Security considerations in a multi-programmed computer system

by BERNARD PETERS
National Security Agency
Fort Meade, Maryland

INTRODUCTION

Security can not be attained in the absolute sense. Every security system seeks to attain a probability of loss which is commensurate with the value returned by the operation being secured. For each activity which exposes private, valuable, or classified information to possible loss, it is necessary that reasonable steps be taken to reduce the probability of loss. Further, any loss which might occur must be detected. There are several minimum requirements to establish an adequate security level for the software of a large multi-programmed system with remote terminals.

The management must be aware of the need for security safeguards. It must fully understand and be willing to support the cost of obtaining this security protection. It must be technically competent to judge whether or not a desired security level has been reached. It is important that the management understand that no software system can approach a zero risk of loss. It is necessary to reach an acceptable degree of risk.

As any professional in the field of computers knows, whether or not management is sufficiently aware of and willing to pay the cost of a given software feature is not easy to determine. The costs of security are not exorbitant provided security is needed. However, they are not trivial or negligible. It is very important that the management have an understanding of how the security is gained and what administrative steps must be taken to maintain and protect the security level which has been gained. This paper will not fully explore certain problems which affect security as, for example, physical security which is protection of a space against penetration, or communications security or personnel security.

To obtain the security which software can contribute, the following principles must be followed:

1. The computer must operate under a monitor approved by appropriate authority. This can provide the authority for the expenditures which security may require. The question of appropriate authority is one not easily answered. For the individual business concern it obviously is corporate management. For contractors to the Armed Services it is a security officer of the service. Who can tell who is in charge in a University?

The computer must operate under a monitor because the monitor acts as the overall guard to the system. It provides protection against the operators and the users at the remote terminals. The monitor has a set of rules by which it judges all requested actions. It obeys only those requests for action which conform to the security principles necessary for the particular operation.

The discussion of what makes up an appropriate monitor is rather involved. Therefore, a more complete discussion of the security aspects of a monitor will be given after a summary of the overall security principles.

2. The computer must have adequate memory protect and privileged instructions. These are needed to limit user programs which must be considered to be hostile. Memory protect must be sufficient so that any reference, read or write, outside of the area assigned to a given user program must be detected and stopped. There are several forms of memory protect on the market which guard against out-of-bounds write, thus protecting program integrity, but they do not guard against illegal read. Read protect is as important as write protect, from a security standpoint, if classified material is involved.

The privileged instruction set must contain not only all Input/Output (I/O) commands but also every
command which could change a memory bound or protection barrier. To have less would be to reduce the whole system to a mockery. There can be no exception to the restriction that the computer can not be mounted in a subway car, subject to public tampering. The computer must be in a security protection to prevent local override of the whole system to a mockery. There can be no exception to the restriction that only the monitor can operate these privileged instructions.

3. The computer must have appropriate physical security protection to prevent local override of the monitor. It includes the obvious requirement that the computer can not be mounted in a subway car, subject to public tampering. The computer must be in an appropriately secure environment and not subject to intrusion by random individuals. But this is not sufficient. Certain key switches, such as those which affect the continued running of the equipment and the maintenance panel, must have simple physical barriers to prevent undetected intrusion from occurring. The senior operator provides protection against local override, but he cannot be expected to watch every switch continuously. Simple key-locks provide protection to supplement his attention span.

4. Electrical separation of peripheral devices is not necessary, provided the monitor has been approved by an appropriate authority. This reduces the probability of failure by eliminating troublesome separation hardware. It has been proposed by people who are first inquiring into security, that the peripheral devices be separated from the computer. A properly designed monitor for a multi-programming system will provide sufficient logical separation of peripheral devices to make electrical separation unnecessary. If the monitor is insecure and beyond securing, the system should not be used with the expectation that security will be provided.

5. The computer may operate in a multi-programmed or a multi-processor mode provided the monitor has been approved for use under such modes. In fact, to provide proper security it seems that the computer must be designed for the multi-programmed mode. In the rare case, when only one program is running, it is multi-programming with a single program. Multi-processors, for the purposes of this paper, are assumed to be a variant of multi-programming. It is obvious that this isn’t true; however, it is felt that simple, logical extensions will carry the multi-programming philosophy into the multi-processor mode and still provide adequate protection.

6. Operating personnel must be cleared to appropriate levels. Trustworthy personnel must be available to enforce the rules against intrusion at the computer. This does not mean that every operator in the installation need be cleared, but it does mean that on every shift there must be at least one individual who is fully aware of the requirements of the operating doctrine which is needed to protect the secure data. He must completely understand what it takes to utilize the system appropriately and securely in any circumstance. All other operators must understand that there exists a protection philosophy. They must be subject to some form of discipline, such as termination or suspension for failure to obey regulations. On the other hand, the operating doctrine must be made clear and precise so that it can be obeyed. Specification of the operating doctrine is a nontrivial problem and can be very difficult depending upon the complexity of the tasks assigned to the operator. In the interest of keeping the operating doctrine under control it is proposed that the operator be designed out of the operation as much as possible.

7. A log of all significant events should be maintained both by the computer and the operating personnel. The form and contents of these logs should be approved by appropriate authority. The log is the ultimate defense against penetration. The question of whether or not some data have gone astray must be answered by the log. The log ascertains who does what, and to what extent he has attempted to do something illegal. It also records the utilization of files. It indicates whether or not some security restriction is significantly interfering with the production of the system. The log will provide the review and feed-back which will make it possible to relax security where it is overly stringent and it will provide the factual basis for increasing security where it has failed to prove a sufficient level of confidence.

8. Every user should be subject to common discipline and authority. He shall know and understand the conventions which are required of him to support the security system; it is particularly important that everyone who has access to the more highly classified levels of material fully understand his part in the protection of the data and the system. The remote terminals must also be electrically identified by the station by its permanent connection to a specific set of hubs in the machine. Station identification can not be dependent upon the action of the user unless this action has been carefully considered for its security connotation.

9. It is possible that design considerations for the system require that an individual remote terminal change its particular security level upward at one time or another. It is necessary that this change be accomplished in a secure fashion. An acceptable method would be to issue the user of such a remote terminal a one-time list of five letter groups. The
Security Considerations In A Multi-Programmed Computer System

user would be required to use each group, in turn, once and only once. This would make observing of the particular procedure for raising the station security level valueless to an observer. It has the further effect that should an observer obtain and use the next available five letter group, such use would cause a conflict with the legitimate user's call. Thus there would be detection of illegal use.

Let us now return to a discussion of the attributes of an acceptable monitor. The monitor is the key defense, the key security element in the system. It must be properly designed.

The cost of a properly designed monitor is probably not more than 10% greater than that of a monitor which is minimally acceptable for multi-programming. Monitors for multi-programming need a high degree of program and file integrity to be effective. Improvement of such a monitor to the point where it is acceptable for handling material in a secure fashion, is an achievable goal.

The monitor must perform all input/output (I/O) without exception. No user program can be permitted to utilize any I/O device, except via a call to the monitor. It is very important that the monitor control all I/O which could provide large amounts of information to a penetrating program. The monitor must manage the system clocks and the main console. If the maintenance panel is beyond management by the monitor, it must be secured.

The monitor must be carefully designed to limit the amount of critical coding. When an interrupt occurs, control is transferred to the monitor and the core is unprotected. The monitor must, as soon as reasonable, adjust the memory bounds to provide limits on even the monitor's own coding. This requires that the coding which receives interrupts be certified as error free for all possible inputs and timing. The greater portion of the coding of the monitor need not be certified error free to the same degree because it is operated under the same restraints as a user program. By keeping the amount of coding that can reference any part of core without restriction to a few well-tested units, confidence in the monitor can be established.

The monitor must be adequately tested. It is certainly necessary to adequately demonstrate the security capability to the governing authority. In addition to passing its initial acceptance the monitor must also test itself continuously. For instance, the memory bounds protection can be expected to fail with some probability. Every user program which conforms with the security safeguards will be expected to run without violating the memory bounds protection and therefore will not test such a feature. It is necessary, therefore, that some special program, or some part of the monitor deliberately and periodically violate the memory bounds protection to verify that the bounds protection checker is, in fact, working.

It is extremely important that tests for all significant safeguards built into a monitor be periodically and deliberately verified. The confidence which management has in the security safeguards will, in part, rest on the evidence furnished by such self-tests.

The monitor must keep the user programs bounded by memory protect while they are operating. The individual user program must be free to reference its assigned area of core, but nowhere else. All I/O action, and any out-of-area reference by user programs, must be via calls to the monitor. This will protect information concerning the security levels and authorized users, files, and outputs from access by the operating elements of the monitor, such as loaders and terminators to be treated as though they were individual user programs, thus reducing the amount of coding that is outside the safeguards.

Authority to reference random access peripherals must be established by the monitor and all references must be checked for validity and authority. It is a small cost in a random access I/O routine to determine if a supplied address is legal and within bounds.

What is the monitor to do if there is a violation? If there is a violation of the memory bounds or the use of a privileged instruction by a user program, the monitor must immediately suspend the offending program and must make log entries. It must also prohibit the further use of the offending program by the user submitting the violating program until specifically authorized by a supervisor.

The suspension of violating program requests must be thorough and complete. If the task has been divided into multiple concurrent operating activities, all such activities must be terminated. If the task has resulted in a chain of requests, all such requests must be removed from the queue. Violation of security rules by any activity must result in a complete abort of all parts of that request. This is necessary to prevent a user program from making multiple tries against the security system.

Security rules cannot be suspended for debugging or program testing. It is clear that security rules cannot be suspended for debugging programs because the new program is the one most likely to violate security. The debugging system must live with the security restrictions although some concessions can be made for debugging. For example, one can flag a program that is in a debugging state. Then if a bounds violation occurs, the system could merely log it and send a dump of the program to the user rather than sounding a major alarm.
Tests of security flags must be fail safe. If a flag, when it referenced, is inconsistent or ambiguous it must fail to qualify. Thus security flags must be adequate. As an example, one could use a single bit, but then an error could change this bit to indicate a different but still valid combination; CLASSIFIED could too easily become UNCLASSIFIED. Therefore, single bits are not acceptable and the question then becomes "How many bits should one utilize?"

The specific configuration of bits is dependent upon the goals and requirements of the particular security system and the machine or bit handling capability of the individual machine. The author has used a 60 bit flag in a machine which has 30 bit words with half word capability. This led to four 15 bit configurations which were complementary to give verification. Forty bits would have been sufficient for the particular application, so the next higher multiple of word size was chosen.

Software security is derived by the proper application of a few sound principles which must be consistently applied in many small actions. With a proper understanding and careful review, software security can be maintained so long as physical and personnel security are assumed. The principles set forth in this paper have been generalized from the specific development of a specific system which dealt with multi-levels of classified information.