Assignment 1 - Discretionary Access Controls

I have provided two implementations for performing the fput, fget, create_dir, cd, setacl and getacl command - without and with using sudo, which are described as follows -

ACLs and Setuid

- The ACL for a given file is stored in file .<filename>.acl in the same directory in which the <filename> exists. The ACL for a given directory is stored in file .<directory>.acl in the parent directory of <directory>. The access to any requested file or directory is mediated through ACL to check for permissions to perform the requested operation. Note that the ACL for the file or directory is stored in a separate file with .acl postfix.
- fput, fget, create_dir, cd, setacl and getacl are setuid programs. If the running user is granted permission after checking ACL we seteuid() appropriately and relinquish privileges using seteuid() after given operation is performed.
- Currently ACL is based on user, handling for group and other permission is not presently supported.
- Our handling defends against following attacks/bugs/errors
    - Appropriate permission checking is done using ACL, and unless permission is granted subsequent operations are aborted.
    - All these commands have appropriate error handling for unrecognized options.
    - seteuid() is appropriately called to grant and relinquish privileges during the execution of these commands

Commands Usage:

(1) fget FILENAME
- fget of file is read only after file creation.
- fget prints the file contents on the terminal.
- To perform the fget command the ACL for requested file is checked for read permission for running user in .<filename>.acl

(2) fput FILENAME <string>
- File is created only using fput the very first time. At file creation ACL entries for file is created using DAC of newly created file. For subsequent fput, permission is managed only using ACL.
- To perform the fput command the ACL for requested file is checked for write permission for running user in .<filename>.acl

(3) create_dir [-p|--parents] [-v|--verbose] DIRECTORY ...

-p, --parents

no error if existing, make parent directories as needed

-v, --verbose

print a message for each created directory

- Directory is created only using create_dir the very first time, nothing is done if directory already exists. At directory creation ACL entries for directory is created using DAC of newly created directory.
- To perform the create_dir command the ACL for parent directories of given directory is recursively checked for execute permission for running user and the immediate parent directory is also checked for write permission for running user.

(4) cd DIRECTORY

- To perform the cd command the ACL for directory to which we need to change to is checked for execute permission for running user in .<directory>.acl. Also, execute permission is checked in parent directories of given directory recursively for the running user.

(5) getacl [FILENAME | DIRECTORY]

- To perform getacl command the requested file's ACL is checked for read permission for running user.
- This command dumps the content of .acl file present for the specified file or directory on terminal if running user has permission to view it

(6) setacl [-m|--modify|-x|--remove] acl_spec [FILENAME | DIRECTORY | BINARY FILENAME] ...

acl_spec can be given as u:<username>:[r|-][w|-][x|-]

-m, --modify

Modify the ACL entry of the specified file or directory by adding new ACL

entry or modifying existing ACL entry with the new ACL entry.

-x, --remove

Remove the ACL entry from

ACL of the specified file or directory.

- To perform setacl command the requested file's ACL is checked for write permission for running user.
- A user which does not exist in the Linux system cannot be added as ACL entry using setacl.

Simple sudo

Usage:

sudo followed by respective fput, fget, create_dir, cd, setacl, getacl command, for e.g.

- sudo fget FILENAME
- sudo fput FILENAME <string>
- sudo create_dir [-p|--parents] [-v|--verbose] DIRECTORY ...
- sudo cd DIRECTORY
- sudo getacl [FILENAME | DIRECTORY]
- sudo setacl [-m|--modify|-x|--remove] acl_spec [FILENAME | DIRECTORY | BINARY FILENAME] ...

- To perform fput, fget, create_dir, cd, setacl or getacl command the ACL for respective binary is checked for execute permission for running user.
- The owner for fput, fget, create_dir, cd, setacl and getacl binaries is 'fakeroot'.
- The ACL for fput, fget, create_dir, cd, setacl and getacl have already been created with execute permission given to 'fakeroot'.
- User ‘fakeroot’ can give execute permission to other users for these binaries using sudo setacl, for e.g.
    - sudo setacl -m u:<usename>:--x <path to binary file>
- Apart from the ACL checking difference in sudo mode to determine the permissions for the running user, the rest of behavior for these commands is identical to what is mentioned above for these commands.
- Currently, we assume that these binaries are in /fakeroot folder while searching for them during execution of ‘sudo’ command.