DirViz: Interactively Scaled Treemaps for File Permission Visualization

Jared Chandler*
Tufts University

Lane Harrison†
Worcester Polytechnic Institute

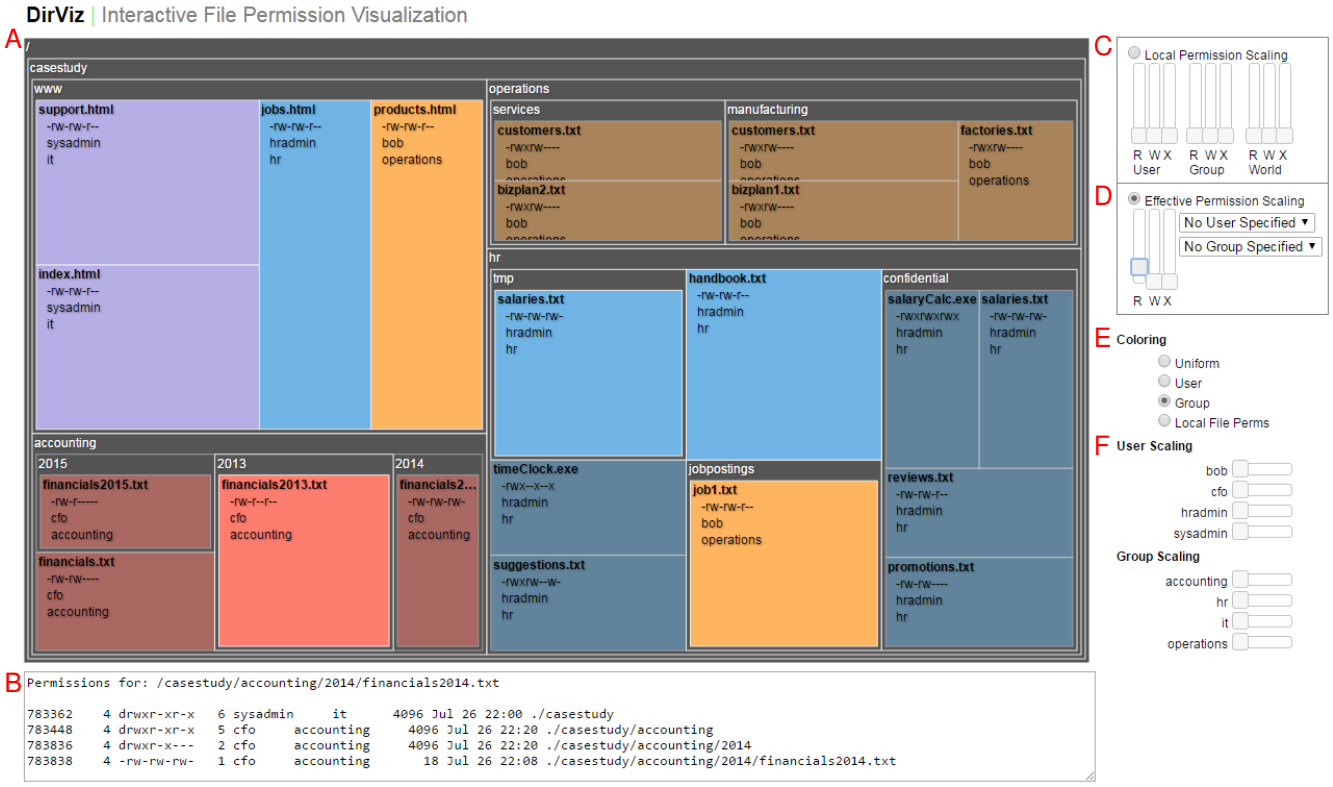

Figure 1: **DirViz Interface** – File Tree Visualization (A), Coordinated Permission View (B), Local Permission Scaling (C), Effective Permission Scaling (D), Color Coding (E), User and Group Scaling (F).

ABSTRACT

File permissions present one of the first lines of defense for computer systems against unauthorized access and misuse. According to Schneier "...security is only as good as its weakest link, and people are the weakest link in the chain" [5]. For many users– including systems administrators and security analysts– the correct use of file permissions remains challenging and mentally taxing.

Limitations of existing tools make it difficult for users to get an overview of the file system permissions [2, 4]. Users that encounter permissions need a tool which accomplishes three goals. First, it should present permission information in context of the file tree hierarchy. Second, it should have an intuitive interface. Third it should promote exploration. To address these limitations and provide users with a way to visually inspect and reason about permissions, we introduce DirViz[1].

*e-mail: chandler@eecs.tufts.edu
†e-mail:ltharrison@wpi.edu

[1]https://www.eecs.tufts.edu/∼chandler/dirvizdemo

DirViz uses a squarified treemap metaphor to show directories and files, encoding permission attributes of interest as the size and color of the treemap rectangles. To support a variety of use cases, DirViz includes interaction techniques and animated transitions to allow users to steer the emphasis of the treemap towards the permission attributes most relevant to their tasks.

The primary goal of DirViz is to support the discovery, analysis and management of Unix file permissions for a file tree. Specifically, the main tasks that DirVis seeks to support are:

- Identifying places where effective permissions differ from the analysts mental model.
- Discovering how permissions for a specific user or group are distributed throughout the hierarchy of the file tree.

Why Squarified Treemap? Squarified treemaps encode hierarchical data as cell size [1]. Larger data values are represented as visually prominent rectangles. This visualization invites comparison between rectangles at the same level within the hierarchy in addition to comparisons between different levels. Further there is a natural mapping between the hierarchy of the file system and that of the elements in the visualization. This correspondence is advantageous to the analyst in orienting themselves within the file system

while using the tool.

Encoding Permission Attributes as Size We developed the following approach to calculating the cell size for a given file. The general algorithm we use is shown in Algorithm 1.

Algorithm 1 Calculate Cell Size from Permissions

```
1: procedure PERMTOSIZE(filePerms, permScales)
2:     size ← 1
3:     for permission in filePerms do
4:         if permission is set then
5:             scalePower ← permScales[permission]
6:             scaleValue ← 2^scalePower
7:             size ← size * scaleValue
8:     return size
```

This function return a value of 1 for all files at initialization. When visualized as a treemap, all cells are uniformly sized. This is shown at the top of figure 2. As the user increases the scaling constants, files with the appropriate permissions yield higher values. When displayed in the treemap, their cells are larger than their counterparts. This is shown in the middle and bottom of figure 2. For example changing the slider for *World Read* to a value of 3 will scale elements which have the world read permission set by a scaling value of 2^3. The total scaling factor for a cell is the product of the scaling values for all permissions which are set. This approach handles both local and effective permissions without modification. We use the same approach for scaling by user and group.

Coordinated Permission View Our solution uses a coordinated view of hierarchical permission information. As the analyst moves the cursor over a cell in the treemap, a panel beneath is populated with the permission attribute records for the target cell and its parents. This method invites easy comparison of directory permissions between different hierarchical levels and allows the analyst to view the data in a standard format.

Effective Permission Scaling Effective permission scaling implemented with a single group of three sliders. The user and group for evaluation of the effective permissions are set through a pair of adjacent menus. These are initially configured with user and group unspecified. This combination calculates the world effective permissions. Moving a slider with automatically transition the treemap to cell scaling using local permissions.

Color Coding Treemap cell coloring can be performed in one of four modes. The default is a uniform coloring of all cells. User coloring will color cells based on owning user. Similarly group coloring will color cells by owning group. This is shown at the top of figure . Finally, cells can be colored by local file permission. This is helpful for visualizing whether a group of files have similar or dissimilar permissions set. We utilize nominal color scales created in ColorBrewer2[2] for our treemap.

Exploration Existing methods such as using regular expressions to filter the output of Unix shell commands such as *ls* and *find* are declarative, requiring the user to explicitly indicate what they would like returned. This is useful if the user knows what they they are looking for and is familiar with the specific syntax to extract it, but cumbersome when the user is attempting an open-ended exploration.

Our approach to exploration is distinguished by the use of animated transitions between display states. An example of this type of transition is shown in figure 2. This allows the user to maintain global context, while utilizing animation to draw attention to differences in the data between visualization states. This approach

[2]http://colorbrewer2.org

Figure 2: Visualization of Files with World Read Permission. Top: Initial uniform sizing. Middle and Bottom: Increasing size and color emphasis of world readable files.

is in keeping with the guidelines Misue suggests for preserving an analyst's mental map between states [3]. Each transition is immediately reversible allowing users to explore a projection of the data and then return to the previous view if they so desire.

References

[1] M. H. Bruls and K. van Wijk. Jj, squarified treemaps. In *Proceeding of Joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42, 2000.

[2] A. Heitzmann, B. Palazzi, C. Papamanthou, and R. Tamassia. Effective visualization of file system access-control. In *Visualization for Computer Security*, pages 18–25. Springer, 2008.

[3] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183–210, 1995.

[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*, pages 1473–1482. ACM, 2008.

[5] B. Schneier. *Secrets and lies: digital security in a networked world*. John Wiley & Sons, 2011.