1. characters). user can enter a valid

Psychology, Behaviorism

1. Possiblecopying of arbitrary data, command injection OR segfault because the passwordinput is not null terminatedLine: 23, 27Description: The password used as an input is notnull terminated, an attacker can copy arbitrary data in this field. This couldalso lead to a segmentation fault. This should probably use std:: stringrather than character arrays (including returning one). If we still usecharacter arrays the pointers should be marked as const to enforce notmodifying them within the function and allow compiler optimizations. Passwordneeds to be properly quoted and escaped to stop command injection (and allowpasswords with spaces and special characters). 2.

Possiblecopying of arbitrary data, command injection OR segfault because the path inputis not null terminatedLine: 25, 27Description: The path used as an input is not nullterminated, an attacker can copy arbitrary data in this field. This could alsolead to a segmentation fault. This should probably use std:: stringrather than character arrays (including returning one). If we still usecharacter arrays the pointers should be marked as const to enforce notmodifying them within the function and allow compiler optimizations. Path needsto be properly quoted and escaped to stop command injection (and allowpasswords with spaces and special characters). User can enter a valid pathfollowed by “; rmdir C: Windows /s”? OR any other command. 3. Possiblecopying arbitrary path Line: 68Description: The cin.

getline() assumes theminimum size of path to be n = 1024. But 1024 is the intended maximum size ofpath. It is terminated by a null character and therefore the size should be nomore than 1024-1 = 1023.

If we run into the end of the file or enter path morethan 1024, the function will still return True. Due to this an attacker canprovide arbitrary path. 4. Possibleto execute executeMount() function with incorrect password and path Line: 74, 84Description: It is already established that anattacker can provide arbitrary input (password & path) and due to thesystem call functions for executeMount() it is possible execute with lessprivileges and arbitrary password and path which can unmount or replace thedisk. 5. Incorrect use of return function Line: 27, 48 Description: std:: string should be really used her, for the return value as well.

Thendereference . c str to get a const char * pointer. Since it is not intended tojust return the std:: string. c_str from the function however, std:: stringwill got out of scope and an undefinedbehavior can be invoked.

6. Nopermission required to call mountvolLine: 21, 24Description: Mountvol can be called through anadministrator account password however separate privileges must be set to callmountvol directly. Other threats The following threats have either no direct impact on the securityor the information provided is not sufficient to determine the level of thethreat: 1. Strlen, Strcat, StrcpyLine: 21, 35, 37, 23, 24, 25, 40, 42, 44, 46, 22, 39Description: For safer practice, strlenshould be replaced with strnlen; Strcat should be replaced with strncat; strcpyshould be replaced with strncpy or strlcpy. 2. Constantsdeclared with #defineLine: 10, 11Description: constant should be defined withconst unsigned mechanism rather than using #define.

Because once a constant isdefined it can never be changed or undefined. 3. ANY user cansee DISK_INFOLine: 63Description: It is possible to call thesystem() command at line 63 and obtain DISK_INFO without any privileges.