DETECTION OF COMPUTER CRIME (POUR DETECTER LE CRIMINAL DE L'INFORMATIQUE)

John M. Carroll
Computer Science Dept.
University of Western Ontario
London, Canada

ABSTRACT

It is difficult to detect computer crime because magnetic media shows no trace of forgery and entry to computer systems is largely anonymous. Most computer criminals have been caught by traditional methods. However,the computer has the potential for defending itself if properly instructed. Some techniques that may prove useful include threat monitoring, use of security consoles, and development of audit trails. (C'est difficile detecter le criminal de l'informatique parce qu'on peut changer l'information sur une ruban magnitique sans evidence et on peut entre dans une systeme d' informatique sans donner son nom. On a detecté la plupart des criminals de l' informatique par les moyens traditionalles. Cependent, il y a quelque moyens modernes pour defendre l'ordinateur: l'announcement automatique d'une attaque; l'utilization du moniteur de securitié; et la verification de la traffique.

Computer crime is hard to detect because unlike conventional documents magnetic media undergo no change in appearance when altered, and the actions of a person unlawfully accessing a computer terminal can be cloaked in perfect anonymity.

In 54 incidents occuring during the last ten years only two came to light because of procedural safeguards. Ten were uncovered when the perpetrators were found in possession of compromising print-out, or tried to sell the results of their penetration. Eight incidents came to light because of information furnished by informers. In six cases the

perpetrator was observed by witnesses; one was seen on closed-circuit television. Five incidents became known because of unsolicited confessions.

Eight incidents were revealed subsequent to discovery of shortages in financial accounts; three subsequent to discovery of shortages in physical inventories; and three pursuant to investigation of customer complaints.

Five incidents were unearthed during investigation of conventional crimes; two when major changes in the system were undertaken, and one when the defalcator went on vacation and was unable to intercept compromising documents.

Thus for it may be said that the detection of computer crime has followed a pattern not remarkably different from the detection of any other type of white-collar crime. What is significant is that the inherent capability of highly "intelligent" machines to detect their own misuse remains largely unused.

THREAT MONITORING

To play an active role in its self-defence, a computer should possess the capability to log in machine-sensible format the details of all apparent violations of security and to scan these logs and retrieve information selectively from them.

The transactions upon which attention should focus include: entry to the system, entry to sensitive files, execution of certain specified programs, allocation of systems resources such as peripheral storage devices containing sensitive information, and all input and output transactions.

Moreover, the recording of these security relevant transactions should be accomplished in such a way that the log is protected against unauthorized intervention by the operator, users, or systems personnel. A good case can be made for the use of a dedicated minicomputer to control the logging function.

An immediate problem arises of how to recognize a security violation. Certain events deserve to be viewed with suspicion in this regard. These include:

1. The abnormal termination of a job.
2. An abnormal shutdown of the system.
3. Failure of any hardware or software protection mechanism.

4. Unsuccessful attempts to log-on the system.
5. Unauthorized attempts to access sensitive files.
6. Unauthorized attempts to use privileged instructions or sequences of them.
7. Attempts to exceed the user's allocated address space.
8. Unauthorized attempts to access systems resources.

When any suspicious action is apprehended, the recording mechanism should be able to specify the identity of the devices involved (terminal and storage device, for example), the type of violation, the identifier appropriated by the user, the date and time of the incident, and the identification of the file, program or resource being sought.

SECURITY CONSOLE

The proposition has been advanced that there should be a control console at which a security officer would sit and grant or deny access to the system or any of its critical resources. This would appear to be the very antithesis of modern information systems design.

There is merit, however, in having a console that would filter the information that commonly inundates a computer operator and give an immediate warning of incidents such as a third successive and unsuccessful attempt to log-on; improper response to a request for re-authentication, or an attempt to use an unauthorized instruction or instruction sequence.

Warning could also be given of acts such attempts to exceed authorized address space, attempts to gain unauthorized access to system resources, any attempt to read newly allocated memory without first having written on it, or any unauthorized attempt to enter a more highly privileged operating state.

The security console could likewise warn of the failure of any systems protective mechanism or of the inability of the system to verify successfully any systems program, security table, or address relocation table.

ENTRAPMENT

It is a good idea to salt lists of names or other sensitive data files with dummy entries so that if the list is, for example, unlawfully sold to a direct-mail house, the security officer will become aware of the offence when he begins to get solicitations at drop addresses intended for

fictitious persons he has invented. With a little ingenuity all kinds of trap accounts, files and programs can be devised to lure a perpetrator into penetrating them and thus reveal his presence.

ANALYSIS OF TRENDS

There is an intellectual attractiveness in catching crooks by statistics. It does, however, entail a good deal of planning and foresight. One must first develop the capability to sense significant deviations from expected patterns in systems activity and useage of resources.

One will, by the first instance, have to develop base data from the study of console logs, terminal logs, systems accounting records, records of hardware/software failures and subsequent restorations, and any data available from hardware or software monitor probes.

Naturally systems utility programs and editing routines will have to be developed to extract the security-relevant facts from so great a mass of data.

The products of these analyses will then be activity profiles specific to the system as a whole, to each user of the system, each terminal, all sensitive files and programs.

Table I suggests some of the items that might constitute an activity profile and conjoins them with the specific profiles in which they would be most applicable. There is an implicit time dependency connected with every profile constituent.

AUDIT TRAILS

An EDP system should possess the capability to construct an audit trail of all accesses to sensitive information and use of critical resources. Documentary evidence must be retained so that each step of an audit trail can be independently confirmed.

In a system that deals with extremely highly classified material, one should be able to develop the history since first employment of any given staff member in respect of access privileges internal to the computer system having been granted and altered, or revoked. Similarly, one should be able to develop the history, location, and current status of any access-control item used internally within the system. (e.g., passwords and password lists.)

There are three kinds of audit trail that should be kept irrespective of the security level of a system.

1. It should be possible in respect of any file deemed to be sensitive to go back at least six months and reproduce the transactional history of any given record.

2. It should be possible to reconcile transactions with authorizing documents for a minimum of three machine-sensible generations of any data file.

3. It should be possible to reconstruct the history of any sensitive systems or applications program in respect of its development and testing, implementation, maintenance, utilization, and modification, if any.

COUNTER - INFILTRATION

The activities of so-called Tiger Teams is the subject of controversy. Their use derives from the old adage "Set a thief to catch a thief". The teams are made up of young, highly competent and imaginative systems analysts and their mission is to penetrate surreptitiously the defences of EDP systems, usually those of the remotely accessed, resource-sharing variety.

The dangers are obvious. Any defalcator, if caught, could aver he is, infact, a Tiger. Moreover, some Tigers may decide to change their stripes and become defalcators themselves. Furthermore, argument has been advanced that Tiger-Team attacks are forever doomed to be largely unproductive inasmuch as they seek to prove a negative, which is always very difficult to do.

CONCLUSION

I expect this paper has done more to raise questions than to provide answers. If so, it well represents the current state-of-the-art in the field of detection of computer crime.

Scientists of a mathematical persuasion who have addressed themselves to this problem have often sought a neat configuration of logical safeguards that are proveably complete and correct.

I tend to be vastly skeptical of any such putative solution. What one human mind can do, another human mind

can undo and correctness is proveable only within the system; it is not possible to prove that the system itself is correct.

Quoting Robert Service: "It is always the slow and the plodding ones that win in the lifelong race."

The best answer to the detection of computer crime seems to lie in the direction maintaining detailed records regarding activity at an EDP centre and reducing these data by means of heuristically based programs valid within the local context to obtain evidence of penetration or defalcation.

In Table II, I have enumerated twenty danger signals that may serve as an initial checklist for anyone vested with the responsibility of detection of computer crime.

TABLE I

INCIDENTS TO BE RECORDED

Profile Constituent	SPECIFIC TO: SYSTEMS	USERS	TERMINALS	FILES	PROGRAMS
Restarts	X				
Unsuccessful log-ons	X				
Communications errors	X				
Program reruns	X				
Hardware/Software failures	X				
Down-time per shift	X				
Successful log-ons		X	X		
CPU time/Core per job		X	X		
Security Incidents		X	X		
Changes to Authorization Tables		X	X		
On-line memory residence		X			
File accesses		X			
Instructions per job		X			
Supervisor calls per job		X			
I/O requests per job		X			
Connect time per job			X		
reads per job (run)				X	
write (appends) per job (run)				X	
Unsuccessful executions					X
Successful executions					X
Time per execution					X
Output per run					X

TABLE II

CHECKLIST OF INCIDENTS DESERVING INVESTIGATION

1. Compromise of classified information,
2. Loss of any sensitive asset,
3. Unexplained operator intervention in the running of any job,
4. Presence of an intruder,
5. Unexplained absence of any person possessing access to sensitive information,
6. Unexplained appearance of trap names on an extra-departmental file.
7. Attempts to access trapped records or files, or use of trap passwords,
8. Unexplained increases in systems usage especially during off hours or normally silent periods,
9. Loss of identification, access-control, or recognition items, or unauthorized use of same.
10. Unexplainable client complaints regarding events including improper billing, mis-addressing, incorrect balances, or omitted,improper or incorrect payments,
11. Inability to balance any account,
12. Unexplained inventory shortages,
13. Inability to reconcile cheques,
14. Unexpected frequency of unsuccessful attempts to obtain service or other systems protocol violations,
15. Excessive demands for input or output,
16. Unexplained changes in patterns of communications traffic,
17. Unexplained appearance of new code in operating system or utility programs,
18. Unexplained changes in job profiles,
19. Unexplained accesses to classified files,
20. Unexpected incidence of hardware of software failures.