Ensuring that securing of files used by a web domain requires a thorough understanding of file permissions in Unix-like systems. Unix file permissions determine who can read, write, and execute the files. Moreover, understanding regular expressions is crucial for various tasks, such as searching for specific patterns in files or filtering input data.

File permissions are critical in maintaining the security and privacy of data in a Unix-based system. In Unix-like systems, each file and directory are assigned a set of permissions, that control the access rights for the owner, the group, and others. The set of permissions are a string consisting of 10 characters, with the first character indicating the category, whether it’s a directory or regular file. The rest of the remaining characters are separated into three groups that consist of three bytes, that represent the owner, the group, and other associated permissions, respective to that profile.

Examining the five file listings provided, we can understand their associated permissions, attributes, and chmod commands to set these permissions. For example, the install.sh file in question possesses read, write, and execute permissions for the owner, the group can read and execute while read only permissions are set for the others. Using the chmod command to set these permissions, "chmod 754 install.sh".

Looking at the User\_Guide.txt file we can see that this is a regular file with read and write permissions for the owner and read permissions for the rest. The file's owner and group are both set to 'staff'. To establish permissions that allow only the owner to modify the file, while permitting group members and others to view its contents, utilize the chmod command: "chmod 644 User\_Guide.txt".

Evaluation of the Admin\_Guide.txt file is shown to be a regular file as well. Establishing read and write access to the owner, giving read-only access for the group, while allowing no access permissions for the others. This configuration ensures that only the owner can refine the file, and only members of the group can view it. The file is inaccessible to others. Both the owner and group for the file are assigned as 'staff'. To configure the appropriate permissions, execute the chmod command: "chmod 640 Admin\_Guide.txt".

The Documents directory is shown to have the read, write, and execute permissions assigned to the owner, while the group and others have only read and execute permissions. With regards to a directory, the execute permission enables users to access its contents. Here, the owner can create, modify, or remove files, while both the group members and others can only read the files in the directory. With both the owner and group designated as 'staff', apply the necessary permissions using the chmod command: "chmod 755 Documents".

Taking a look at the httpd file, we can see that the file that has all the permissions, read, write, and execute assigned to it. Additionally, the file has been set with the (setuid) byte for the owner of the file and (setgid) byte for the group, along with read and execute permissions. These bytes enable the file to be executed with the effective user and group ID, instead of a user just running the file. In this scenario, both the owner and group are designated as 'nobody'. This setup is frequently employed for processes that necessitate elevated privileges. To establish these permissions, apply the chmod command: "chmod 6755 httpd".

Regular expressions serve as patterns for identifying character combinations within strings. They play a crucial role in various tasks like searching, editing, and validating data. Regular expressions include metacharacters such as the carrot (^) to indicate the beginning of a line on the terminal, the dollar sign ($) to indicate the end of a line, an asterisk (\*) to match any instances of any character, and lastly, the question mark to match exactly one instance of any character. Examples of regular expressions in different situations include: using the expression ".\*txt$" will match all files that end with "txt"; while employing the expression ".a.\*sh$" that will match any file with that has a second character of "a" and also ending in "sh"; and utilizing regular expressions to identify files that begin with a numerical character, contain blank space, or correspond to a standard U.S. phone number in a specific format.

Das, S. (2013). Your UNIX/LINUX: The ultimate guide (3rd ed.). McGraw-Hill Education.

Friedl, J. E. F. (2006). Mastering Regular Expressions, 3rd Edition. O'Reilly Media.

"Regular expression" (2021). In Wikipedia. Retrieved April 21, 2023, from https://en.wikipedia.org/wiki/Regular\_expression

Nemeth, E., Snyder, G., Hein, T., Whaley, B., & Mackin, D. (2017). Unix and Linux System Administration Handbook, 5th Edition. Addison-Wesley Professional.

"chmod" (2021). In Linux man pages. Retrieved April 21, 2023, from https://linux.die.net/man/1/chmod