Software Security Strategies for PC-Based Education Laboratories: A Case Study

Eduardo Boemo  Fernando Barbero  Juan Meneses
Universidad Politecnica de Madrid

The election of Personal Computers in education laboratories is a worldwide tendency. However, DOS does not have a complete set of commands in order to maintain the integrity of the software. Thus, errors, viruses or vandalic acts can interrupt training and produce a serious overhead if the software needs to be reinstalled. In order to contribute towards eliminating the disadvantages described above, a prototype program called Kipitwel was developed at the School of Telecommunication of Madrid, Spain. When installing, Kipitwel performs two preliminary actions:

- Creates a file called permits.dat, containing a list of permitted commands with its corresponding path. The administrator must edit this inventory and disable every conflicting command or program. After that, permits.dat is automatically hidden and encrypted.

- Organizes the hard disk in two zones: the user area, a subdirectory confined to a virtual drive created using the DOS command subst, and the system area in the rest of the hard disk, where all application program and configuration files, are included.

When running, the core of the security scheme is supported by two resident programs: checker and checkop. The first routine intercepts the DOS function of loading a program from disk to memory and executing it. After that, it verifies whether or not the program's name and path are in permits.dat. If true, checker releases the control of execution to DOS function exec. If not, it returns an error message. Checkop, the other resident routine intercepts the calling of copy, rename and delete, and them verifies if these commands only affect files situated in the user area. If not, it does not execute the command and returns an error message. Checker and checkop definitely guarantee the integrity of application programs.

- Users cannot execute format, attrib, etc. because these commands are not included in permits.dat.

- Users cannot access permits.dat due to the fact that the file is hidden and encrypted. Furthermore, there is no maintaining tool authorized in permits.dat.

- Users could hypothetically edit and compile a virus program, but would be impossible to run it (the virus name is not in permits.dat).

- Users cannot introduce a "Trojan Horse" program by renaming a virus (or other program) with the name of a permitted command. Checkop does not allow renaming or copying in the system area of the hard disk, and checker verifies if the complete path of the program agrees with the specified path in permits.dat.

In order to separate user files, each student has a login password that permits entrance into his or her own virtual drive. After booting, a program called login.exe, presents a login screen and checks the validity of the password. Following a successful entrance, the program changes to the subdirectory assigned to that particular user. The form of such a directory is c:\people\group xx, where xx is a number between 1 and 30. The blank space between the word "group" and the group number does not allow changing directories. For example, a sentence like cd c:\people\group 4 will not work in DOS. This barrier also prohibits executing commands like copy, dir, type, delete, etc. from other subdirectories. Thus, the confidentiality of the student's work is guaranteed.

In Kipitwel, users are assigned one of two different privileges: normal and superuser. Normal users can run total or partial lists of programs. They can create, copy, modify or delete their own files. They cannot load new programs or change configuration files. The second level of availability corresponds to superuser or system administrator. His or her main activities are deleting or installing programs, and authorizing or forbidding the entrance of users. The superuser has access to all the resources of the PC without any restriction.

The protection scheme described above could be worthless if the execution of kipitwel.exe (invoked by the autoexec.bat) where interrupted. In order to complete the security scheme it is necessary to control the booting process. In the first line of the config.sys file there is a driver called keymask.sys that temporary inhibits keyboard utilization. It makes it fruitless to try control-c or similar key combinations in order to suspend the booting process. As a complement to the keymask.sys driver, it is also indispensable to forbid booting from drive a: Some PCs allow it by changing and protecting the setup. When this option is not available, the simplest trick is to connect diskette drive a: to the connector corresponding to b: This action eliminates drive a: Consequently, no booting from this drive can occur.

The program has been successfully used since 1990 on 20 286s and 386s PCs in a 450-qualified-students laboratory. A complete technical report can be found in [1].

References