Reading Response 4:

CSEC 759 - Mehul Sen

Yee [1] identified several guidelines for ensuring secure interaction design. He presented five authorization guidelines that limit the likelihood of undesired events by controlling the authorization, as well as five communication guidelines that ensure good communication between the user and the system. These guidelines can be used to ensure that a system correctly interprets a user's desires and that a user understands how the system works. In their paper, Maxion and Reeder [2] attempted to improve user-interface dependability and reduce human error when interacting with file permission interfaces. They sought to understand why human errors in file-permissions interfaces occur and how the design of the user interfaces for these file permissions can be modified to mitigate these errors. To achieve this, they initially evaluated the existing Windows XP file permissions interface in a lab study involving twelve participants who were tasked with modifying permissions for users and groups in the file permissions interface under certain constraints. Following this, they used the design principle of external subgoal support to create a new interface called Salmon. Another lab study was conducted with twelve participants assigned the same tasks and constraints. The results showed that Salmon outperformed the traditional XP File permissions interface, with a 94% reduction in goal errors and a much higher task success rate.

By using the guidelines pointed out by Yee, we can categorize the changes made in Salmon that might have led to its success and reduction in goal errors. Four guidelines primarily stand out within the Authorization category: "4. Maintain accurate awareness of others' authority as relevant to user's decision." Salmon provides a comprehensive "set permissions" section that allows the user to control not only their permissions but also the permissions of the groups and other users within the group. The "effective permissions" section provides visual feedback to ensure the users know their actions and inputs on the interface. Another guideline is "5. Maintain accurate awareness of user's authority to access resources." Salmon shows which permissions can be controlled and modified by the user. For example, in Figure 2, the user can modify Tux, ProjectE, and Jack's permissions. However, they cannot modify "GRP Everyone" and "USR" permissions, which are shown in the "effective permissions" section, since they have an impact on the system but not on the "set permissions" section. Moreover, within the Communication category, Salmon follows guideline "7. Enable the user to express safe security policies in terms that fit the user's task." It does so by providing a marked "EFFECTIVE PERMISSIONS FOR" section at the bottom of the window. The task that a user would have to open the file-permissions interface would be to modify/update permissions for an entity. By indicating which permissions are applied, Salmon can express the security policies in a way that aligns with the user's task. Lastly, the guideline "10. Indicate the consequences of decisions that the user is expected to make." is also followed due to the "EFFECTIVE PERMISSIONS FOR" section, which lets the users know which of their actions results in the modification/override of which prior permissions and the consequences to other groups/users.

Although Salmon makes considerable improvements to the file-permission interface, there are ways it can be further improved. One such suggestion is based on the guideline "8. Draw distinctions among objects and actions along boundaries relevant to the task." When viewing the Salmon interface, users are shown all 13 atomic permissions, even though the majority of the tasks they will perform will have nothing to do with several of these permissions. This information overload could lead to additional work and confusion for the user, as seen in Figures 5 and 6, where the user has to click the corresponding permission labels in the "Set perms" section. Users

spent much more time on Salmon than on the traditional interface. A possible solution would be to show only relevant information to the user. This can be done by researching the most used permissions, such as Execute, Read, Write, Delete, and Administrate, and displaying only those while hiding the more advanced permissions that are only accessible to users who have sufficient knowledge about the system to locate them. Moreover, for the "set permissions" section, instead of being able to modify permissions for all users and groups, having the ability to modify permissions for the current selection (either a user or a group) while being able to toggle through the other selections would make the interface more manageable. Lastly, instead of showing all effective permissions for every category that might be impacting the user's permission, we can employ Yee's design strategy for security by admonition by providing notifications where the current selection permission is not the effective permission and which category (user/group) it is being overridden by.

One way to test the hypothesis and improvements suggested is by conducting a lab study that compares the existing Salmon interface to a modified version with the decluttered and improved interface. Participants would be randomly assigned to one interface and asked to complete several permission-setting tasks, as done in the original Salmon study. The data collected in this study would include the task success rate, which measures how many participants succeeded in completing their tasks, the task time, which measures the duration it took for participants to understand the interface and complete the tasks by observing how long each participant takes after given access to the interface, and the errors made by the user, which show if hiding some information has any adverse effect on the interface by observing the participants. Additionally, user feedback would be collected through questions about the interface to help assess whether they felt more comfortable using it.

References

[1]: K.-P. Yee, "Guidelines and Strategies for Secure Interaction Design," 2005

[2]: R. A. Maxion and R. W. Reeder, "Improving user-interface dependability through mitigation of human error," *International Journal of Human-Computer Studies*, 2005