UNIXvisual: A Visualization Tool for Teaching UNIX Permissions

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

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

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

ABSTRACT

UNIXvisual is a user-level visualization tool designed to facilitate the study and teaching of access control in UNIX. UNIXvisual is aimed at both novice users, who need only to control access to their own files, and students of computer security, who need a deeper and more comprehensive understanding. The system allows students to analyze permission settings in the underlying real file system, as well as in a combination of real and pseudo file systems defined through a specification file. It also allows a student to trace the value and effect of credentials within an executing process. UNIXvisual gives instructors flexibility in the allocation of lecture time by supporting self-study, lowers the overhead required for teaching access control by running under an ordinary user account, and enhances learning through the use of visualization.

We also present the results of an evaluation of UNIXvisual within a junior-level course on concurrent computing. The evaluation indicated that UNIXvisual helped students understand UNIX permissions and enhanced the course coverage of UNIX permissions, regardless of their prior UNIX experience.

Categories and Subject Descriptors

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

Keywords

UNIX, Security, Visualization

1. INTRODUCTION

Increasing concern about the security of user data has led to the implementation of mandatory access control systems with domain-specific models. Due to the complexity of policy development and maintenance, mandatory systems are often used together with the traditional discretionary UNIX permissions. The mandatory system is used to protect system data while user data continue to be protected with UNIX permissions. Additionally, many administrators choose to use only UNIX permissions, even for the protection of system data. Hence, it is critical that students understand traditional UNIX access control.

In our experience, many students believe that they understand UNIX permissions even though their understanding is incomplete. We believe that this is because of the difficulty of testing the effect of a particular permission bit setting, as it typically requires access to multiple user accounts. Additionally, through our graduate program, we have found that formal coverage appears to vary dramatically.

In order to address these concerns, we developed UNIXvisual, a tool to facilitate education on traditional UNIX access control. UNIXvisual runs from an ordinary user account. It provides several perspectives that help students explore the effect of permission bit settings. It also allows students to track process credentials in their running program. It supports a Query Mode in which students may test their understanding through standard questions. It also supports a Quiz Mode that allows instructors to conduct quizzes through the system outside of classrooms.

The remainder of this paper is organized as follows: Section 2 presents related work. Section 3 presents our tool, Section 4 has a detailed study of our findings from student evaluation, and Section 5 has our conclusions.

2. RELATED WORK

Tools have been developed to address the usability of access control models using a graphical interface. Eiciel [3], as a built-in component of the GNOME file manager, provides an interface that lists users and groups of an opened file and allows the direct click-and-check of ACL properties of that file. Intentional Access Management [1] also supports permission management through an interface and can automatically generate WebDAV policies from user input.

Some tools also leverage visualization to facilitate access control. Expandable Grids [4] shows effective access control using an interactive matrix given a policy. DTEEdit and DTEView [2] by Hallyn and Kearns illustrate an input DTE

ITiCSE '17, July 3–5, 2017, Bologna, Italy.

DOI: http://dx.doi.org/10.1145/3059009.3059031

policy as an interactive graph to help policy analysis. While visualization methods were used to help teaching many security fields and other access control models [5], we failed to find tools developed for the presentation and teaching of the UNIX permissions.

3. SOFTWARE DESCRIPTION

UNIXvisual requires a user-defined root directory to begin. The root directory defines the starting point at which the data from the underlying file system is extracted. The root directory can be specified directly in UNIXvisual or through the import of a specification file. A specification file allows the user to define a hypothetical file system (including permission settings) that overlays the underlying file system, users and groups.

The visualization illustrates the process of determining the access a user or group has to objects. UNIXvisual also uses visualization to help students monitor and control process credentials, e.g., for set-user-id and set-group-id programming within C programs. This part of visualization displays the executing sequence of system calls and shows the success and failure of the calls along with real and effective UIDs and GIDs. UNIXvisual also provides a query and quiz subsystem that leverages the visualizations. These features are described in more detail below.

3.1 Perspectives

UNIXvisual supports four main perspectives. The Decision Mode View illustrates a single decision by the access control system, excluding directory traversal. The Object View explores which users and groups have access to a selected object. The User View and Group View explore the set of objects accessible to a user (through the user bits) or to a group (through the group bits) respectively. Finally, the Program Trace View allows a student to trace the value and effect of process credentials within their running program.

3.1.1 *Decision Mode*

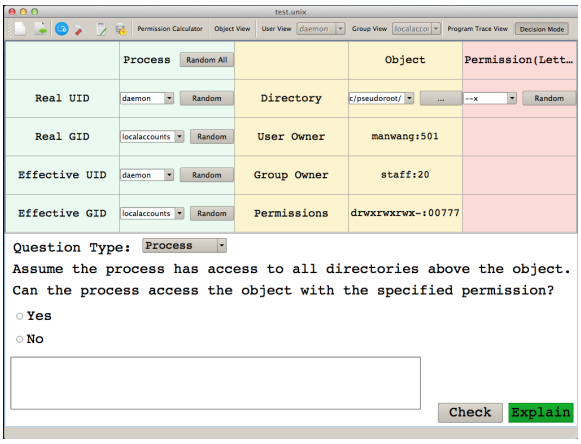


Figure 1: Decision Mode

The Decision Mode aims to provide obvious access to an interactive question system in order to encourage students to test their understanding. The interface has two parts (Figure 1). Parameters for the questions are configured in the top part. One of several questions can be chosen and the according choices are displayed in the bottom part. This mode provides two types of commonly asked questions: 1) whether a process with certain credentials can access an object with a selected permission, and 2) conversion between the letter and octal permission notations. Students can answer the question and click on the "Check" button for the correct answer. If further explanation is needed, clicking on the "Explain" button will initiate an animation to guide students through the solution.

3.1.2 *Object View*

The Object View asks for an object of interest and illustrates the determination of which users have access to the object. Figure 2 shows a snapshot of the Object View. The top-left UI section asks for the necessary information such as the path of an object, a user and its group to perform the analysis. The visualization shows the analysis in a matrix form. On the left, paths from the root directory to the target object are represented as nodes with permission information at each directory level. This defines the rows of the matrix. On the right, the permission bit groupings, "Owner bits", "Group bits" and "Other bits", define the columns.

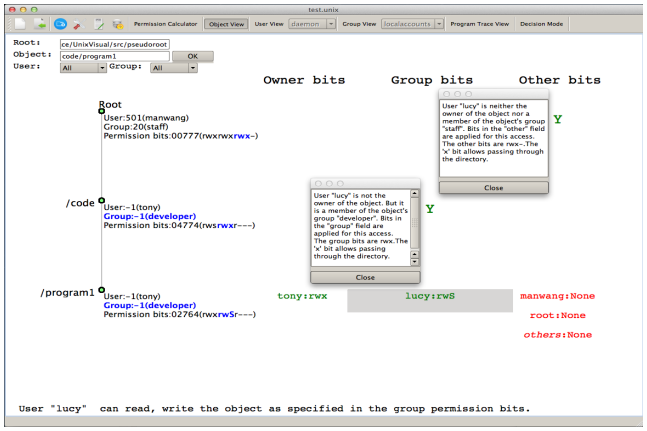


Figure 2: Object View

Students may choose an individual user or group, or they may use the (default) wildcard All option. In the case where all users are considered, results of multiple users' access are shown in the last row as color-coded user names. The columns in which a user name appears indicate the bits applied at the object level. Clicking on a user name enables an analysis of the user's access. Color-coded letters of "Y" and "N" are placed in the row which corresponds to object level, and the column which corresponds to the group of bits that are applied. In Figure 2, the user has selected lucy from the last row to obtain more information on the access lucy has to the object. Clicking on the letters of "Y" and "N" allows another level of detailed explanation of why these bits are applied and why the access is or is not allowed. This triple-layered analysis from color-coded user names to detailed explanation avoids showing complete explanation all at once, and thus encourages students to think about how the permissions work.

3.1.3 *User and Group View*

The User and Group View illustrates the access allowed by a user or group through the file permission bits to objects under a user-specified directory. An example of the User View is given in Figure 3. The visualization can be divided into two parts. The left part contains information about a user (above) or group. A user is represented as a node connected with three nodes to represent the owner, group

and other permission bits. A group is represented as a node connected with all its member users (not shown). The right part has four sections. The top-left window is the Permission Setting section. In this section, students may choose the type of object access they want to investigate. The bottom-left window is the Directory Tree section. It shows the object structure in a standard directory tree hierarchy. Directories are clickable to expand and contract for one directory level.

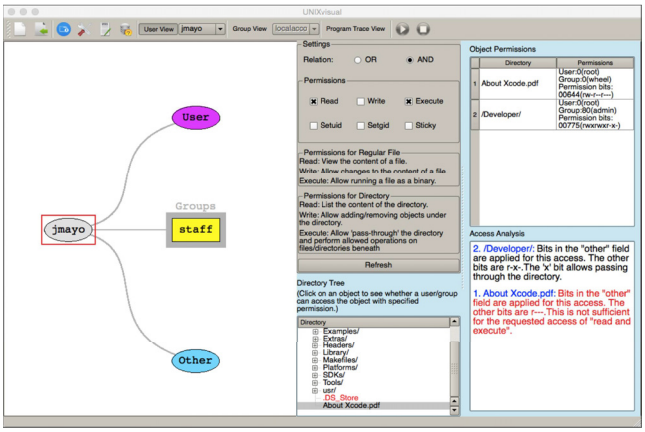


Figure 3: User and Group View

The two windows on the right are the Object Permissions and Access Analysis sections. They are blank initially. Once a user/group is selected, each object in the object hierarchy window is checked against the specified permission in the Permission Setting section. If an object can be accessed by the user with the specified permission, the object remains black. Otherwise, it is shown in red. Clicking on an object in the Directory Tree enables the Object Permissions and Access Analysis windows which show a detailed analysis. The permission information for the object selected from the Directory Tree and all directories up to the root directory is supplied in the Object Permissions section. An explanation of the access is given in the Access Analysis section. The analysis includes an explanation of which bits were applied and the access decision at the level.

3.1.4 Program Trace View

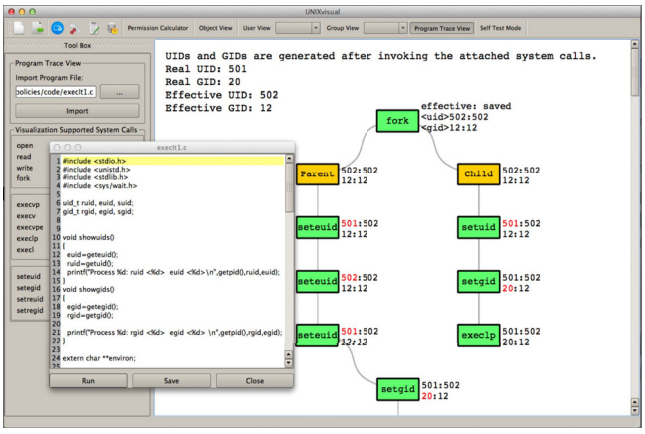
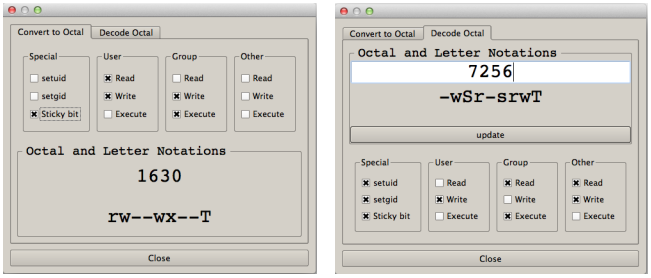


Figure 4: Program Trace View

The Program Trace View is designed to help students understand initial assignment of credentials to a process, dynamic modification of credentials, and the effect of these credentials on an access request. This view allows the import of a C program and tracks process credentials across access control-related system calls, like open, fork, setresuid, read, write, etc. Figure 4 shows an example of the visualization. After loading a C source code or binary, the initial real and effective user/group IDs are shown in the top left corner. Invoked system calls are depicted sequentially as blocks with effective and saved user/group IDs. The success of a system call is reflected in its block color: green indicates success and red indicates failure. The credentials on the side use red highlighting to indicate changes in credential values after the system call.

3.2 Permission Calculator

Octal and letter notations are frequently used to specify UNIX permissions values through the command line. The conversion between these two notations can be tricky for beginners. The Permission Calculator is designed to help students learn different permission notations. Figure 5 (a) and (b) show the interface of the letter-to-octal and the octal-to-letter notation conversion. Both interfaces have three ways of expressing permissions: a matrix of checkboxes denoting permission bits, octal notation and letter notation. With the checkbox matrix and both notations side by side, it is easier to interpret the meaning of each bit and how each bit is expressed in different notations.



(a) Letter to Octal Notation (b) Octal to Letter Notation

Figure 5: Permission Calculator

3.3 Query and Quiz

UNIXvisual also contains a Query Mode and a Quiz Mode. The Query Mode includes a list of commonly-asked questions on UNIX permissions. Question parameters are configurable through the interface and answers to the questions are presented through guided visualization. This mode provides the convenience of having problems clarified outside of the classroom at any time. The Quiz Mode provides an interactive environment for conducting quizzes. Text-based and visualization-based questions can be asked. All the questions are multiple-choice questions and can be configured to accommodate instructors' teaching goals. The questions that comprise a quiz are written through a text file that adheres to a prescribed format. Students can start the quiz by loading the question file distributed by the instructor. Each question will have to be answered before moving to the next one. At the end of the quiz, a dialog will show the location of the student's answer file and the student will be able to send the instructor an email which prevents manual changes.

4. EVALUATION

4.1 Environment, Procedure and Goals

The evaluation was conducted in a required junior-level Concurrent Computing course with a total of 55 students.

In a 75-minute session, students were asked to take a pre-test on UNIX permissions, followed by a 35-minute UNIX permissions lecture and a 15-minute demo of UNIXvisual. Students were allowed to use UNIXvisual in the following two weeks, and completed a post-test and an evaluation form. The pre-test and post-test plus evaluation are treated as quizzes. UNIX permissions is a standard topic in Concurrent Computing for the shared memory component, and there was no mechanism to enforce the use of the tool. Additionally, it is unfair to divide the students in this class to treatment and controlled groups so that one group of students would not use the tool. Therefore, we can only compare the performance between groups of students who only attended the lecture and who both attended the lecture and used the tool. The sample sizes of students who used the tool without attending the lecture and who neither used the tool nor attended the lecture are too small to be used for a meaningful statistical analysis.

We collected 40 valid pre-tests, 44 valid post-tests, and 44 valid evaluation forms. We also collected 51 final exam papers, and recorded grades of the UNIX permissions section. Of the 44 students who submitted the evaluation form, 21 used UNIXvisual, 40 attended the lecture, and 38 submitted the pre-test, post-test and final exam. The participants who used UNIXvisual majored in Computer Science (13 students), Software Engineering (5 students), and Computer Engineering (3 students).

4.2 Test Problems

The questions in the pre-test, post-test and final exam have the same form with the same level of difficulty. There are 10 questions in each test (1 point per question). Questions Q1 and Q2 (Group 1 or G_1) convert between the octal and letter notations of UNIX permissions. Questions Q3-Q6 (Group 2 or G_2) ask about access requests to an object without directory traversal. Questions Q7-Q10 (Group 3 or G_3) are about the access requests to an object with directory traversal. All these test problems are available through the link at the end of this paper.

Table 1: The Means (μ) and Standard Deviations (σ) of the Pre-test, Post-test, and Final Exam Questions

Pre-test											
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
μ	0.80	0.78	0.98	0.88	0.88	0.80	0.34	0.88	0.37	0.44	7.05
σ	0.40	0.42	0.16	0.33	0.33	0.41	0.48	0.33	0.49	0.50	1.66
Post-test											
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
μ	0.98	0.95	0.95	0.89	0.91	1.00	1.00	0.98	0.68	0.75	9.09
σ	0.15	0.21	0.21	0.32	0.29	0.00	0.00	0.15	0.47	0.44	1.29
Final Exam											
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
μ	0.92	0.90	0.88	1.00	0.96	0.98	0.69	0.88	0.61	0.78	8.61
σ	0.27	0.30	0.33	0.00	0.20	0.14	0.47	0.33	0.49	0.42	1.60

Table 2: The Means (μ) and Standard Deviations (σ) of G_1, G_2 and G_3 in the Pre-test, Post-test and Final Exam

	Pre-test			**Post-test**			**Final Exam**		
	G_1	G_2	G_3	G_1	G_2	G_3	G_1	G_2	G_3
μ	0.79	0.88	0.51	0.97	0.94	0.85	0.91	0.96	0.74
σ	0.41	0.32	0.50	0.18	0.24	0.36	0.29	0.21	0.44

Table 1 and Table 2 have the mean and standard deviation of each question and question group in these tests. The correctness of questions in G_1 and G_2 are above 91% in the post-test and the final exam, and G_3 in all three tests has the lowest means and the highest standard deviations. It is reasonable that G_3 received the lowest means as access request questions with directory traversal include more levels of permission checking and thus make the questions more challenging. Figure 6 depicts the group comparison of the three tests. It shows that 1) students' overall performance in all groups improved in the post-test and the final exam; and 2) students performed better in G_1 and G_3 in the post-test than in the final exam.

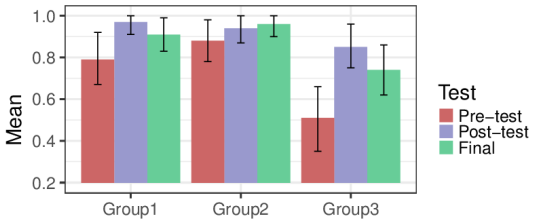


Figure 6: The Means with Confidence Intervals of G_1, G_2 and G_3 in the Pre-test, Post-test and Final Exam

Table 3 has the means and standard deviations of question scores of students who used UNIXvisual and who did not use UNIXvisual in all three tests. Students who used UNIXvisual received higher scores in all question groups in the post-test and the final exam than students who did not use UNIXvisual.

Table 3: The Means (μ) and Standard Deviations (σ) of G_1, G_2, G_3 and Total Scores of Students Who Used and Did Not Use UNIXvisual

Students Who Used UNIXvisual												
	Pre-test				Post-test				Final			
	G_1	G_2	G_3	Total	G_1	G_2	G_3	Total	G_1	G_2	G_3	Total
μ	0.84	0.86	0.46	6.95	0.98	0.98	0.92	9.52	0.95	0.99	0.88	9.35
σ	0.37	0.35	0.50	1.58	0.15	0.15	0.28	0.81	0.22	0.11	0.33	1.23
Students Who Did Not Use UNIXvisual												
	Pre-test				Post-test				Final			
	G_1	G_2	G_3	Total	G_1	G_2	G_3	Total	G_1	G_2	G_3	Total
μ	0.76	0.90	0.54	7.29	0.96	0.90	0.79	8.70	0.89	0.94	0.65	8.13
σ	0.43	0.30	0.50	1.65	0.21	0.30	0.41	1.49	0.32	0.25	0.48	1.65

4.3 Test Problems Analysis

In this part, significance tests were applied to find out 1) whether students' performance in the tests improved; and 2) whether UNIXvisual introduced the improvement. The significance tests include Student's t-test, ANOVA (parametric) and Kruskal-Wallis (KW) test (non-parametric), and repeated measures ANOVA (parametric) and Friedman test (non-parametric). We mainly used parametric methods with the non-parametric methods as backups. All significance tests were conducted at 95% significance level. The p-values below are from parametric tests and their non-parametric counterpart, and they agree on the test results.

To evaluate students' performance throughout the tests, we first compared the pre-test and post-test using Student's t-test and KW test. Only Q1, Q2, Q6, Q7, Q9, Q10 and the total score had p-value less than 0.05. With their increased means from the pre-test to the post-test (Table 1), this indicates that the students' performance on notation conversion, access requests with directory traversal, and the total score had significantly improved. We also applied Student's t-test and KW test to compare the pre-test and the final exam. The results show that students performed differently in only Q4, Q6, Q7, Q9, Q10 and the total score. As the means of these questions increased (Table 1), the performance significantly improved on questions of access request

without and with directory traversal and the total score in the final exam. The scores of the post-test and the final exam were also compared using the same method. The results indicate significantly improved performance in Q4 and declined performance in Q7, which means that the performance in other questions and the total score did not differ significantly. Therefore, for the declined performance in G_1 and G_3 from the post-test to the final exam in Figure 6, we know that the performance decline in G_1 is insignificant, and that Q7 is the only question showed a declined performance in G_3. Note that G_3 has the most challenging questions in the tests. Since there was no homework or project on UNIX permissions between these two tests, the declined performance in Q7 was likely caused by students' less familiarity with the material over time due to a lack of practice.

To investigate the reason for the improvement throughout the tests, we looked into the students who submitted all three tests. As they participated in the UNIX permissions lecture and the UNIXvisual demo, and the use of UNIXvisual, this student group forms an important sample to assess the effect of the lecture with demo and the use of UNIXvisual on the scores of the tests. The repeated measures ANOVA and Friedman test were used, and the p-values of Q1, Q5-Q10, and the total score are less than 0.05. As the means of the total score of the post-test and the final exam are higher than that of the pre-test (Table 1), the lecture with demo and the use of UNIXvisual helped students perform better in the post-test and the final exam.

We further examined whether the use of UNIXvisual helped the improvement in the post-test and the final exam. We applied Students' t-test and KW test to the post-test question group scores of students who used UNIXvisual (21 students) and students who did not use the tool (23 students). The results show that G_2, G_3 and the total score had p-value less than 0.05. With their means in Table 3, the performance of students who used UNIXvisual is significantly better than those who did not use the tool. We also divided students who took the final exam into a group of 20 students who used UNIXvisual and a group of 31 students who did not use the tool, and compared their performance using the same tests. G_3 and the total score had p-value less than 0.05. Given their means in Table 3, students who used UNIXvisual performed significantly better than those who did not use the tool in G_3 and the total score. Lastly, we evaluated the background of students who used UNIXvisual and who did not use the tool by comparing their pre-test scores. The t-test and KW test show that the p-values for all question groups and the total score are greater than 0.05. Therefore, these two groups of students had similar background.

So far we have seen that UNIXvisual helped students improve significantly from the pre-test to the post-test, and that the improved performance continued in the final exam. Students who used UNIXvisual and those who did not use the tool had similar UNIX permissions background. But students who used UNIXvisual made significant improvement and received higher scores in all question groups in the post-test and the final exam than students who did not use the tool. More specifically, the use of UNIXvisual significantly increased the scores of G_3 in the post-test and the final exam. This suggests that UNIXvisual is very effective in helping students understand the access to objects with directory traversal, which forms the most difficult questions in the tests.

4.4 Evaluation Form

We used a set of questions (Table 4) to collect information on students' perception of the effectiveness of the tool. We also gathered information on the time spent on using the tool and the students' major. The first 12 rating questions study the effectiveness of UNIXvisual. Q1 and Q2 examine the overall effectiveness; Q3 and Q4 relate to the two views that show object permissions without and with directory traversal; Q5 and Q6 are about the views that interpret permissions from the perspective of a user or a group; Q7 and Q8 examine the Permission Calculator; Q9 is about the Query; Q10, Q11 and Q12 are about the interface design. The choices are: 1:strongly disagree, 2:disagree, 3:neutral, 4:agree, and 5:strongly agree. Q13 and Q14 study the time participants spent on the tool. The choices for Q13 are 1:once, 2:twice, 3:3-4 times, 4:5-10 times, and 5:more than 10 times. The choices for Q14 are 1:less than 5 mins, 2:5-14 mins, 3:15-29 mins, 4:30-60 mins, and 5:more than 1 hour.

Table 4: UNIXvisual Rating and Usage Questions

Rating Questions	
Q1	UNIXvisual helped to better understand UNIX permissions
Q2	UNIXvisual enhanced UNIX permissions course coverage
Q3	Decision View helped to understand which users have access to a certain object and why
Q4	Object View helped to understand which users have access to a certain object and why
Q5	User View helped to understand how decisions are made to the access request from a particular user
Q6	Group View helped to understand how decisions are made to the access request from a particular group
Q7	Permission Calculator was helpful for understanding the meaning of each bit in UNIX permissions
Q8	Permission Calculator was helpful for understanding how to specify permissions for an object
Q9	Query was helpful for understanding UNIX permissions
Q10	The use of colors in the visualization is effective
Q11	The size of items is reasonable and clear
Q12	The layout of items is reasonable and clear
Usage Questions	
Q13	How many times did you use UNIXvisual
Q14	How long did you use UNIXvisual in total

Table 5: The Means (μ), and Standard Deviations (σ) of UNIXvisual Evaluation Questions

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
μ	3.81	4.05	4.00	3.81	3.85	4.10	4.43	4.29	3.89	3.81	3.71	3.29
σ	0.60	0.50	0.58	0.91	1.09	0.72	0.81	0.85	0.60	0.75	0.72	0.90

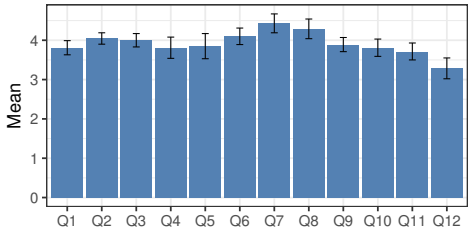


Figure 7: The Means with Confidence Intervals of UNIXvisual Rating and Usage Questions

Table 5 and Figure 7 have the means and standard deviations, and the means with confidence intervals of UNIXvisual rating and usage questions, respectively. All questions except Q12 received a mean greater than 3.7. Students generally believed that UNIXvisual helped the understanding of UNIX permissions and enhanced the course coverage. Permission Calculator received the highest rating (Q7, Q8). The layout received the lowest score (Q12). The reason may be,

as mentioned in a student comment, due to the Object View not being scaled properly. Some text overlap was reported. The means of Q13 and Q14 are 1.95 and 2.43, indicating that students generally used UNIXvisual twice for 10 minutes in total. We used the middle point of a range for estimation.

4.5 Evaluation Form – Student Comments

The four write-in questions were used to collect information of the participants' major, their thoughts on the most and least useful features of the tool, features to add, and problems with installation.

Of the 21 students who used UNIXvisual, 15 students considered the Permission Calculator as the most useful feature. One student wrote that *"the permission calculator can be quite useful to ensure you know how the permissions will look for some file"*. Four students favored the User/Group View. Other two stated the overall features of *"being able to actually check access to a file and check different scenarios"* and *"the instant feedback of whether something works or not with quick explanation"* as the most useful. As for the least useful feature, 16 students did not state any, and three students answered the Permission Calculator. One student mentioned that *"I personally am familiar with permissions, so the calculator was not as helpful"*. Therefore, while 71% of the students considered Permission Calculator as the most useful feature, it is also considered as the least useful one due to the familiarity to the notation conversion on those students' part. Another student considered the Object View the least useful as the view did not scale properly and texts had some overlap.

All students did not encounter any installation problem. When asked to suggest features to add, students were content with the available features. They wrote *"I think it is very well designed. Nothing needs to be added"*, and *"the software was very friendly at aiding further learning and understanding of UNIX permissions the way it currently is"*. There are also some comments for further improvements. Students suggested to add *"something to detect if your files are visible by anyone else"*, and *"having a video tutorial on how to use the software"*.

4.6 Summary

UNIXvisual was evaluated in a classroom setting. Students were introduced the basic knowledge of UNIX permissions including the octal and letter notations, the access to an object with and without directory traversal. Students' average scores went from 7.05 in the pre-test, to 9.09 in the post-test, and 8.61 in the final exam. The final score calculation includes an additional of 11 students who did not attend the lecture and demo. We found significant improvement in students' performance in the post-test and the final exam. We also found that the use of UNIXvisual is very effective in helping to understand the more challenging permissions to an object with directory traversal.

The feedback to UNIXvisual was positive. The Permission Calculator was rated the highest, and was also considered as the most useful feature. The layout received the lowest rating. This may be due to the fact that the Object View could not scale properly on some monitors. No installation issue was reported. Most students believed that the software provides what is needed and is user-friendly. We plan to use the same materials in the following years and other classes so that UNIXvisual could be evaluated with more extensive and multi-year samples.

5. CONCLUSIONS

The paper presents UNIXvisual which is designed to facilitate the teaching and self-learning of UNIX permissions. Students can practice UNIX permissions configuration on the basis of a real as well as a hypothetical file system. They can examine the result of their permission bit setting through visualization, and evaluate their understanding of the model. Instructors can use the tool to teach the UNIX permissions, easily demonstrate steps to solve in-class questions and conduct quizzes. The tool can also demonstrate how process credentials are established and modified.

From the tests conducted in the evaluation process, students showed significant improvement in the tests taken after the use of UNIXvisual. UNIXvisual is very effective in helping to understand the more challenging permission to objects with directory traversal. Our evaluation showed that the feedback was positive. Students believed that UNIXvisual helped them understand the UNIX permissions better and enhanced the course coverage of UNIX permissions. We received suggestions on improving the tool and will incorporate them in the future.

UNIXvisual is a part of larger project to develop access control visualization tools that is supported by the National Science Foundation. In addition to UNIXvisual, DTEvisual for the Domain Type Enforcement access control model, and MLSvisual for the Multilevel Security, and RBACvisual for the Role-based Access Control have been developed. The tool, UNIX permissions slides, pre-test, post-test and final exam problems, and evaluation form can be downloaded at the following URL:

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

6. REFERENCES

[1] X. Cao and L. Iverson. Intentional access management: Making access control usable for end-users. In *Proceedings of the Second Symposium on Usable Privacy and Security*, SOUPS '06, pages 20–31, New York, NY, USA, 2006.

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

[3] R. F. Ibáñez. Eiciel website, GNOME file ACL editor. http://rofi.roger-ferrer.org/eiciel, 2015.

[4] R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, K. Bacon, K. How, and H. Strong. Expandable grids for visualizing and authoring computer security policies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1473–1482, New York, NY, USA, 2008.

[5] D. Schweitzer, M. Collins, L. Baird, U. States, and A. F. Academy. A visual approach to teaching formal access models in security. In *Proceedings of the 11th Colloquium for Information Systems Security Education*, CISS '07, pages 69–75, 2007.

Acknowledgements

This work was supported in part by the National Science Foundation under grants DUE-1140512, DUE-1245310, IIS-1456763, IIS-1455886, and DGE-1523017.