Computer, Viruses and Lawyers

by

J.W.F. Burnside

Delivered at a meeting of the Medico-Legal Society held on
4th April 1989 at the Royal Australian College of Surgeons.
The Chairman of the meeting was the President,
Mr. D. Graham Q.C.

INTRODUCTION

A virus, as the doctors know, is a nucleic acid core in a protein coat. On its own, it is virtually inert. To exist and reproduce it must invade a living cell, where it subverts the metabolic processes of the host.

Once inside a host cell, the virus uses the host as the mechanism of its own reproduction. This may be by the direct production of entire virus particles; but in some cases, specifically amongst viruses which live in bacteria, lysogenic reproduction is encountered. In lysogenic infections, the viral genome forms a stable association with the host's genetic material, and the two sets of genetic material divide and replicate together. Each new bacterium thus contains the viral genomes, but the infection is hidden until the viral genome triggers viral replication in which case new viruses are released from the now numerous host bacteria.

This much is no doubt commonplace to the doctors. For practical purposes it exhausts my knowledge of organic viruses. But the viruses in the title of this paper are not the organic viruses familiar to doctors. They are fragments of computer code which have characteristics startlingly analogous to organic viruses. The first computer virus was detected only about three years ago. Computer viruses are important because they pose an awesome threat to computer installations all around the world. In computing, viruses are being given the same anxious attention as is AIDS in medicine.

A typical computer virus was that found at Lehigh University in Pennsylvania in 1987. It was a piece of code occupying less than 200 bytes of stack space in the COMMAND.COM file, which executes every time DOS is booted. Since the virus occupied stack space, the file size was not increased. On execution it did a directory check on all disks on the system. If COMMAND.COM in its original form was found on any other disk, the altered file was copied onto the unaltered file and an internal counter was incremented. When the counter registered four, all files on the logged disk-drive would be erased.

COMPUTERS

Computing is a very recent development in human intellectual achievement. To understand an organic virus it is probably not necessary to refer to Vesalius and Paracelsus for background. In the case of computer viruses, a little history is justified.

A bit of history

Mankind has essayed a number of devices to facilitate the computation of numbers. The abacus dates at least from Roman times and has been in regular use in the eastern world ever since. Until the 17th Century A.D., the abacus was without any rivals, but the flowering of mathematics in 17th Century Europe produced the slide-rule, based on John Napier's discovery of logarithms; and it also saw the invention of the first true mechanical calculating machine. Blaise Pascal was one of the great figures of modern thought. He is remembered for expounding the mathematics of probability, discovering the fundamentals of hydraulics, hydrostatics and hydrodynamics, and for his writings on religious philosophy. But Pascal's father was a tax official at Rouen, and to help his father's work, Pascal devised and built a mechanical calculator which was the basis for his earliest fame.

Gottfried Wilhelm Leibniz was a philospher, mathematician, logician, lawyer and political adviser. In his early twenties he obtained a doctorate in law and was offered a professorship at the University of Nuremburg, which he declined. Independently of Newton, he devised integral and differential calculus. He worked as an engineer, he founded the science of geology, he perfected a binary numbering system and he created an improved version of Pascal's calculating machine, all this before his 30th birthday. Incidentally, because of his position as historian to the House of Brunswick, it fell to Leibniz to establish the genealogical right of George Louis to ascend to the English throne as the first Hanoverian King, George I, after the death of Queen Anne.

The nineteenth century

The next great landmarks in the history of computing occurred almost simultaneously but quite independently. The first, but perhaps the least important, was the work of Charles Babbage, who between 1822 and 1833 designed and attempted to build an elaborate calculating machine which he called the Difference Engine. Although supported by 17,000 pounds in Government grants, Babbage was unable to complete the Difference Engine. Undaunted, in 1834 he began work on the Analytical Engine which, in concept, was the first true programmable computer. His collaborator in the project was Augusta Ada Byron, Countess Lovelace; the only legitimate daughter of Lord Byron and a striking refutation of the idea that women cannot be good at mathematics. The Analytical Engine could be programmed by means of punched cards. This idea had been borrowed from Joseph-Marie Macquard, whose automatic looms had caused great industrial disturbances amongst silk weavers in 1806, five years before the Luddite movement began. However the Analytical Engine was never built: although its design was sound, its engineering demands exceeded the technology of the 19th Century. The design lay forgotten until Babbage's notebooks were discovered in 1937.

Although the design of the Analytical Engine was ignored, the proposed use of punched cards had great consequences. In the late 1880's, Hermann Hollerith proposed the use of punched cards of the same general sort as a means of storing and manipulating information for the 1890 US census. His tabulating machine, which incorporated the idea of the punched cards, was adopted for the census and was instantly successful. It dramatically reduced the time taken in compiling census results. Hollerith thereupon founded a company which prospered greatly and is now known around the world as IBM.

The 1830's saw two other important developments. In 1832, an American proposed that information could be transmitted along wires as electrical pulses. Ironically, electronics was not his field of endeavour. He was a painter, and is regarded now as one of America's finest 19th Century portrait painters. His interest in electricity was aroused when studying at Yale. His father was a Congregational minister, and evidently a serious person of good purpose, who disapproved of his son's dabbling with painting (he thought it frivolous) and also disapproved of his interest in the subject of electricity, presumably because it was still very new and had no obvious uses. Well, like many serious and well-intentioned fathers, Jebediah Morse got it all wrong, because Samuel Morse is now regarded as one of America's finest 19th Century portrait painters and his Morse code survives in use to the present day.

It was not only his father who thought Morse's idea of sending information along wires to be foolish. Morse's attempts to establish telegraph links between American cities were initially regarded with scepticism. However, by 1839 he had enlisted enough political support to construct a telegraph line from Baltimore to Washington. His first message sent by telegraph was 'What hath God wrought?'. No doubt his father would have approved the sentiment.

The first telegraph link was immediately successful. There followed a barrage of litigation, in which friends and collaborators asserted rights to Morse's invention. The litigation ended up in the American Supreme Court where, in 1854, Morse was successful. Europe and America took to the telegraph, and as Morse code spread across the globe, Samuel Morse became wealthy. The fame which had eluded him as a painter came to him as a philanthropist. I do not dwell to draw the moral from that.

Boolean algebra

Just as Morse was securing his legal rights and assuring his fortune, an English mathematician was independently laying the foundation stone on which modern digital computers are built. George Boole was the son of a shoemaker. He received some informal training in mathematics from his father, but until he was 17 his intellectual preference was for classical languages. However, to the great profit of mankind, Boole began to teach himself mathematics at the age of 17. He was soon recognised for his skilled and insightful writing on mathematics and in 1854, the year of Morse's victory in the US Supreme Court, Boole published a treatise entitled 'An investigation of the laws of thought, on which are founded the mathematical theories of logic and probabilities'.

The Encyclopaedia Britannica says of Boole:

> 'He did not regard logic as a branch of mathematics ... but he pointed out such a deep analogy between the symbols of algebra and those which can be made, in his opinion, to represent logical forms and syllogisms, that we can hardly help saying that logic is mathematics restricted to the two quantities 0 and 1'.

To anyone who has even a passing acquaintance with the subject, the significance of that passage to the operation of digital computers is striking, and the more so when it is noted that the quotation comes from the 9th Edition of the Britannica, published in 1898.

Boole's system of symbolic logic permitted the expression of logical propositions in equations which used only two quantities, 0 and 1, and operated in a manner analogous to orthodox algebra. His work was for the most part disregarded at the time, by mathematicians and philosophers alike.

Boole died when he was 49. He had five daughters. One married a mathematician; one became a mathematician; one became the mother of a mathematician; one became a professor of chemistry; and the youngest, Ethel Lilian, married a Polish scientist, Wilfrid Voynich, and became a successful novelist. One of her novels, The Gadfly, became so popular in Russia that three operas are based on it, and Shostakovitch wrote a film score based on it. The Countess Lovelace would have been proud.

It was not until the 1940's that it was recognised that Boole's ideas could be the mechanism of a technological revolution. The reason Boolean algebra is so important in modern computing is a function of the technology which modern computers employ.

Digital computers

There are two sorts of computers: analogue and digital. Almost all modern computers are digital computers. What this means, is that the infromation being handled by the computer is represented in numeric form, rather than by an electronic quantity analogous to the information. A fuel gauge in a car is an example of an analogue system: the position of the needle represents how full or empty the tank is. Most cars also have an analogue speedo: the needle moves around the dial in a way which corresponds with the speed of the vehicle. On the other hand, some cars have a digital speedo: it simply displays a number representing the vehicle's speed in kilometres per hour.

Binary systems

Boolean algebra is not only digital — it is based on a binary system. Our ordinary counting system is a decimal system, that is, it is on a base of ten. Each number position away from the decimal point represents a tenfold change of magnitude (tens, hundreds, thousands, etc.). In a binary system, each number position away from the decimal point represents a twofold change in magnitude (twos, fours, eights, sixteens, etc.). There is no magic in a decimal counting system. It just happens to be the number system which took root in most advanced civilisations. It is easy to suppose that there is something natural about the decimal system. However, the Mayans used a number system based on 20; various other South American Indian tribes used number systems based on three, four or five. The Babylonians used a system based on 60, and some Australian Aboriginal tribes use a numbering system based on two (that is to say, a binary system).

A reminder that a decimal system is arbitrary, even if effective, can be seen in the curious counting system we use to record time: 60 seconds per minute, 60 minutes per hour, 24 hours per day.

A binary system has only two digits: zero and one. Combinations of those digits can represent any number, just as combinations of the digits zero to nine can represent any number in a decimal system. The crucial point for the present purpose is that electronically, the binary digits can readily be represented by various phenomena having two states: on–off; high voltage–low voltage; positive–negative.

It is this fact which makes binary systems so suitable for electronic computers: the electrical and magnetic representation of two states is essentially simple. So, a binary number system enables digital computers to perform arithmetic operations, and Boolean algebra enables them to perform logical operations.

By the time the 20th Century dawned, the intellectual precursors of computers were in existence, but the technology was not.

It is an odd thing, but for all their startling capabilities, computers are fundamentally a large collection of switches, linked together and capable of operating at high speed. However, the utility of a digital computing device depends on the number of switches and the speed at which they operate.

At the beginning of the 20th Century, the only available switching devices were physical switches. The vacuum tube valve did not emerge until 1907. Because of the limits of the available technology, a useful digital computer was not feasible: this was the same fate which befell Babbage's Analytical Engine. Because the capabilities of digital computers were beyond the realm of imagination, there was no concerted effort to produce the technology which would make them possible.

Turing's machines

The final intellectual push which culminated in the development of the first electronic computer, came from a number of people, but perhaps the greatest, and certainly the quirkiest, of them was Alan Turing. He was born in 1912 and died just before his 42nd birthday. At school he excelled in mathematics, to the exclusion of almost everything else.

Turing had interested himself in Bertrand Russell's pursuit of a single and complete system which could encompass all valid principles of mathematical reasoning. Russell's hopes were dashed in 1931 when Kurt Goedel demonstrated that in any consistent axiomatic system there existed propositions which could not be decided within the bounds of that system. Turing turned his mind to the question of whether rules could be formulated which would enable the undecidable propositions in a system to be identified mechanically. If so, then the Goedelian oddities could be chopped off the edges of otherwise complete systems. His inquiry led him to consider the nature of logical machines. This became an exercise in itself.

He first described a theoretical machine which could solve specific problems by following defined rules, and then described a more general machine, the Universal machine, which could be so coded as to emulate the operation of any of the specific machines which he had described. Turing's Universal machine was a theoretical model only, but it did two things: it demonstrated logically that the problem identified by Goedel's theorem was embedded in the very fabric of all logical systems and was not capable of being mechanically excised; and it provided the conceptual model for a general purpose computer.

During the Second World War, Turing got the opportunity to apply in practice his theory of machines. Along with a small number of others he was installed at Bletchley Park, near Cambridge, to devise a system for breaking the German 'Enigma' code system. For the first time in history, the theory of numerical machines and the best available technology were pressed into service with great urgency. The process resulted in a great improvement of techniques and technology: the first Enigma message took two weeks to decipher. Eventually, the Bletchley Park team could decipher intercepted Enigma messages in minutes.

Electronic computers

The machine used by the team at Bletchley Park was electromechanical. The first electronic computer was ENIAC, completed in 1946. The basic electronic device which performed the switching operations in ENIAC was the vacuum tube valve. Although ENIAC represented a dramatic improvement over mechanical computation devices, it was big, slow and unreliable by today's standards. An electronic valve is the size of a small light globe, and generates almost as much heat.

Valves impose some serious practical limits on the construction of computers. Valves have a finite, and not very long, life. The failure of any one valve in the computer is likely to cause the entire computer to fail. Suppose the average life of a valve is 2,000 hours. If the computer has 20,000 valves, which is quite a modest number, then, in theory, after a certain period of operation one valve will fail about every six minutes.

In 1959 the second generation of computers emerged. In the second generation, the function of the valves was performed by transistors. A transistor is a device produced by treating a semi-conductor, typically germanium or silicon, with a specific dopant. Ordinarily, semi-conductors resist the passage of electricity. However, if specific impurities are introduced in tiny amounts, the electrical properties of the semi-conductor change. By manipulating the type and position of the impurities, the semi-conductor can be made to permit or resist the flow of electricity depending on the state of a control current applied at the region where the impurities are present. So, transistors do what valves do, but a typical transistor is about the size of a drawing pin; a typical valve is about the size of a light globe. This meant that the physical size of computers reduced significantly, and their heat output was also reduced.

In the late 1950's it was recognised that, in principle, a number of transistors (and other more simple electronic components) could be manufactured on a single piece of silicon. During the 1960's, techniques for creating numerous electronic components on a single piece of silicon were developed and improved. The rate and extent of development has been astonishing. In 1961, a chip the size of a thumb-nail contained the equivalent of five components. By 1971, it contained 2,000 components; by 1981 a chip the size of a drawing pin could hold nearly half a million components; by 1991, it is thought that the same chip will hold 10 million components.

The first valve computer, ENIAC, filled a large room. Its silicon chip equivalent would fit on a five cent piece.

The silicon chip

To fabricate a silicon chip, it is necessary to lay down areas of impurity, pathways of a conducting metal and pathways of insulation, within and on the surface of a piece of pure silicon. The technology required to achieve this is awesome.

The first step is to grow a crystal of silicon. The crystal is generally about five inches in diameter and about a foot long. Once the crystal has been grown, it is sliced into discs less than a millimetre thick. Each disc will eventually contain about 150 chips.

A number of masks are made. They delineate the areas on the chip where each of various substances will be deposited. They look a bit like very complicated road maps. These masks begin life on very large sheets of transparent plastic. They are then photo reduced to the size of the chip and are replicated so that hundreds of chips can be produced simultaneously on the same wafer.

The process of depositing foreign substances on the silicon wafer has many stages, so that what results is a series of more or less flat patterns laid one upon the other on the wafer of silicon. Vertical connections are made by allowing areas of one pattern to overlap the one next below it. Separation between adjacent layers is achieved by selectively depositing an insulating material.

The process of depositing each layer involves the following steps:

1. The surface of the wafer is oxidised;
2. The silicon wafer is coated with a light sensitive substance called photoresist;
3. The mask is laid over the wafer, and is exposed to ultraviolet light;
4. The wafer is washed in hydrofluoric acid. This dissolves those areas of photoresist which were not exposed to light, and etches away the silicon dioxide in areas no longer protected by the photoresist;
5. The areas of silicon which are exposed by the removal of the photoresist layer may then be treated with any of the various substances to be deposited on it.

These steps, with some variations, proceed layer after layer until as many as eight or ten layers of patterns have been built up. The wafer is then sliced up, to separate each chip from its neighbours.

The masking process is fundamental to the fabrication of silicon chips. You may get some appreciation of the fineness of the processes involved, if you realise that the factory that is presently limiting the further compression of components on a chip is the wavelength of light: if the patterns on the masks are made any finer, light with a shorter wavelength will be needed if it is to be able to pass through the clear areas of the masks without diffraction.

Consider this. Old fashioned electronic circuits used ordinary pieces of hook up wire to connect the various components to each other. If each of the conducting strips in a silicon chip were the size of ordinary hook up wire, the chip would be about the size of a football field.

Since 1960, the electronics of computers has become smaller, faster, cheaper and more powerful to an extent which is difficult to grasp. It is said that if aviation technology had developed at the same pace, a Jumbo jet would now be able to fly a million miles an hour, it would fly to the moon and back on one gallon of fuel, it would cost about $3.00, and would fit in your pocket!

The development of the silicon chip changed the face of computing. Most of the cost of a chip is in its design. The unit cost of manufacture is very low. Ultimately then, the selling price depends on volume. In 1954, IBM estimated the U.S.A. market for computers to be approximately fifty machines. There are now tens of millions of computers in service in the U.S.A. alone, and the market is expanding. The single reason for this dramatic expansion in the market is the fact that the silicon chip facilitates the manufacture of exceedingly complex electronic devices which are both small and, if produced in large numbers, inexpensive. Thus, the silicon chip made it both possible and economically desirable to expand the market for computing devices. A simple illustration of the market result can be seen in calculator-watches. The first of these was marketed in 1974 by Hewlett Packard. They sold for $1,000. Ten years later they sold for as little as $15.

For a computer to do anything at all, it must be programmed. The physical components of a computer are called hardware, and the programs and data it contains are called software. To illustrate the distinction — and to illustrate the profound importance of software — if a human being is hardware, then his software includes everything he has ever learned — how to walk, how to spell, how to take out tonsils and how to recognise a Rembrandt. Without software, the silicon chip is incapable of doing anything.

Of bits and bytes

A bit is a binary digit. It is thus the most basic unit which a computer can handle. There are only two binary digits: zero and one. The information potential of a single bit is correspondingly limited: it can represent only one of two states. To enable a computer to handle data in useful quantities, microprocessors are so designed that they can handle eight, sixteen or thirty-two bits in parallel. Functionally, these are viewed as one, two or four groups of eight bits each. Each group of eight bits is a byte. (In the quaint language of the computerist, a group of four bits is called a nibble).

Because a byte contains eight bits, the information carrying potential of a byte is much greater than that of a bit. As a bit may have either of 2 states, a byte may have any of 256 states (256 is 2 raised to the power of 8).

The information carrying potential of a group of bits can be ascertained by considering how many different states can be represented by them, in the following way:

1 bit can be 0 or 1 (2 states);
2 bits can be 0 0
0 1
1 0
1 1
(4 states, i.e. 2 squared);
3 bits can be 0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

(8 states, which is the same as 2 raised to the power of 3)
4 bits can represent 16 states (i.e. 2 to the 4th power) and so it goes.

Software and data have to be loaded into memory to enable the computer to operate on them. Various storage techniques have been used. One of the earliest was the punched card. However, the commonest storage media now in use are magnetic disks and magnetic tape. There are various types of disks. Floppy disks are flexible plastic disks coated with a magnetic oxide and contained in a protective envelope. They are inserted into a disk drive when required. Typically they store less than one megabyte of data. Hard disks use the same technology as floppy disks, but the disk is rigid and is much more securely protected from the environment. For these reasons, it is able to spin at a much greater rate and is manufactured to much closer engineering tolerances. Both of these circumstances mean that a fixed disk can be made to store much more information than a floppy disk, and can be read at a much greater speed. Hard disks typically hold anything from 10 megabytes to 300 megabytes of data.

300 megabytes is a very large amount of information. For all practical purposes, one byte represents a single character, for example a letter or a number or a punctuation mark. A megabyte is approximately a million of them. An average page of text is about 1500 bytes. This paper contains about 25,000 bytes, so that a 300 megabyte disk could hold about 12,000 papers of this length.

Hackers, time bombs and Trojan Horses

Vast amounts of human effort and ingenuity are involved in writing software. It is an engrossing task, filled with intellectual challenge. Not surprisingly, some people become truly addicted to it. But a curious breed of computer addicts has emerged. They are called hackers. The average hacker is gifted in his understanding of software, but derives entertainment from invading other people's computer systems uninvited.

Many computers are connected permanently to a telephone line. This enables them to communicate with other computers. Access to these computers is restricted by various security measures — all implemented in software. Hackers devise ways of getting past the security measures and, once inside the unwitting host computer, they browse through data files, or steal programs, or cause havoc to prove that they were there. It is in this shadowy silicon demi-monde that computer viruses exist.

In orthodox hacking attacks, there are many techniques with colourful names. For example, the salami technique, which involves slicing off fractions of a cent when interest is calculated on numerous account balances, and accumulating those fractions to a dummy account accessible to the hacker. Many thousands of dollars have been stolen this way.

The time bomb is a technique in which a program introduced by the hacker watches for a particular event to occur and then corrupts or deletes the data base.

The Trojan Horse is a hacking technique much favoured by the cognoscenti. It involves the insertion into a legitimate program of a set of instructions which will execute when the program is run. The existence of the added instructions is invisible to the user until their consequences are seen. The point about the Trojan Horse technique is that it is a method of introducing unauthorised program instructions into a computer — the nature of those instructions depends on the inclinations of the hacker. Many hackers are interested only in leaving their mark — by corrupting or deleting a data base, or by causing the system to go haywire.

VIRUSES

A virus is a specific form of Trojan Horse attack. The virus only exists as an unauthorised portion of code hidden inside an authorised program. What distinguishes viruses from orthodox Trojan Horses is that they are able to reproduce themselves and they watch for, and take, the opportunity to spread to other systems.

The virus discovered at Lehigh University in Pennyslvania was hidden in the COMMAND.COM file — a program which ran automatically every time the host computer was switched on. Each time the COMMAND.COM file executed, the virus would check to see whether the computer held any other copy of COMMAND.COM. If so, it would copy itself into the uninfected file, and would increase a counter with which it kept track of the number of other files it had infected. Once the counter reached four, the virus would delete every file on the disk. If that does not make you shudder, you are not a computer user. You will recall that a hard disk could hold approximately 12,000 papers of this length: that is a lot of work to lose. A different measure is this. A standard size hard disk on a small computer can hold approximately $85,000 work of standard commercial software. That is a lot of software to lose.

When the virus deletes the contents of a disk, it deletes itself in the process, so that it is very difficult to determine what happened. But the damage does not stop there. By the time the first disk has been deleted, there are three copies of the virus on other systems, each of which has been reproducing itself in the meantime.

A virus found in the computer system at the Hebrew University in Jerusalem was designed to reproduce itself much more quickly than the Lehigh virus: so quickly that, within half an hour of an infected machine being switched on, its disk space was filled to capacity with program files which had become bloated by the reproduction of the virus. The operating speed of the infected computers dropped to 20% of normal. The speed with which the virus spread was probably unintended, because it caused the virus to be detected before it could enter its next phase: it was programmed to erase every file on every infected computer at midnight on the 13th May, 1988 — the 40th anniversary of the end of the British mandate in Palestine and the creation of the State of Israel.

A virus found in the U.S.A. last year altered the cycle time of a video control board inside the host computer. The result was that the affected board would overheat and cause a fire inside the computer.

A virus which revealed itself in December 1987 spread throughout IBM's worldwide data network, but started in a European academic network. When the infected program ran, it caused a picture of a Christmas tree to appear on the host computer's screen. However, whilst that was happening, the program was reading the electronic mail directory on the host computer: the names and addresses of everyone with whom the owner of the machine had had contact by elecronic mail. The virus then copied itself to every computer on that list. Once it had entered those computers, it went through precisely the same routine. Within days,the virus had spread to tens of thousands of systems around the Western world.

LAWYERS

It is an interesting reflection on the mentality of the authors of the viruses so far detected, that they seem to be motivated more by a desire to attact attention than by the possibility of profit. It is an interesting reflection on the operation of the law that (with the exception of the fire bomb virus) none of the activities I have described falls within any orthodox criminal category. So far, very few governments have enacted laws making this sort of behaviour illegal. Several American states have. I am pleased to say that Victoria, alone amongst the Australian states, has effectively made hacking (whether for pleasure or profit) a criminal offence as of 1st June, 1988.

However, it remains to be seen how effectively hacking can be prosecuted. I have in mind the scene in Court at the start of the trial of a sophisticated hacker. The bar table is buried under huge piles of computer printout, containing a listing of the entire contents of an infected disk, all in hexadecimal notation or even worse, in binary code, and a similar dump from an uninfected disk. One expert witness holds reconstructed audit trails to demonstrate that there must have existed an unauthorised routine to record all passwords used on the system; another expert stands ready to explain the significance of an operating system call during the execution of code in the stack space of the EXE2BIN.COM file; the jury of unmarried mothers in moccasins and unwashed students in thongs sits expectantly as a young man in a horse hair wig and block gown rises earnestly to address a not-so-young man in a horse hair wig and red robes trimmed with ermine . . .