UCRL-75656
PREPRINT
Conf-740427--2

LAWRENCE LIVERMORE LABORATORY
University of California/Livermore, California

COMPUTERS AND CRIME

Robert P. Abbott

April 25, 1974

This paper was prepared for presentation at
The Eighth Annual Crime Countermeasures Conference,
University of Kentucky, College of Engineering, Lexington, Kentucky

COMPUTERS AND CRIME

Robert P. Abbott, Manager, RISOS Project*

Lawrence Livermore Laboratory, University of California

Livermore, California 94550

ABSTRACT

The involvement of computers in criminal acts can take the form of the outright theft of private data in storage, the changing or manipulating of stored data, or simply the use of the computer as a convenient tool or accessory in the commission of a crime. In the latter case the computer is used as a storage or output medium for criminally manipulated information that has escaped the data checking and auditing procedures that are part of the system. This paper describes types of computer-related crime that have been discovered. In particular, it focuses on subversion of a computer's operating system — i.e. that set of programs that control the allocation of a computer's resources for the execution of the various problems that the computer must solve. A typical case history of computer crime is looked at in detail. This paper also describes the activities of one group currently investigating the problem of operating system security. This group is the RISOS Project (for Research In Secured Operating Systems) at Lawrence Livermore Laboratory. The project, which is funded by ARPA, is developing a wide variety of computer-assisted "tools" and methods for the testing and evaluation of software security.

---

*This work is performed under the auspices of the U.S. Atomic Energy Commission. The RISOS Project is sponsored by the Advanced Research Projects Agency of the Department of Defense under ARPA Order No. 2166.

Introduction

Over the past two years the news media has printed many stories relating crimes and some type of computing activity. A few representative headlines were:

- This man stole $1 million from the phone company.
- Police under investigation for alleged sale of crime data in police data bank.
- Crooked operators use computers to embezzle money from companies.
- Computer theft by computer.

Is this a passing fancy or will we see more of it? Why is there a sudden interest in committing a computer related crime? If we are going to think about the subject of crime and computers, we may as well begin by examining those two classical aspects of a crime: Motive and Modus Operandi.

Motive

The question is: Why should a computer be involved in a crime? The answer must be that either the computer contains something of value or it can assist someone in obtaining something of value. Whether we recognize value or not it is, nonetheless, fast becoming a crime to obtain certain types of computer contained information. Computers, Privacy, Confidentiality, Security: these four words bring to mind a host of legislation which seeks to protect the individual citizen.

All forms of government and commerce make use of the computer. Privacy relates to the fact that information contained in these computers is frequently, or deliberately, descriptive of an individual person. That individual has a

right to his privacy. His privacy will be infringed if data about him is released without his due authorization and approval. Such a release is a violation of the confidentiality which he expects the computer and its supporting personnel to maintain. The computer and its supporting personnel must maintain policies and procedures which provide security for the contained information. In short, privacy relates to the individual, confidentiality relates to the treatment of data about the individual, and security relates to the policies and procedures which govern the computer's operation.

Within the government sector there are data banks which contain information on criminals, criminal activities, national security, military posture, taxes, health, education and welfare. In commerce there are computer records of inventory, credit and ownership. The intelligence and military communities have long had regulations for the protection of their data. It is only recently that the remaining portion of government and private industry have focused on legislation making it a crime to violate an individual's privacy through the unauthorized distribution of confidential data.

Thus, if we ignore the national security and military issues, there is some motivation to obtain confidential data. This information could be turned to profit either monetarily or politically. Blackmail and extortion have traditionally paid high rewards. Character assassination has been put to use by some individuals for political purposes. The release of income tax information about Ronald Reagan, Governor of California, is a case in point although computers were not used in that particular release.

It is my personal opinion that computerized records of ownership are likely to become a more common target than personal data. Money is not stored in a computer. The record of who owns it is stored in computer form. The same is true of property, both real and personal.

As an example, consider the possibility of modifying the record of ownership of a large sum of money. More specifically, suppose that the record is modified on a Friday to show that I own that money. Suppose further, that I invest that money in commercial paper. At this point the bank officials will examine my record to see if I own the money and the computer will verify that I do. Everyone believes the computer printout. On Monday I will liquidate the commercial paper, withdraw the profit, and change the computer record back to show the original ownership.

Although this example is rather sophisticated, the financial reward can be of such magnitude as to justify the effort and risk required to accomplish the deed. I believe that this type of crime will begin to occur simply because more and more records of ownership are being stored in computer form.

Modus Operandi

Just how might a computer be used in a criminal activity? One use can be as a tool. Perhaps the classic example of all time for this use can be seen in examining the Equity Funding case. Here, one computer was used to prepare data which was used to fool another computer for the purpose of committing an insurance fraud. The first computer prepared records of nonexistent insurance policies which were then included, by the corporate computer center, into the records of the company. The auditors could not detect the difference.

Equity Funding was not a computer crime. The crime was the manipulation of the value of Equity Funding stock by the appearance of a continued increase in insurance sales.

Another use of a computer in the commission of a crime is to make an assault on the computer itself. In the previous example of modifying the record of ownership of a sum of money, an assault on the computer would be required. Normally, the word assault would bring to mind a violation of physical security controls — some use of force to gain entry. There are effective countermeasures against this type of assault. Today's computer systems provide a new avenue for assault — telecommunication lines and the remote terminal.

There are physical security aspects to telecommunication lines just as there are physical security aspects to controlling access to the actual computer center. For the purpose of this paper, let us leave guards, guns, fences, wire tapping, etc. to those of you who are experts on those subjects. We shall restrict ourselves to considerations of software security and how they can be used in an assault on the computer. To put it another way, we shall examine software security as a Modus Operandi in committing a crime.

Security in the Computer Itself

The major responsibility for security inside the computer rests in the Operating System of the computer as compared to any of the Application Programs which run in the computer. The relationship between an Operating System and an Application Program is very similar in nature to the relationship of a flight controller and the pilot of an airplane. A pilot is in absolute control of his aircraft and its contents. An Application

Program is in absolute control of the process which it performs. The application process may be payroll, a management information system, or a calculation of the forces generated by a new rocket propellant. The pilot relies heavily on the advice and counsel of the flight controller to insure that his aircraft does not infringe on the airspace of another aircraft.

In a time-shared or multi-access computer system, The Operating System is in charge of all computer resources: memory space, disk, tape, card reader, printer, etc. If an Application Program requires memory space in which to run, the Operating System allocates the space. If the Application Program requires disk, or tape, the Operating System allocates disk or tape and actually performs the act of reading or writing on the device at the request of the Application Program. Since it controls all resources, allocates all resources and can access all resources, the Operating System must be of the highest security integrity. If it should be compromised, all data within the computer would be similarly compromised.

Methods of Attack

There are a number of methods by which an Operating System can be compromised. Each method can be invoked in many different ways. To illustrate the point, consider the following example which will work on many Operating Systems (OS):

> You write a program which requests that the OS provide you with read/write access to a file named JOE. In reality, you know that there is no file named JOE. The OS will not know that it does not exist until it has conducted a search of its file directory and

failed to match the name. In conducting the search, most OS will make use of some of your assigned memory space to conduct the search. That is to say, the OS will read a block of file directory into some of your memory space and search it for the name JOE. The OS will continue to do this until it reaches the end of the directory, at which time the OS will return control to your program along with some message to you that there is no file name JOE.

Most OS will not have zeroed out the memory space which it used. If this is the case, your program may search this space and print out for you whatever was in the last block of the directory. This may consist of the file names of other peoples' files, their owners' name, the access combination, etc. It takes little imagination to project what use could be made of this information.

In the software security business, this technique is called "Scavenging." It comes in many different sizes, shapes, and colors. A more difficult, but much more rewarding method of compromise would be to trick the OS into thinking that your program is actually a part of the OS. This is possible because of the relationship between the OS and an Application Program. The OS operates in a privileged or Master Mode where it will be afforded all privileges and accessibilities.

At this point in this paper we must recognize that the OS which are in use today were not designed with security in mind. There are very few countermeasures built into an OS. Each OS represents a large investment in time and money. It is impossible to rapidly make a new OS which is a secure entity with sufficient built in countermeasures. Some kind of retrofit process is necessary to make use of our present OS. The retrofit process must include an examination of the OS to reveal the location of security flaws so that

appropriate countermeasures can be invoked. If the examination is thorough, as it must be, it would be applicable as a verifying process in checking out OS which are advertised as being secure.

An examination or audit of the security features of an OS is a very recent occurrence. The first attempt probably took place in 1970. From 1970 until 1972 security audits were conducted by a small number of informally organized teams of extremely competent computer scientists. In fact the number of groups numbered less than five and contained on the order 2-3 persons each. The entire process was oriented towards demonstrating that there was indeed a security problem. As such, it was sufficient to locate one or two flaws and then make the observation that since this only took them a few days to find, it must be true that there are thousands more in any OS. From an academic point of view, this proved the case that there was a need for the complete redesign of all existing OS for the purpose of implanting security. Note these points:

- Even if securely designed OS were available, there would still be the need for an audit to verify the statement.
- The methods used by the early audit teams were random in nature.
- A systematic and thorough examination procedure must be devised.
- Existing OS must be used until they can be economically replaced.
- Flaws in existing OS must be located and repaired to the point that the OS are usable in a secure sense.
- The early teams rapidly lost interest in performing such examinations.

The RISOS Project

To address these issues, a group was formed to conduct Research In Secured Operating Systems (RISOS). RISOS consists of about 15 to 20 computer scientists, electronic engineers, and statisticians located at the Lawrence Livermore Laboratory. The Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense has provided $1.8 million to sustain the group over a three year period. The group's attention is focused on those time-shared and/or remote access systems which support the hardware of the seven largest computer manufacturers. The fact that there are similarities between the various OS allow RISOS to generalize a flaw found on a particular system to be applicable to other systems.

One of the first tasks undertaken by RISOS was to understand the nature of the much rumored student assaults on campus computers. The story was that university students could find all system flaws with almost no effort and in fact did it daily as an exercise in offing the system. Our findings revealed that this was far from the case. In fact, it appears that if 100 students were to be turned loose on a particular system, the chances are very strong that they would each find the same error rather than each one finding a different error. This fact serves to emphasize the need for a systematic methodology for the examination of an OS for security flaws. It is easy to identify the giraffes, but it is difficult to trap the mice.

The two largest efforts within RISOS are: 1) to identify the classes of security errors and 2) to make use of computers themselves in locating their errors. The previous example of "scavenging" is one class of error which can be present within an OS. At the present time there appears to be

less than twenty distinct classes of security flaws. The need to automate the search for errors can be seen by examining the size of today's OS. They contain anywhere from 2 to 6 million lines of instructions. Not even the designers of such systems are able to retain intimate knowledge of any sizable portion of these OS.

RISOS makes use of computers in both the analysis and the examination of an OS. During the analysis, the OS is treated as a data base. Several programs operate on this data base and point out to the skilled examiner certain areas which might contain logical or implementation errors which would influence software security. It is still a human chore to verify that the machine-located areas do or do not contain errors. A project of longer range is to use the computer to model the interactions of an OS in order to understand how time dependent sequences of activities may impact security.

During the evaluation of a specific OS, RISOS makes use of a mini-computer to assist in conducting the examination. The mini serves as a means of recording the test events and results, retrieving and presenting to the system under test those strategies which were previously successful in compromising similar systems, and serving as a powerful tool in assisting in the overall test process.

The technology which is being developed by RISOS is specifically intended to be applicable in the systematic and methodical examination or audit of an OS. Techniques are used which are designed to prevent the most sophisticated assaults on a computer's security features. It has been difficult to obtain information on actual cases in which any sophisticated subversion has been applied in order to illegally obtain either access or information. This is due in part to a great reluctance to reveal such occurrences. Governments are

reluctant to make it known that nationally sensitive information has been compromised. Businesses are similarly reluctant to make their clients aware of the fact that their computer security is lax to the point that a client's data is available to others — perhaps their competitors.

The only source of information which is available to us is in the court records of cases which have resulted in criminal or civil litigation. The result is that we cannot report on a large number of instances in which sophisticated compromises have been successful. Donn Parker of Stanford Research Institute has assumed the role of historian of computer criminal actions. He has published two reports on this subject. From information provided by Mr. Parker and from information gained by attending the trial proceedings we shall review one case in which special knowledge and circumstances resulted in a criminal violation of a computer center. As we go along, we shall point out certain security vulnerabilities.

The UCC-ISD Case

Information Systems Design (ISD) is a computer service bureau in Oakland, California offering Univac 1108 computer services. University Computing Corporation (UCC) is another computer service bureau which, at the time of this incident, was offering services, also on a Univac 1108, from Palo Alto, California. A third company, Shell Development Corporation was a client of both ISD and UCC. Shell purchased computer time by remote terminal connection from both companies. Shell was in the practice of using either provider to execute its programs. This provided Shell with a form of backup insurance which is advised by most experts on the physical aspects

of computer security. Shell was a good client, so that when Shell requested that its account number be the same at both UCC and ISD there was no hesitation at granting the request.

ISD, of course, had other clients, one of which was Aerojet General of Sacramento, California. A principal advantage of the ISD service to Aerojet was a graph-plotting program which had been proprietarily developed by ISD. The plotting program operated with unusual efficiency, thereby reducing the amount of computer connect time required to perform plotting. Attempts by UCC to undersell ISD at Aerojet were always hampered by the efficiency of the plot program.

Jeff Ward, an employee of UCC, embarked upon a plan to acquire a copy of the coveted plot program. Ward was a member of the UCC system staff and as such had intimate knowledge of the UCC system. He was unusually well prepared to attack the ISD system. His plan was simple. A visit to the common client, Shell, revealed that the phone numbers of both ISD and UCC were plainly visible on the wall near Shell's terminal. Returning to the UCC premises, Ward dialed the ISD number, used the common Shell account number, and gained access to the ISD machine.

It is not known whether there was a security feature which required a secret password in addition to the account number. It doesn't really matter since Ward was not detected in his surreptitious entry to the system. If there was such a password mechanism, it did not stop him, it did not record his presence, and it did not notify the ISD security officer that an attempt was being made to gain entry. All of these functions should be the responsibility of the OS in any system.

Once he was connected to the ISD computer, Ward quite literally browsed around inside the computer until he had located the plot program. His search was facilitated by the fact that Aerojet had innocently given the name of the program to UCC representatives. The name and location of the desired program was kept in a file directory, while the program itself was kept in a file; both items were thus resident in the computer. Access to either the directory or to the file should have been controlled. Here again, password and access checks should have been made by the OS. Current legislation which is pending before many state legislatures and before the congress give proper address to this issue by requiring that a record be kept of all accesses to personal data kept in computer data banks. Thus, the mere fact that Mr. Ward accessed the directory and subsequently accessed the file itself should have been recorded by the OS.

Ward caused the plot program to be printed on his terminal at UCC. He had succeeded in acquiring the program. Unfortunately for Ward, he did not stop there. Why not have the ISD machine prepare a punched card deck for him also? This would do away with the tedious and boring job of keypunching the program for entry to the UCC system. In making this request, Ward overlooked a feature of the ISD system. Whereas the UCC system evidently punched decks on customer's own card punch units, the ISD system did not. Instead, it punched the decks at ISD and had them physically transported to the client's premises. And so it was that the deck in question showed up at Shell, was rejected by Shell as not having requested any such deck, was

returned to ISD and was finally identified as a copy of the much valued plot program. ISD was now alerted and began to investigate.

Telephone taps revealed that Mr. Ward was still hooking up to ISD. His location was identified, and a search warrant was issued. Among the usual potpourri contained in the search warrant was the unusual demand to search the UCC computer. How do you do that? How many law enforcement agencies are equipped to carry out such a directive? In this case, a representative from ISD accompanied the law enforcement officials to the UCC premises. At his request, the contents of memory and the contents of various tapes and discs were printed out and confiscated. Naturally, the ISD representative examined the printouts for purposes of identifying the stolen ISD property. This whole course of action is very strange to attorneys and humorous to knowledgable computer people. Part of the computer's contents had to be the records of all valid UCC clients, their account activity, and any data files which they were, in good faith, maintaining on the UCC machine. These data files could contain information which these companies held to be proprietary and confidential in nature.

The entire ISD-UCC story might be viewed as a comedy of ignorance. That would make the experience a humorous one. Unfortunately, it was a very serious lesson for all of us. It is obvious that neither ISD nor UCC had previously paid much attention to security — and I mean all aspects of security. The physical, administrative, operational, and software security of ISD, Shell, and Aerojet contributed to the success of the incident. These three companies are not that much different from the companies represented here at this conference as far as security awareness and security practices are concerned. Although

no suits resulted from the information leak resulting from the search of the computer, we must look at the potentially contributive role which the act of law enforcement may have played. It certainly serves to highlight the need for more understanding of appropriate procedures. As far as Mr. Ward is concerned, he feels that he has done no wrong. Donn Parker reports that this is the usual reaction on the part of computer related people who have stepped across the legal line. Finally, as was the case with the ISD-UCC trial, the courts do not know how to analyze the complexities of computer technology which arise in a litigation. From a law enforcement point of view I think this means that much more care must be used in collecting the evidence to be used.

Conclusion

This paper has illustrated why crimes involving a computer should be on the increase. Sophisticated methods of gaining access to computer based data were reviewed. An in-depth analysis of software security mechanisms represents an effective countermeasure against unauthorized or accidental disclosure, modification, or destruction of computer contained data.

END