M.I.T. Laboratory for Computer Science

Request for Comments No. 258
June 12, 1984

Trip Report: IBM/MIT Workshop on Security

by Bob Baldwin, Deborah Estrin

In conjunction with research funding that IBM provides to the MIT Laboratory for Computer Science, IBM and MIT annually hold a workshop on a topic of mutual interest. At this year's workshop 14 people from IBM and 9 people from LCS discussed problems and solutions in the area of computer network security and software protection. The MIT participants were members of the theory group, primarily, whereas most of the IBM participants were from product divisions.

IBM and MIT attendees played different roles in the workshop. The IBM attendees presented problems and requirements whereas the MIT attendees presented tools for solving problems. But, the solutions, as described, were not matched to the problems posed. It became clear that the MIT representatives and those from IBM had different ground rules. Among the theorists from MIT the goal was to show that breaking some cryptographic protocol is equivalent to solving a difficult mathematical problem. Whereas, among the IBM designers, managers, and marketers, the objective was to get the best possible security at no extra cost (According to IBM, customers do not want to pay much, if anything, for security.).

Bill Murray, staff advisor on product security in the Information Systems and Communications group at IBM, opened the workshop with a description of the problems of extending single-computer security goals to multi-computer networks. Many of the problems dealt with the joining and partitioning of name and protection spaces (e.g., how are the access privileges specified for users who are not registered locally). The remaining problems concerned distributing the reference data that must be checked to authenticate users, while maintaining each manager's ability to control the people and resources in his or her domain. All of Murray's examples exhibited a tension between security goals and transparency requirements.

Jack Sanders, a Systems Network Architecture (SNA) architect at IBM, presented the only talk by an IBM person in which solutions to some of the above problems were described. He characterized the network environment in which their problems arise, and he described the SNA solution to how two hosts can establish an authenticated and private link between themselves. Basically, a challenge-response protocol is used to authenticate a session key which is used to insure the privacy of further communication. Other IBM participants pointed out that this scheme required that every pair of hosts that wish to communicate share a secret key. According to Petre Turcu (IBM) this sort of N-squared key management was acceptable for their current networks. It was quite surprising to the MIT people that IBM did not use some sort of authentication server.

Oded Goldreich of MIT pointed out a second problem with the IBM scheme. When host-A opens a connection to host-B, A challenges B to encrypt a randomly chosen value under their shared secret key. If B responds with the right answer then A has authenticated B. The bug is that B in the mean time can open a connection to A and challenge A to encrypt exactly the same random value that A had originally given B. Of course, A's response to B's challenge will be an acceptable response to A's challenge of B. Apparently, the people representing IBM's Cryptographic Competency Center (Carl Meyer and Mike Matyas) knew about this attack, but they had not been consulted when the design decisions were being made.

Petre Torcu of the Architecture and Telecommunications group at IBM described security issues in the design of an application called SNA Distribution Services (SNADS). In SNADS, the access control list that belongs to a document is included within the document itself and authentication is left to lower layers. In general, the approach described by Torcu is to push all security concerns down to supporting levels of the architecture, as opposed to dealing with them in the application itself. Interestingly, the SNADS application runs on top of the system described by Sanders but does not use the security features because many existing products, with which SNADS must be compatible, do not implement the features. Murray raised a concern that information used in authentication and access control be provided by the recipient's host or application, instead of by the sender's.

The two talks presented by MIT professors, Silvio Micali and Shafi Goldwasser, covered the use of random function to solve the problems of authentication, digital signatures, sharing secrets, and flipping coins over a network. They have given the same talks at MIT, so we need not say more about their presentations. Although the methods presented did not provide immediate solutions per se to the problems presented earlier in the workshop, the discussion surfaced some important differences in the underlying assumptions used by the MIT and the IBM attendees. For example, the MIT researchers presented assumes that encryption is an inexpensive operation, whereas IBM system designers do not. On the other hand, most of the IBM people did not previously appreciate the importance of tying the breaking of a cryptographic protocol to solving a hard mathematical problem, nor did they appreciate the effort to carefully characterize the properties that a function needs to have to be suitable for use in certain cryptographic protocols. Several IBM people seemed content with the assumption that DES is resistant to all forms of attack, and thus no further characterization was necessary.

The closing talk was given by John Oseas on software protection. He gave an analysis of the kinds of software piracy that exist and the ground rules for acceptable solutions. He suggested that a range of solutions is needed in order to match the resources of various types of abusers.[1] In emphasizing the range of solutions that are needed, he pointed out that two programs each worth one million dollars in sales are quite different if a company expects to sell five copies of one for $200,000 each and two thousand copies of the other for $500 each. The constraints on software protection mechanism design are similar to security mechanism design, i.e., mechanisms must be convenient, if not transparent, to end users. The convenience in this case is often the ability to make a backup copy off of an unreliable storage medium or to copy the original onto the medium of choice. Given this concern, the hardest requirement to meet is insuring that no more than one copy of a program is in use at any given time.

[1] One unusual form of piracy described is due to salespeople who work on a commission bases. They do not actually sell pirated software; they make it part of a deal to sell the hardware (e.g., "When you come in tomorrow to buy the system, I'll give you $1000 of software to go with it.").