Pbs 17913

Technology, Computer

PROCESS BASED SECURITY, WHAT IS IT AND HOW IT WILL CHANGE YOUR COMPUTER'S LIFE

This note is for the purpose of explaining a new way to implement operating system security called process-based security. First, an explanation of traditional operating system security is given. Next, an explanation of process-based security is given. Finally, there are presented three examples of failure on the part of traditional security with how process-based security would not have allowed such a problem.

Before talking about process-based security, it is important to go over current state of the art in operating system security. As operating systems are implemented today, security is designed to give access to resources based on a user who has been identified. When it comes time to make a resource available to a running program, the operating system checks to see if the user, who is associated with the program, has rights to the resource. No consideration is given as to what the program itself might need with regard to the requested resource. This current view of security has the disadvantage of allowing programs access to resources outside of their intended use, i. e. using a hex editor to work with accounting files.

In a process-based security system, access to resources is based on the process running. When the administrator of a system loads a new program onto that system, he/she determines the needs of the program and sets up the security profile for it. As the program runs on the system and requests access to resources, its security profile is checked, to determine whether the resource can be made available. The operating system is not concerned with

who initiated the requesting program, only what the program wants. User interface programs can be installed to control what programs are available to what users.

In the early days of UNIX, one quick way to determine user passwords was to encrypt all the words in an English dictionary, get a copy of the password file, and compare the two. The password file only needs to be accessible to the login program and user maintenance program. With UNIX’s security based on users or groups of users, restricting access to just those programs is not possible. Thus people had read-only access to the password file and were clever enough to figure out the relationship of passwords to common English words. In process-based security, no access could have been made to the file, because it is very easy to restrict only the login or user maintenance programs to see/use the file.

A much more pressing need for process-based security has surfaced because of internet browsers and has become elevated with the addition of Java. An internet browser without Java can be used to obtain information about the user’s computer outside of their control. An internet browser with Java can be used to destroy any data the user normally accesses. With process-based security, the internet browser could be given a profile allowing limited access to a “ just-in-case” directory only, not allowing any other access that could be potentially harmful. Most people think the Java access security built into browsers can keep this from happening. However, Java code can be written to test what type of machine is currently being used, then call machine specific code that uses the operating system directly, bypassing the

browser’s safeguards. There are over fifty-million IBM-style personal computers out there running Windows with Java enabled browsers that virus writers would love to tackle.

Today computer viruses seem to be the norm. A malicious program attaches itself to a friendly program, causing the friendly program to terrorize the system. In two ways this is stopped with process-based security. First, the virus, being unknown to the system, would not have a security profile allowing it to access any files or I/O ports, thus keeping it from running or infecting in the first place. Secondly, if the friendly program was infected, before being loaded by the administrator, the extent of damage is minimized. No program is going to be given free rein of the system, and access rights are not increased when the program is run by the administrator.

Operating system security has always been looked at from the perspective of what the user should be able to access. Unfortunately that allows files and devices to be accessed by programs other than those programs originally intended to manipulate those files and devices. When operating system security is looked at from the angle of what a program should be using, not only can access be given on an “ as needed basis” only, but security installed in programs themselves, cannot be defeated by simply accessing the data files outside of the intended program.

In conclusion, it is important to remember “ people don’t delete files, programs do.” Restricting a program’s access to what it needs is the best way to keep out unwanted deletions or additions. Process-based security

gives you a much better handle on how your computer/network is used and makes it a much more secure environment against malicious or naive attacks thus making your life and your computer’s life less complicated.

Word Count: 850