removal of any role, user or object are also available. In this mode, any edit applied will cause immediate update of relations and depict the effect. Figure 5 shows the role hierarchy of a policy before and after user assignment to roles. Before the assignment, the role hierarchy was suggested as dashed lines based on the permissions of roles to objects; no users are assigned to roles. After modification, users elvis, cathy and dan are assigned to DEVELOPMENT while elvis, cathy, dan and fran are assigned to TEST. That is, the users of TEST are no longer a subset of the users of DEVELOPMENT. Hence, there is no suggested inheritance relation between them in Figure 5 (b).

3.4 Specification and Query



(a) Specification Diagnosis (b) Query

Figure 6: Specification and Exercise Modules

The Specification Window in Figure 6 (a) shows the text-based specification of the existing policy. It can be edited via graphical operations on views or textual edit within the window. Changes will be reflected immediately. The Query Window in Figure 6 (b) contains questions commonly asked about an RBAC policy. Parameters for certain questions can be configured on the interface and answers to questions can be found in the bottom field, with the most recent answer being highlighted.

3.5 Practice and Test

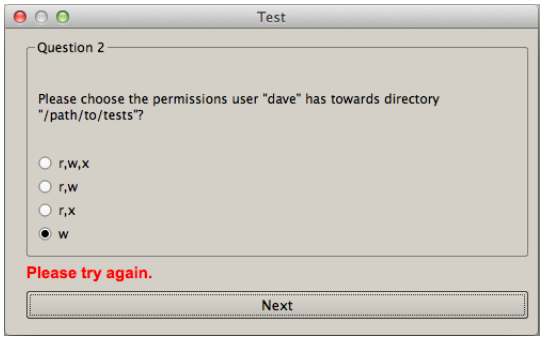


Figure 7: Multiple Trial Quiz Mode with Wrong Answer

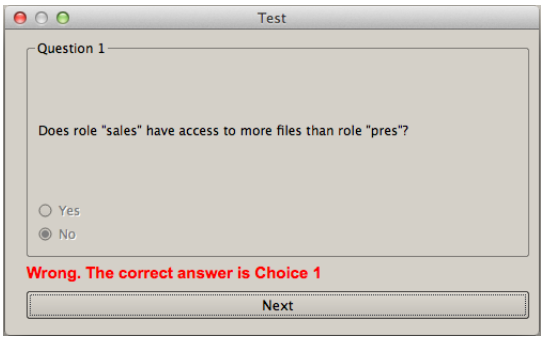


Figure 8: Self-test Quiz Mode with Wrong Answer

RBACvisual allows an instructor to give a series of questions (or a "quiz") to students. Three quiz modes are available which control how a student may progress through the questions. The questions are configurable so that instructors can use their own questions to achieve various teaching goals. All the questions are multiple-choice questions. Instructors specify the quiz mode and the questions that comprise the quiz through a file that adheres to a prescribed format (given in the Instructor Manual). Instructors can share the question file with their students and a test can be started by importing the question file into the system through a dialog.

RBACvisual supports three quiz modes. The first mode is Traditional Mode where students' answers will be sent at the end of the quiz. The quiz moves forward after the first response to each question. The second mode is Multiple Trial Mode. In this mode, students are allowed to try multiple times until they get the correct answer to a question. The number of attempts for each question will be stored. Figure 7 shows an example of the interface when a wrong answer is chosen. The third option is Self-test Mode. Correct answers will be shown to the students after a choice has been confirmed for a question, as depicted in Figure 8.

A dialog confirming the submission will show up as the last step of any test. The system will attempt to bring up Thunderbird to send answers to the instructor. If Thunderbird is not installed, a warning dialog will show up indicating where the answer file is stored and the student will be able to send the email manually. In this case, the answer file will include the student user ID and be encrypted using the instructor's public key. Instructors later can retrieve the readable answer files by applying their private keys. Encryption helps ensure a submitted file was generated by a particular student. With the unique user ID stored and encrypted, it would be difficult for a student to submit a file with the identity of another student. However, students can still take a quiz multiple times (and change their answers) if time allows. The vulnerability of this approach to cheating is similar to a take-home exam. For Multiple Trial Mode and Self-test Mode, the quiz is intended as a practice for students and there is no intention to prevent cheating. Our goal is to let students practice and know the answers for self-evaluation and to let instructors know that a student took the quiz and how the student performed.

4. EVALUATION

Table 1: Rating Questions

Q1	Matrix View helped understand RBAC
Q2	Hierarchy View helped understand RBAC
Q3	Toolbox made it easy to create/edit policy
Q4	Context Menu in Hierarchy View is convenient for policy editing
Q5	Query helped study RBAC policy
Q6	RBACvisual made correct modification on policies easier
Q7	Matrix View was intuitive and clear
Q8	Hierarchy View was intuitive and clear
Q9	Hierarchy View helped understand role inheritance
Q10	Colors used can distinguish different items
Q11	Width of edges was reasonable
Q12	Understood RBAC better after using the tool
Q13	The tool helped find mistakes in my policy
Q14	RBACvisual enhanced the course
Q15	The software was easy to use
Q16	How long did it take you to understand the RBAC model by using the software
Q17	How many times did you use the software
Q18	How long did you use this software in total

The RBACvisual evaluation included two parts: 18 rating questions and seven write-in comments. The rating questions are listed in Table 1. The first 15 rating questions study the effects of RBACvisual. The choices are: 1:strongly disagree, 2:disagree, 3:neutral, 4:agree, and 5:strongly agree. Q16, Q17 and Q18 study the time participants spent on the tool. The choices for Q16 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 Q17 are 1:once, 2:1-3 times, 3:3-5 times, 4:5-10 times, and 5:more than 10 times. The choices for Q18 are 1:less than 5 mins, 2:5-15 mins, 3:15-30 mins, 4:30-60 mins, and 5:more than 1 hour. This evaluation was conducted in a senior-level Computer Security course. For this first use of RBACvisual, students were given extra credit if they

used the tool to solve their assignment questions. A survey was distributed at the end of the semester for students to participate in voluntarily. We collected eight valid forms from students, five of whom major in Computer Science, one in Computer Systems Science, one in Software Engineering, and one in Computer Engineering.

4.1 General Discussion

Table 2 has the means, standard deviations and confidence intervals (at 95% significance level of mean) of rating questions Q1 to Q15. The ratings of questions are no less than 3.88. Their overall mean value is 4.34 with a standard deviation 0.69, suggesting that the feedback to the tool was positive in general. Q6 and Q13 have the lowest mean of 3.88 with standard deviation of 0.99 and 0.64, respectively. Q6 investigates whether the tool makes the correct modification of policies easier. The lower scores it received might be because some modifications did not introduce big changes in visualization and thus some efforts should be taken to examine the changes. Q13 probably shares the same reasoning when changes are applied and it is hard to tell the correctness of a change as it depends on the users' intention, which is hard to detect. The means and confidence intervals of Q7 and Q8 are 4.29 and 4.57, (3.92, 4.65) and (4.18, 4.97), indicating that students generally thought the Matrix View and the Hierarchy View were intuitive and clear. Q1, Q2, Q12 and Q14 received scores no less than 4.13. This suggests that RBACvisual helped students understand the RBAC model better and enhanced the course. Q3, Q4 and Q15 on the easiness of using the tool were rated over 4.25 and thus showed that the tool was easy to use.

Table 2: Mean (μ), Standard Deviation (σ) and Confidence Interval

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
μ	4.38	4.25	4.38	4.25	4.38	3.88	4.29
σ	0.52	0.89	0.74	0.89	0.74	0.99	0.49
CI^-	4.02	3.64	3.86	3.64	3.86	3.19	3.92
CI^+	4.73	4.86	4.89	4.86	4.89	4.56	4.65

	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
μ	4.57	4.86	4.57	4.29	4.13	3.88	4.25	4.63
σ	0.53	0.38	0.79	0.76	0.64	0.64	0.46	0.52
CI^-	4.18	4.58	3.99	3.73	3.68	3.43	3.93	4.27
CI^+	4.97	5.00	5.00	4.85	4.57	4.32	4.57	4.98

Confidence Interval = (CI^-, CI^+)

The last three questions (Q16 to Q18) are about the usage of the tool. Table 3 has the distribution of answers. On Q16, 62.5% of students selected Choice 2 and 37.5% chose Choice 4. This implies that all students were able to understand the RBAC model within 15 minutes. As for Q17, the distribution indicates that half of the students used the tool for one to three times and all of them used the tool for less than 5 times. Answers to Q18 suggest that 75% of the students used the tool for less than 30 minutes while there were some students who used the tool for up to one hour.

Table 3: Usage Distribution

	Choice1	Choice2	Choice3	Choice4	Choice5
Q16	0	62.5%	37.5%	0	0
Q17	12.5%	50%	37.5%	0	0
Q18	12.5%	25%	37.5%	12.5%	12.5%

4.2 Statistical Analysis

We were interested in knowing the rating correlation of each question pair. To this end, the Spearman rank correlation test was applied to the first 15 questions. We found 16 out of the 105 question pairs had a p-value less than the level of significance $\alpha = 0.05$. This means that nearly 85% of the question pairs did not have a significant monotonic correlation. Moreover, all Spearman ρ's between Q10 and other questions were insignificant, meaning the rating of the use of colors is likely to be independent of the rating of other questions. Figure 9 shows the 16 pairs with the value of ρ shown on each edge. It is clear that the ratings of Q2, Q3, Q5, Q6, Q8 and Q11 were very closely inter-related with a Spearman ρ value of at least 0.78. Therefore, the Hierarchy View, Query, easy policy modification, and policy creation/editing were rated similarly in a monotonic way. Q5, Q12, Q13 and Q9 formed a linear chain with ρ(Q5,Q12) = 0.839, ρ(Q12,Q13) = 0.820 and ρ(Q9,Q13) = 0.764. This indicated that if a student rated "Query helped study RBAC policy" (Q5) higher this student would very likely provide higher ratings to questions "RBACvisual helped understand RBAC better" (Q12), "RBACvisual helped find mistakes" (Q13), and "Hierarchy View helped understand role inheritance" (Q9). It is interesting to note that the Spearman ρ between "RBACvisual enhanced the course" (Q14) and "RBACvisual was easy to use" (Q15) is 0.4 with a p-value of 0.374. As a result, we cannot reject the null hypothesis, which means there was no statistically significant monotonic correlation between the rating of Q14 and the rating Q15. On the other hand, the two high Spearman ρ dangling pairs ρ(Q6,Q14) = 0.882 and ρ(Q3,Q15) = 0.794 were perhaps coincidences. We also

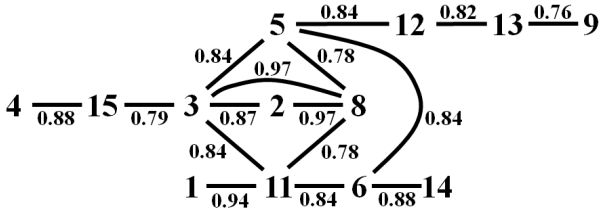


Figure 9: Graph of Significant Spearnman Correlation Pairs

used a Student's t-test to compare the differences among ratings. While the sample size is small, Student's t-test is rather robust and still can be used in this study [2]. We first looked at the "helped" question group (Q1, Q2, Q5, Q9, Q13). Pairwise t-test shows that except for pairs (Q5,Q9) and (Q9,Q13) with p-values 0.03 and 0.00, respectively, all other p-values were larger than 0.1. This suggested that except for (Q5,Q9) and (Q9,Q13), the null hypothesis (that the questions were rated equally) cannot be rejected. The p-value for pair (Q7,Q8) is 0.17, and, hence, students rated the Matrix View and the Hierarchy View equally even though the means were 0.429 and 0.458, respectively. Finally, we looked at three summary questions "Understood RBAC after using the tool" (Q12), "RBACvisual enhanced the course" (Q14), and "The software was easy to use" (Q15). The mean values of Q12, Q14 and Q15 were 4.13, 4.25 and 4.63, respectively, and the p-values for (Q12, Q14), (Q12, Q15) and (Q14,Q15) were 0.60, 0.03 and 0.08, respectively. Therefore, the rating difference between Q12 and Q15 is statistically significant, and students considered ease of use higher than improved understanding of RBAC after using the tool. Since only 13 out of 105 pairwise t-tests were significant and many ques-

tion pairs were not directly related, the rating differences would be small. Coupled with high ratings of questions, we conclude that the evaluation results were very positive for this sample.

4.3 Student Comments

The seven write-in questions were designed to gather suggestions from participants for further improvement. The aspects include: representation in visualizations, the effects of in-class demo of the tool, new feature suggestions, and performance and installation of the tool.

The overall feedback to visualization representations was positive. Some students stated "*I enjoy this view when looking at who has permissions quickly. I can click on what I need to know and it will light up anything corresponding to.*", "*This was the best part. The hierarchy showed the role dominance and which users belonged to which roles very clearly.*", and "*The hierarchy view helped me understand what roles are ranked higher and lower than one another.*" Some issues were mentioned: (1) to add header scrolling in the matrix view; and (2) presentation of permissions to objects that fits the visual theme better than the text presentation.

The in-class demo received neutral feedback. For the students who sent evaluation forms back after final exams, it was hard to remember the in-class demo afterwards, and they generally gave a neutral feedback. For the feedback received on time, the feedback was positive. Students mentioned "*I think the most advantage is [that] I am involved and get a helpful feedback quickly.*", and "*I think the most helpful part is being able to click on elements and see the relations between roles, users and objects.*"

Students also provided some general comments for further improvement. They suggested: (1) the Matrix View should have multiple selections that allow comparisons; (2) Keyboard shortcuts should be supported; and (3) the specification should be directly editable in the Specification Window. No performance or installation issues were reported.

In summary, we found that students who rated "Query helped study RBAC policy" tend to give high ratings to "RBACvisual helped understand RBAC better", "RBACvisual helped find mistakes" and "Hierarchy View helped understand role inheritance". We also found that students rated the Matrix View and the Hierarchy View equally. Combined with the high ratings of questions and students comments, we believe that RBACvisual effectively helped students understand and the instructor teach the RBAC model better with intuitive visual representation.

5. CONCLUSIONS

The paper presents a tool RBACvisual which is designed to facilitate the teaching and self-learning of the Role-Based Access Control model. Students can practice RBAC policy design without taking time to learn the details of a security specification language. They can also take quizzes or run the query subsystem to evaluate their understanding of the RBAC model and an individual policy. Instructors can use the tool during lecture to discuss complex examples and easily demonstrate the effect of policy modifications.

Our evaluation showed that RBACvisual was effective in helping students understand the model better and enhancing the course. The general feedback was positive with mean value of 4.34 and standard deviation of 0.69 for all questions. As suggested in the feedback, we will improve the tool as follows: (1) allowing multiple selections in Matrix View; (2) supporting keyboard shortcuts; and (3) making the Specification Window directly editable.

RBACvisual is a part of larger development of security visualization tools supported by the National Science Foundation. Besides RBACvisual, DTEvisual for the Domain Type Enforcement access control model and MLSvisual for Multilevel Security have been developed. Visualization tools for UNIX, tutorials for each model, and the ability to run programs under a given policy will be available in the future. The tool, user guide and demonstration video are accessible at the following URLs:

`acv.cs.mtu.edu/RBACvisual.html`
`www.vimeo.com/109193019`

6. REFERENCES

[1] 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*, 2002.

[2] J. C. F. de Winter. Using the Student's *t*-test with Extremely Small Sample Sizes. *Practical Assessment, Research & Evaluation*, 18(10):1–12, 2013.

[3] D. Ebeling and R. Santos. Public Key Infrastructure Visualization. *The Journal of Computing Sciences in Colleges*, 23(1):247–254, 2007.

[4] D. Ferraiolo and R. Kuhn. Role-Based Access Control. In *Proceedings of NIST-NCSC National Computer Security Conference*, pages 554–563, 1992.

[5] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed NIST Standard for Role-based Access Control. *ACM Transactions on Information and System Security*, 4(3):224–274, 2001.

[6] S. Hallyn and P. Kearns. Tools to Administer Domain and Type Enforcement. In *Proceedings of USENIX Conference on System Administration*, pages 151–156, 2001.

[7] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.

[8] D. Schweitzer and W. Brown. Using Visualization To Teach Security. *The Journal of Computing Sciences in Colleges*, 24(5):143–150, 2009.

[9] 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.

[10] 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.

[11] 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.

Acknowledgements

This work was supported in part by the National Science Foundation under grants DUE-1140512, DUE-1245310 and IIS-1456763.