Biological *versus* Computer Viruses
A better understanding for a better defense

by **Dr. Daniel GUINIER** , *ACM , IEEE and IACR member.*

Strasbourg University ,
Centre National de la Recherche Scientifique.

Abstract

To understand biological viruses, some notions of the fundamental knowledge of the structure of DNA, the genetic code, the biosynthesis of proteins, the transcription, replication and transfer processes,... are presented so as to give an idea as to how the genetic information is decrypted by biological mechanisms and consequently, how viruses work.

A computer "virus" can be defined as a piece of code with a self-reproducing mechanism riding on other programs which cannot exist by itself. In contrast, a worm can exist independently. A computer 'virus' can be considered as another category of computer user, the problem of protection against such a 'virus' can be reduced to the problem of protection against users.

The choice of the term **Self-Reproducing Program (SRP)** appears to be unambiguous in comparison to the word 'virus'. After having created the computer in 1948, **John Von Neumann** said in 1949 that it must be possible to imagine logical or mechanical engines that would be able to be self-reproducing. We propose that **"good" SRP's should be useful** for the automatic maintenance of software, by infection of old versions by the most recent version in the form of such an SRP.

Protection is possible by a **better understanding of computer systems** and their mechanisms of exchange of data and processes. Such a study is presented for the DOS which should be protected by a watchdog system and suggests the need for a real-time analysis on the most vulnerable points. Security models including Cryptography should offer preventive solutions and 'vaccines', the treatment of minor troubles...while **prevention** requires a **better understanding of men and their ambiguities.**

The idea that there is a need for a better knowledge of *SRP's, Worms, Trojan horses,..* justifies a **call for the constitution of a special database** concerning them.

Key-words : Computer virus, Biological virus,Trojan horse, Worm, SRP, Core war, Computer security, Data integrity, Program integrity, WORM, Cryptography, Watchdog, Security policy, Database.

Mailing address : 14 rue du Bitzen,
F 67200 ECKBOLSHEIM, France.

Biology and Computer science

1.On Biological viruses - Computer viruses -Trojan horses - Worms

1.1. Preamble to biological viruses and associated mechanisms

Before understanding biological viruses *and the way they work, we need to review our knowledge on the structure of genes and associated biological material. Here I build upon my main work in bio-computing at the French government research lab : Laboratory of Molecular Genetics of Eukaryotes of the Centre National de la Recherche Scientifique (CNRS Strasbourg) directed by Professor Pierre Chambon (Pendlebury D. (1989)) and also to try to answer to a remark from a French colleague involved in research in Mathematics :* ***"Cryptography... what is the relationship with Genetics ? ".***

In 1944, it was determined by the works of O.T.Avery, M.McCarty and C.M.Mc.Leod that genes have a precise chemical nature : they are composed of Desoxyribo Nucleic Acid (DNA) molecules.

1.1.1.Structure of the DNA

A biochemical structure for DNA

The molecule of DNA is a long molecule composed by a serial chain of a chemical group : a **phosphate** and the **desoxyribose**. On each of these phosphate-desoxyriboses is attached a nitrated **base** to form the **nucleotide**. There are only four different bases :

two **purines**	Adenine	(A)	Guanine	(G)
two **pyrimidines**	Thymine	(T)	Cytosine	(C)

and the genetic information is contained in a very simple language using a 4-letter alphabet : { **A , T , G , C** }.

A physical structure for the DNA

James D. Watson and F.H.C. Crick proposed a physical structure for the DNA : the molecule of DNA is maintained as a stable double helix which enable us to establish the chemical complementarity of the two strands to each other and to understand how the genetic information can be replicated and transmitted to new generations.

The association of the different units of nucleotides is mediated by strong covalent chemical links and the two strands are intertwined to form the double helix, maintained by single hydrogen links which are easy to break and permit the association of complementary pairs of bases (BP) : G-C or A-T or C-G or T-A.

1.1.2.The genetic code

The **genes** are **fragments of DNA** whose role is to determine the structure of **proteins** which are constituted by simple biochemical elements called **amino-acids**. There are **only 20 different amino-acids** involved in such a process in biological systems.

A gene is able to determine the nature of the successive amino-acids and we can say : 'this gene codes for such a protein' because there is a law of correspondance : **the genetic code**. The rule is : **to three successive nucleotides (codon) corresponds one given amino-acid.**

A degenerated code : If only a 4-letter alphabet is given, in one codon (3 letters), we should build 64 different words and not only 20. The genetic code is said to be "**degenerate**" and some codons are said to be "synonymous" , they are different but code for the same amino-acid.

1.1.3.The replication of DNA

The copy of DNA is realised under the contol of another **enzyme** : the **DNA polymerase.** Here the rule of incorporation of bases is : G-C or A-T or C-G or T-A. The two strands are first physically separated and then independently copied according to this rule.

1.1.4.Protein biosynthesis

Transcription of mRNA : The **genetic information is not directly decrypted from the DNA,** there is an additional process known as "**transcription**" which operates on gene after gene or groups of genes using an **enzyme** : the **RNA polymerase.** The result of the copy is a RiboNucleic Acid (RNA), the "**messenger RNA**" : **mRNA.** The **mRNA** has just one strand and, by comparison with the DNA, has the base Thymine (T) replaced by Uracil (U) to give the rule of incorporation of bases : G-C or A-T or C-G or U-A. The reading frame is in the single direction 5' --> 3' (expressed in universal carbon atom numbering).

Role of the maturation of the pre-mRNA : In *Eukaryotes,* more evolved organisms which have a nucleus in the cell (mammals for example), the information necessary for the biosynthesis of proteins is fragmented : a gene consists of **coding sequences,** the **exons,** separated by long non-coding sequences, the **introns.** Introns and exons are copied to form the premessenger RNA (pre-mRNA) which is submitted in the nucleus to a set of reactions for maturation which remove introns and form the definitive mRNA.

Role of the mRNA : The mRNA obtained passes through the nuclear membrane and a translation system is initialized **to decrypt information** : amino-acids are assembled in the **cytoplasm** at the level of the **ribosomes,** which consist of RNA and proteins, this complex association permits the translation of the genetic code and the synthesis of proteins.

Role of the tRNA : Because there is no direct relation between the mRNA codons and the corresponding amino-acids, an interface is necessary involving "**transfer RNA**" : **tRNA** which is obtained as a copy of a DNA segment under the control of the RNA polymerase. This tRNA has an anti-codon complementary to the codon. At the level of the ribosomes, the tRNA reads a codon from the mRNA using its anti-codon and delivers the corresponding amino-acid to the growing chain. The elongation of the chain continues until it finds a codon for the end-of-translation (*'stop'* codon), at this moment, biosynthesis of the protein is finished.

These proteins are structural elements of membranes, ribosomes ... therefore such elements are used in their own biosynthesis,the whole achieved with only a 4-letter code : (A, T, G, C).

*To the question : **Cryptography ... what is the relationship with Genetics ?** We can now answer : **Genetics ... what a powerful Cryptosystem !***

1.1.5. ... and now,viruses

A virus can be defined as a self-reproducing biological entity which uses the cellular mechanisms and material to reproduce itself and cannot exist by itself.

There are four main categories of viruses : DNA or RNA viruses each with either single or double strands, whose size varies from 3000 to 300,000 base pairs (BP).

The known DNA or RNA sequences, not only virus sequences, are annotated and stored in nucleotide databases such as **GenBank**, the genetic sequence data bank (release 19) from the Los Alamos National laboratory or the **EMBL** nucleotide sequence data library (release 59) of the European Molecular Biology Laboratory (Heidelberg, RFA), they contain around 25,000 annotated entries totalling more than 30,000,000 codes.

Now that we have reviewed the tools used by biological viruses, we can write **the simplest of viral life protocols for such an hypothetical virus :**

At the beginning is ***one virus*** *--->*

.Entry into the cell : injection of viral DNA or RNA in the host cell.
.Transcription and replication mechanisms.
.Translation mechanism.
.Assembly of progeny virus particles.
.Exit from cell.

---> At the end : ***a lot of viruses.***

1.2.Computer "viruses"

A computer "virus" can be defined as a piece of code (program) containing a self-reproducing mechanism riding on other programs and which cannot exist by itself.

The use of the term "virus" is clearly appropriate considering the previous definition of biological viruses.

Later, when a normal host program is run, it will execute this set of instructions which will infect other programs, triggering the trojan portion of the program contained within the "virus".

The danger : the epidemy !

A "viral" code can cause damage to the hard disk and infect other programs which in turn infect other programs. It can also infect programs on floppy diskette and there after...any user can unknowingly infect other machines, in the present location or connected to a network ! (e.g. the Internet "virus" of November 2nd 1988 caused an epidemy on 6,000 host computers implementing TCP/IP protocol ! (Eichin M.W. and Rochlis J.A. (1989)).

1.2.1. Is the word computer "virus" realistic ?

It is probably interesting to compare this new kind of program with existing concepts but, for most people, the word *virus* refers to Biology and has another significance : for naive people, **it is the worst** to choose because it is probably confusing; for newspapers, **it is the best** ! (After the paper from DeWitt P.E. (1988) published by TIME magazine, many more have been published in the newspapers, most of them with humorous titles...). We prefer to use a more general term : **Self-Reproducing Program (SRP).**

... and John Von Neumann created the computer in 1948... then, one year after... this notion of self-reproducing design : *in his paper "Theory and organization of complicated automata" given at the Institute for Advanced Study at Princeton he said that it must be*

possible to build an engine, logical or mechanical, that would be able to be self-reproducing (Penrose L.S. (1959)).

*The first SRP's were mentioned in the 1980's by Thompson K. (1984) in his Turing award lecture : "**Reflexions on trusting trust**" and by Cohen F. (1984,a and b) who used the term 'virus'.*

1.2.2. Self-reproducing mechanism in programs

The description of such mechanisms are detailed in Thompson K. (1984), Cohen F. (1987), Dewdney A.K. (1989) and Highland H.J. (1988).

1.2.2.1. Description of SRP's

Definition

***Fingerprint** is the identifier which is a part of the code used by a SRP to recognize itself to avoid a new infection in a given program. Because infection is exponential, a 'virus' will infect only once a program so as to avoid relatively rapid detection due to a considerable increase in the size of the infected program in a very short period of time. Such a reaction speed marks the difference between electronics and biological processes.*

Trigger** is the function which verifies that the conditions for action are suitable (time, counters to a value,...) and gives authorization for "**ACTION**" **for better or worst.

Scheme of a simple viral protocol

*At the beginning is **one SRP** --->*

```
_____        Phase 1 : INFECTION Process        _____

A :     .Search in a given directory if file.
            .If (file is found) then
                    .If (fingerprint in file) then
                            .Change to next entry in directory
                            .go to A :
                    .else
                            .Copy the SRP code : The infected code of
                            host must start by the Search function and a
                            move is needed to avoid to overwriting code.
                            .Save preinfection entry of host.
                    .end if
            .else
                    Change directory
                    .go to A :
            .end if
_____        Phase 2 : ACTION Process           _____

        .If (trigger) then
                ACTION process
        .end if

        .go to preinfection entry in host program
```

*---> Now : **a new SRP.***

*---> Later... **a lot of SRP's.***

1.2.2.2. Writing SRP's under MS-DOS

To the question : ***"Is it easy to write such a program ? "*** we can answer "***Yes***" and to illustrate this, Burger R.(1987) proposes several naive "virus" codes written in different languages. We recommend experiments on **a:** when MS-DOS is on this disquette under directory DOS, rather than on **c:** disk !

Remark : *.The last version given by this author uses only six MS-DOS commands : ECHO, PATH, CTTY, DIR, EDLIN and DEBUG !*

.The resulting SRP obtained overwrites host programs (.com, .exe, .bat), if the starting operation directs a move from the appropriate place, it should be more efficient !

1.2.2.3.As programmers, "good" SRP's could be useful

A "virus" can be considered as another category of computer user so that the problem of protection against such a "virus" is reduced to the same problem of protection against users.

The SRP's called "viruses" are destructive, but it should be very interesting **to use such processes for automatic maintenance of software.** As an example : Suppose several packages use an inefficient version of a procedure or routine, **in large organizations, it should be easier to update such a package by such an approach rather than to recompile and link all these packages,** especially in case of large distribution. It is just necessary to use one more fingerprint in the host programs as a protection so as to avoid infection (modification) when it is not required. Each new release could be marked by a new fingerprint, and so on.

1.3. Worms

A **worm** is a program which **can make copies of itself** and can pursue an **independent existence without the necessity of host code** (Dewdney A.K. (1984,85,87 and 89)).

1.4. Trojan horses

A **trojan horse** is a functional piece of code introduced in a program dedicated to secondary tasks to benefit the person who has implemented this harmful code, for example to deteriorate the performence of systems, to destroy data or to obtain information.

Discovering "viruses" & Co."

2.Computer "viruses" - Trojan horses - Worms

2.1.Understanding the vulnerability of MS-DOS

Why we do pay attention to MS-DOS ? Because, it is

easy to understand,
vulnerable by the absence of privileges,
among the most known and sold systems.

2.1.1. MS-DOS disk organization

It is necessary to understand MS-DOS and disk organization to understand how "bad things" can arrive in such a system (see King R.A. (1984))

A disk or a diskette is organized in **tracks** or cylinders (a 5"1/4 diskette has 40 or 80 tracks depending on whether it is double or high density), each track is divided in **sectors** of 512 octets each. Each **cluster** is made of an integer set of sectors. The first 12 sectors numbered from 0 to 11 contain the basic information :

- The **boot** sector (sector 0)
  contains information on disk and the boot program and instructions to branch to IBMBIO.COM and IBMDOS.COM

- The **File Allocation Table** (FAT) (sectors 1 and 2)
  contains sensitive information for the DOS such as the disk type, the status of each cluster (free, used or bad) and the links of files.

- The File Allocation Table **second-copy** (sectors 3 and 4)
  is a replication of the first File Allocation Table

- The root **directory** (sectors 5 to 11)
  contains 32 octets of information on each of the files stored : eight for name, three for extension, size, date and time of creation, attribute (system, hidden, read only, archive) (When a file is "deleted", only the first character from the the eight of the name is changed by E5H, to indicate a futur possible entry in the directory. Also it is possible to restore such a file immediately by an appropriate utility (e.g. Explorer, PCTools,...).

<u>if the diskette is bootable</u>,
when formatted with the /S option, the components of the MS-DOS system follows :

- IBMBIO.COM (sectors 12 to 21)
- IBMDOS.COM (sectors 22 to 55)
- COMMAND.COM (from sector 56)

> **IBMBIO.COM** is the interface between the heart of the operating system and the input/output of the PC. At the beginning is a set of jump instructions to branch to programs of input/output used for keyboard, serial RS232 port, output on printer, date-and-time.
>
> **IBMDOS.COM** is the program for management of transactions on disk by the use of buffers and other utilities.
>
> **COMMAND.COM** is the Command Line Interpreter, the interface between the keyboard and the MS-DOS operating system itself. This interpreter is divided in two parts, <u>one is resident</u>, situated just after IBMDOS.COM, <u>the other is loaded</u> in the high memory part and can be erased by large, overloaded programs.

<u>If it is not a bootable diskette</u>, data are stored from sector 12.

The description of the disk organization shows the vulnerability of such a system in the first 11 sectors : FAT's, sectors 1 to 4 and in the directory, sectors 5 to 11, this vulnerability should be exploited for "viruses", worms and trojan horse strategies.

2.1.2. System activity with MS-DOS

Interrupt vectors

The Intel 808xxx family microprocessors use software interrupts. Interrupt vectors are loaded in the first 1024 octets of the memory. Each vector is the 4-octet address of the corresponding program which performs the interrupts. The first 16 Interrupt vectors are reserved for material interrupts, BIOS-ROM and MS-DOS use the 32 next interrupt vectors. Interrupt vectors for MS-DOS are from int 20H to int 27H. Some of these vectors require special attention, such as :

The interrupt 26H used for absolute write on disk

The interrupt 13H used for diskette access

The interrupt 21H reserved for the call to DOS functions

Some of the strategic **functions of interrupt 21H are :**

EXEC	to load and run a process (program)		4BH
	to stop a process (exit)		4CH
	to report on a finished process		4DH
File and directory access			
	for global access	Open a file	3DH
		Close a file	3EH
		Delete a file	41H
		Create a file	3CH
		Rename a file	56H
	for read and write in a file	Read	3FH
		Write	40H
	for search for files in directory		4EH or 4FH
	to change the size of a file		28H

2.2.Some examples of possible "virus" families under DOS

"*In situ* source code viruses"
they make part of the originally created source code before compilation as trojan horses.

"Intrusive viruses"
they insert themselves in host programs during a short period of time, they are in action when the host program is running, doing their ugly task; the executive velocity of the original program could be degraded,...your hard disk files also !

"Shell viruses"
they do not attack directly a given program but they inject themselves inside the host program respecting the order : original program part 1-"shell virus"- original program part 2 keeping the original date without substituting the current date.

"Calling viruses"
they are reduced to a minimum number of instructions, the manipulation function, heart of the program can exist in a single hidden sample outside of the infected host.

"Good sectors macaradized viruses"
They mask good sectors in bad sectors... where they can operate.

2.3. Example of a "virus" under MS-DOS : *The "Brain Virus"*

Highland H.J. (1988) described in detail this well known "virus" also known as the "Pakistani virus" because of its creators in 1986, Basit & Amjad, Brain Computer Services... it is sufficient to mark the fingerprint of the "virus" (1234) that is to insure that 34 and 12 mark the 5th and the 6th first octets of the sector 0000 of the system boot to be EB 34 90 49 **34 12** ...to simulate an infected disk to avoid infection, regarding EB 34 90 49 **42 4D** ...for a non infected disk. The "virus" is divided in two pieces : the first on the boot sector and the other in six sectors (3 clusters) corresponding to the remainder of its code and a copy of the original boot sector marked as *bad* in the File Allocation Table.

2.4. A first lesson : *The internet "virus" of November 2nd 1988*

Eichin M.W. and Rochlis J.A. (1989) at the recent IEEE symposium on Security and Privacy, May 3rd, in Oakland, gave a complete analysis of **the Internet self protected "virus" of November 2nd 1988 which infected no less than 6,000 machines** under Berkeley Standard Distribution UNIX. They give a complete description of the program, subroutine after subroutine. The name "virus" was discussed in the paper and it was decided that its use was accurate.

entering by

sendmail	(Sun)	
fingerd	(VAX)	(using a finger daemon bug)
rexec	(remote)	(using */etc/passwd* file)
rsh	(remote)	(using */etc/hosts.equiv* and */.rhost* files)

fortunately, it does not, <u>in general</u> break in as
root
... and did not gain privilege access !

compiled and named *sh*
like the Bourne shell command interpreter, and constants strings used by it were encrypted by a single xored function with a single constant.

One open problem : *the password encryption under UNIX*
/etc/hosts.equiv file contains the user encrypted passwords; the UNIX password encryption algorithm used is based on the well known NBS Data Encryption Standard (DES) , (see Diffie W. and Hellman M.E. (1977) , Morris R. , Sloane N.J.A. , Wyner A.D. (1977). Two problems remain in the UNIX implementation of the DES :

- new algorithms are more and more powerful
- computers run faster and faster

that is a fast implementation of *crypt ()* could work fast and break the password to enter anywhere in the system, even in *root* . Solutions should be :

- to hide the passwords rather to let them be public-encrypted thus taking the risk to reveal the password in clear when the privileged account is cracked.
- to replace the actual implementation of DES by the fastest available implementation of the DES, or other cryptographic

algorithms : Guinier D. (1988) proposed Sharing Partial Key System, a secure threshold bi-dimensional system which permits to share access availabilities with identification.

Diversity in operating systems gives a better protection
As in nature, diversity is in favor of a good evolution, rather than a single standard operating system variety offers much security.Also, at host level, attachment of new local defenses including a high level logging information could be more efficient than security in the network.

3. How to protect systems against "viruses"-trojan horses,worms

We have to consider two different ways to protect systems against computer "viruses" : preventive and curative. We should remember that the cure that consists of **restoring a destroyed system from good backups is sometimes better than to try to figure out the damages !**

Customer policy
Do not accept any "exogenous" software, it could be infected ! *(e.g. in our lab, benign "viruses" have infected diskettes of more than 200 users, creating minor problems and have required the use of "vaccine").*

Intrusion models
Several models have already been designed, Denning D.E. (1987) proposed a model for a real-time intrusion-detection expert system, including profiles and audit records. The model is system-independent and provides detection of penetrations and security violations.

Watchdog processors
Concurrent error detection using watchdog processors has been described in Mahmood A., McCluskey E.J. (1988)).

Cryptography
Davida G.I., Desmedt Y.G., Matt B.J. (1989) present a system able to defend computer systems against "viruses" through cryptographic authentification under distribution. Their master idea is that the vendor generates a numerical signature that may be verified by the user who customizes the software, producing a user locally executable code under control of a device named as the *authenticator*. It is used randomly to control if the operating system uses correctly the device.

Optical WORM disks *(Write Once Read Many, not worm !)*
Originally delivered information (programs, data) is stored on optical WORM technology disks making it impossible to modify this information.

"Anti-virus" programs

What should an "anti-virus" program do ?

A typical "anti-virus" under DOS could be a program supervisor with the following features :

Checksumming CMOS RAM

In PC-AT type machines, important parameters (type of hard disk in use, memory size,...) are stored in a non-volatile memory, called CMOS. If these parameters are changed, a problem occurs when the system is rebooted. A CMOS RAM check needs to be done to see if a foreground program attempts to change CMOS memory and to advise user. A digest is associated to the file name and can be recomputed as equal by an elementary hashing function.

Permission for files

Permission to write and/or read protect full classes of files. Write / read protected or protection excluded files have to be named in a security file.

Write protect recommended files : *.COM
*.EXE
*.SYS
*.BAT
Some drivers, Write processor fonts,...

Read protect recommended files : AUTOEXEC.BAT
CONFIG.SYS

***Remark** : Floppy diskettes could be write protected as soon as possible when current operations involve only 'read'.*

Checksumming files

Automatic check for programs to determine if they have changed since you last looked at them or each time the program is loaded to run or after a given period of time in fractions of second. Other easy verifications on date, time and size can be done.

Frequently checksummed files : COMMAND.COM
IBMBIO.COM
IBMDOS.COM

Look for "Terminate and Stay Resident" programs

Reveals any program that attempts to terminate and stay resident before having been legitimated first by the user in a security data file.

Authorization to run for sensitive programs

Programs like the DOS *FORMAT* program need to be under control of actual rights user.

Protecting the security file and the "Anti-virus" program

The security file is a sensitive file in the middle of a watchdog system, it needs special attention for protection, under superuser password. "Anti-virus" programs need also to be self-protected, remember the virus disguised in Flu_shut 3 the "anti-trojan / anti-virus" ! (Jackson K.M., Hruska J. (1988) give list and short presentation of such products for "virus"-detection or prevention and cryptography in *The PC Security Guide 1988/1989*)

Flu_shot+ the "anti-trojan / anti-virus" for preventive protection

Flu_shot+ is an "anti-trojan / anti-virus" protection program which has been developed by Greenberg R.M. (1989). It is **distributed as a shareware**, the last updated version is 1.6 (May 1989).

A new product named as **Flu_shot++** - *"the ultimate in computer security ! .." (sic)* will be available **as a commercially distributed product** an enhanced version of Flu_shot+ with an easier method of implementing these protections from next July !

Additional features :

Password protection of certain directories.
Data encryption.
Disk disaster recovery.
Permission enabling only authorized programs to be run.
Prevention of unauthorized copying of files off the system.
Logging of attempted security violations and "anti-virus" triggers.

3.1. Physical protection

Appropriate backups

It is important to **do regular, full and adequate backups.** They can be done under the control of specialized programs (e.g. FASTBACK+). The use of low-price **extractable disks to associate efficiency of random-access devices and security is recommended** (e.g. when using 30 MO E-PACK's (Tandon), a complete backup takes about 3 mn.).

Write protect access on disks

Diskettes need to be physically write protected, as soon as possible in respect to their read-write activity. We have to regret the absence of standard write protection on harddisks for microcomputers ...their mini's brothers have one ! Do not forget to store the write protected diskette containing the original system and applications in a safe place, immediately after delivery, you will probably need them at least once !

3.2. Logical protection

The PC Security Guide 1988/1989 byJackson K.M and Hruska J. (1988) presents the most commonly found products commercialised, including a technical description and evaluation.

Gligor V.D. et al. (1987) present the design and the implementation of Secure Xenix which is an experimental integrated system to run on PC's. Their system includes supervision of control access privileges by mandatory access control, classification for files, directories and processes, protection of authentification and authorization data, user identification and audit mechanism.

3.3. Legal protection

The recent French law no. 88-19 of January 5th. 1988 (Art.462-2 to 8) **relative to computer fraud** is adapted to intrusion, illegal use, modification or destruction of data,... considering people, but is a little ambigous when considering "viruses". Probably, computer evolution and the social consequences go so fast that new amendments will be necessary in the next years. This law finds its **equivalent in the United States Criminal Code, title 18, sections 1030.**

... but what of the case of a late "***post-mortem* virus**" that could operate ten years or more after its installation ! Another problem : the presentation of a prosecutable case and prediction of its outcome is difficult because **few judges or jurors have a sufficient knowledge** of computers and their tricks.

4. Who are computer "viruses" makers ?

Educational measures including those making people sensitive to the immediate and long term consequences of such acts combined with a better security organization would probably improve computer security. This raises one question : ***"Who are these computer "viruses" makers ?"***

It is difficult to give a typical portrait of such white neck bad guys because most of the cases are unknown to the judicial and police authorities because of the difficulty of obtaining proof and because real cases are not given publicity, especially in the financial world. However, it appears that at least one of the following motives : **test of *skill*, game, profit** or **vandalism** are characteristic of these acts. Men are sometimes ambiguous in their individual choices, actions, beliefs, attends of environmental responses, mixing necessity, intention and reality, (March J.G.; Olsen J.P. (1979)).

Considering the main conclusion of a 45-page report entitled *"The Computer Worm"* from the internal investigative commission at Cornell University, on the internet "virus" of November 2nd 1988 : ***"...Robert Tappan Morris,Jr., worked alone in the creation and spread of the Internet worm computer program that infected approximately 6,000 computers nationwide last November ..."***. A copy of the report is available upon request from the office of the vice president for information technologies , 308 Day hall, Cornell University, Ithaca, NY 14853-2801.

Morris R.T.Jr. was a first year computer science graduate student. He intended the "virus" to spread to all the host computers connected to the network and to self-replicate after infection but he did not intend that the "virus" should destroy data or alter the functioning of computers after penetrating them. He was completely aware of the consequence of his act.

Security flaws can be used by hackers before they are fixed !

The realization of such "viruses" requires more perseverence than real talent, many undergraduate students are capable of developing a "virus", if they know security flaws reported by UNIX developers. Hackers can use such flaws before they are fixed !

Conclusion

Computer security losses have an unsupportable cost per year in all computerized modern countries. Research for a better understanding at the same time of computer systems and of the ambiguities of people and what they do is probably necessary to avoid large scale problems in the future, we should remember the lesson of the November 2nd. 1989 "Internet virus".

We must also know more about SRP's, Worms, Trojan horses... for this reason, we present a separate short paper entitled :***"Proposal for a "C-VIRUS" database dedicated to SRP's, Worms, Trojan horses,.."***, it should help the academic community !

Bibliography

Alberts B., Bray D., Lewis J., Raff M., Roberts K., Watson James D. (Nobel Prize 1962 in Medicine and Physiology) **(1983)** : Molecular Biology of the Cell, Garland Publi, Inc., New-York & London, pp.1-1146.

Bell D.E., La Padula L.J. (1976) : Secure computer systems; unified exposition and Multics interpretation, ESD-TR-75-306, MTR 2997 Rev. 1, The MITRE Corporation, march.

Burger R. (1987), translated by C.Stehly(1989) : VIRUS, la maladie des ordinateurs, Ed. Micro Applications, Paris,pp.1-322.

Campbell A.M. (1976) : How Viruses Insert their DNA into the DNA of the Host Cell,*Scientific American*, December, Vol.235, No.6, pp.102-113.

Cohen F. (1984,a) : Computer Security, a global challenge, IFIP/Sec'84, Toronto, Proceedings published by North Holland.

Cohen F. (1984,b) : Computer Viruses, 7th. DoD / NBS Computer Security Conference, Proceedings, pp.240-263.

Cohen F. (1987) : Computer Viruses : theory and experiments, *Computer and Security*, January, Vol.6, No.1, pp.22-35.

Davida G.I., Desmedt Y.G., Matt B.J. (1989) : Defending Systems Against Viruses through Cryptographic Authentification, IEEE Symposium on Security and Privacy, May 1-3rd., Oakland, Proceedings, pp.312-318.

Denning D.E. (1987) : An Intrusion-Detection Model, IEEE Trans Soft Eng., Vol. SE-13, No. 2., pp.222-232.

Denning P.J. (1988) : Computer Viruses, *American Scientist*, may-june, Vol. 76, No.3, pp.236-238.

Dewdney A.K. (1984) : In the game called Core War hostile programs engage in the battle of bits, *Scientific American*, May, pp.15-19.

Dewdney A.K. (1985) : A Core War bestiary of viruses, worms and other threats to computer memories, *Scientific American*, March, pp.14-19.

Dewdney A.K. (1987) : A program called MICE nibbles its way to victory at the first Core War tournament, *Scientific American*, January, pp.8-11.

Dewdney A.K. (1989) : Of worms, viruses and Core War, *Scientific American*, March, pp.90-93.

DeWitt P.E. (1988) : Invasion of the Data Snatchers ! *A "virus" epidemic strikes terror in the computer world*, TIME Magazine, September 26th., No.39, pp.40-45.

Diffie W., Hellman M.E. (1977) : Exhaustive cryptanalysis of the NBS Data Encryption Standard, Computer, Vol. 10, No.6, pp.74-84.

Eichin M.W., Rochlis J.A. (1989) : With Microscope and Tweezers : An Analysis of the Internet Virus of November 1988, IEEE Symposium on Security and Privacy, May 1-3rd., Oakland, Proceedings, pp.326-343.

French Computer Fraud Law 88-19 (1988) : Law no. 88-19 from January 5th. 1988 relative to computer fraud, Journal officiel de la République française, Chapt.III, Art. 462-2 to 8, January 6th.1988, p.231.

Gligor V.D. et al. (1987) : Design and Implementation of Secure Xenix, IEEE Trans Soft Eng, Vol. SE-13, No. 2., pp.208-221.

Greenberg R.M. (1989) : Flu_shot+ (TM) : Anti-Trojan / Anti-Virus Protection, Version 1.6, January, Software Concepts Design.

Guinier D. (1988) : S.P.K.S. : Sharing Partial Key System, ACM-SIGSAC *REVIEW* , Vol.6, No.3, pp.10-13.

Hellman M.E. (1979) : DES will be totally insecure within ten years, IEEE Spectrum, Vol.16, No. 7, pp.32-39.

Highland H.J. (1988) : The *Brain* Virus : Fact and Fantasy, Computer and Security, Vol.7, No.4, August, pp.367-370.

IEEE Transactions on Software Engineering (1987) : Special issue on computer security and privacy , February 1987, Vol. SE-13, No. 2.

Jackson K.M., Hruska J. (1988) : The PC Security Guide 1988/1989, Elsevier Advanced Technology Publications, 4 chap. and annexes.

King R.A. (1984) : Guide du PC-DOS, Sybex, pp.1-321.

Mahmood A., McCluskey E.J. (1988) : Concurrent Error Detection Using Watchdog Processors - A Survey, IEEE Trans Comp, Vol. TC-37, No. 2., pp.160-174.

March J.G., Olsen J.P. (1979) : 1.Organizational choice under Ambiguity, in Ambiguity and Choice in Organizations, Universitetsforlaget, Norvay, pp.10-23.

Morris R., Sloane N.J.A., Wyner A.D. (1977) : Assessment of the National Bureau of Standards proposed Federal Data Encryption Standard, Cryptologia, Vol.1, No.3, pp.281-291.

Penrose L.S. (1959) : Self-reproducing machines, *Scientific American*, June, pp.105-114.

Pendlebury D. (1989) : Life Sciences 100, 1987-1988, Surveying The Players, Molecular recognition was chief focus of the life sciences 100,*The Scientist, Vol.3, No.9,* May 1st., p.12 and p.14.

Thompson K. (1984) : Reflexions on Trusting Trust, Turing award lecture, Comm ACM,Vol.27, No.8, pp.761-763.

Watson James D. (Nobel Prize 1962 in Medicine and Physiology) **(1970)** : Molecular Biology of the Gene, 2nd.Ed., W.A. Benjamin, Inc., New-York, pp.1-662.