Using someone's logged-on workstation is a favorite method used by penetration testers or criminals who have gained physical access to devices connected to a network. Such people can wear appropriate clothing and assume a casual, relaxed air to convince passers-by that they are authorized to use someone else's workstation. Sometimes they pose as technicians and display toolkits while they are busily stealing information or inserting back doors into a target system.

Unattended workstations that are logged on are the principle portal for logical piggybacking. Even a workstation that is not logged on can be a vulnerability, since uncontrolled access to the operating system may allow an intruder to install keystroke-capture software that will log user IDs and passwords for later use.

A simple but non-automatic method is to lock the keyboard by physical removal of a key when one leaves one's desk. Because this method requires a positive action by the user, it is not likely to be fool-proof -- not because people are fools, but because we are not machines and so sometimes we forget things. In addition, any behavior that has no reinforcement tends to be extinguished; in the absence of dramatic security incidents, the perceived value of security measures inevitably falls.

There are two software solutions currently in use to prevent unauthorized use of a logged-on workstation or PC when the rightful session-owner is away:

- Automatic logoff after a period of inactivity
- Branch to a security screen after a timeout

One approach to preventing access at unattended logged-on workstations is at the operating system level. The operating system or a background logoff program can monitor activity and abort a session that is inactive. These programs usually allow different groups to have different definitions of "inactive" to adapt to different usage patterns. For example, users in the accounting group might be assigned a 10-minute limit on inactivity whereas users in the engineering group might get 30 minutes.

When using such utilities, it is critically important to measure the right things when defining inactivity. For instance, if a monitor program were to use only elapsed time, it could abort someone in the middle of a long transaction that requires no user intervention. On the other hand, if the monitor were to use only CPU activity, it might abort a process which was impeded by a database lock through no fault of its own.

Currently, PCs can be protected with the timeout features of widely-available and inexpensive screen-saver programs. They allow users to set a count-down timer that starts after keyboard-input; the screen saver then requests a password before wiping out the images of flying toasters, swans and whatnot. The critical question to ask before relying on such screen savers is whether they can be bypassed; for example, early versions of several Windows 3.11 and Windows 95 screensavers failed to block access to the CTL-ALT-DEL key combination and therefore allowed intruders to access the Task Manager window where the screensaver process could easily be aborted. Today's screensavers are largely free of this defect.
A few suggestions for secure screen savers, timeout and shutdown utilities (these references are not endorsements):

- Check your operating system and important application programs for existing logoff timeouts and enable them with appropriate parameters;
- See NetOFF, which works with Novell Netware and Windows NT -- from Citadel Technology <http://www.citadel.com/products/netoff.html> and its distributors;
- WinExit, part of the NT Resource Kit from Microsoft, is a secure screen-saver that causes an automatic session logoff after a timeout on Windows NT systems (see <http://www.win2000mag.com/Articles/Index.cfm?ArticleID=4541> for details);
- ShutdownPlus family of products from WM Software <http://www.wmsoftware.com/shutdownplus/index.htm> which work with Windows 9X, NT and 2K operating systems and Citrix Metaframe include features for forcing a shutdown and reboot on a specified schedule and running particular applications before and after the shutdown.

* * *

M. E. Kabay, PhD, CISSP is Associate Professor in the Department of Computer Information Systems at Norwich University in Northfield, VT. Mich can be reached by e-mail at <mkabay@compuserve.com>. He invites inquiries about his information security and operations management courses and consulting services. Visit his Web site at <http://www.mekabay.com/index.htm> for papers and course materials on information technology, security and management.

Copyright © 2001 M. E. Kabay. All rights reserved.

Permission is hereby granted to *Network World* to distribute this article at will, to post it without limit on any Web site, and to republish it in any way they see fit.