# PC649: MSc (IT) Summer Internship

Name -Harshil Patel

Roll No - 202012055

Mentor - Lavneet Singh

### 

### Snake-on-a-Tree Algorithm

Source code: <https://github.com/202012055/summer_internship/tree/master/snake-on-a-tree>

|  |  |
| --- | --- |
| **Whats is it ????**   * It is a novel algorithm for making only some files available for accessing. * It works by manipulating the file permissions on any unix-like OS. * To make the process efficient we only change the minimum number of file permissions. * For Example: if no file in the sub-tree of a dir is public then we can just remove the executable permission on the dir and then no process will be able to climb down that dir. * The end result looks like a snake(a series of revoked permissions) on a tree(file-heirerchy) so i named it snake-on-a-tree. | **In-Action**   1. Initial Tree: every thing is private      1. after making dir1/subdir1/file1 public      1. after making dir1/subdir1 public      1. after making dir1 private |
| **Implementation**   * The algorithm is implemented as a bash script library. * It is needed to be sourced by the user script. * It exports 3 functions:  1. setROOT   Sets the root of the tree on which other functions act.  Takes 1 arg, a path to dir.   1. makePublic   Makes that dir/file public.  Takes 1 arg, a path relative to ROOT.   1. makePrivate   Makes that dir/file private.  Takes 1 arg, a path relative to ROOT. |
| **Limitations**   * This algorithm only works on static file-heirerchy. * If the tree’s structure is changed then the resulting structure might not be secure and most likely will not be understood by the later runs of the algorithm. * To circumvent this problem reapply all the permissions and make the tree consistent again, but it is a very expensive opperation. |