the M-commerce site using encryption and certificates, and it is probable that people will expect the repository provider to verify the trustworthiness of the M-commerce site on their behalf, by checking the validity of certificates, before releasing personal data. Fledgling examples of this kind of online service are only now arising — such as Microsoft Passport — and they are certain to play an increasingly important role in M-commerce.

The future

While the role of the handset in a PKI may be limited to transferring short requests, these requests will unlock highly valuable back-end systems. This, coupled with the fact that handsets are more easily lost or stolen than PCs makes it doubly important that the handset does not become the weak link in the PKI. Ideally the handset should verify the identity of the user using a biometric measure, such as a fingerprint scan, although this may be some years away. In the interim, passwords and PIN numbers must be used but are never transmitted across the network in a PKI based system. Handsets, by their very nature, have the potential to transform to the Personal Trusted Devices of the future.

While M-commerce presents many challenges to those implementing security, none are insurmountable. Co-operation among certification authorities, mobile operators, systems integrators and device manufacturers is needed to ensure that mobile security is implemented properly, and that different implementations will be able interact. Uniform standards need to be established and adhered to. Governments and regulators must create legislation, guidelines and practices that allow M-commerce to flourish.

Security can unlock the potential of M-commerce and M-business. It is an enabling rather than restrictive force that builds trusted relationships between businesses, their partners and consumers on the move.

The Computer Virus — From There to Here.

An Historical Perspective.

Berni Dwan

I was sitting at my desk in the Computer Centre of University College, Dublin, checking my E-mail. Suddenly, from the corner of my right eye, I detected an arm swooping down towards my keyboard, like an axe about to fell a tree. This was an arm with a mission. It removed my hand, which was just about to press 'receive' on an interesting looking E-mail involving a Christmas tree greeting. Of course, my better-informed colleague knew that this was the *Christma Exec* virus, and that I would not be doing myself, or anyone else, any favours by accepting and executing it. It was December 1987, and my very first introduction to the computer virus. My love affair with the subject has continued over the past thirteen years (although I do not claim to have a copy of The Little Black Book of Computer Viruses sitting on my bookshelf!), and like myself, the computer virus has evolved into a more complex and sophisticated entity.

The *Christma Exec* program spread to thousands of VM/CMS installations connected to EARN, BITNET and IBM's internal VNET, and would only spread if a recipient received and (albeit unintentionally) executed it. More strictly a worm than a virus, *Christma Exec* was a prophetic precursor to *Melissa* and *VBS/LoveLetter*, utilizing the increased use of networks and desktop connectivity to the Internet, and sending a copy of itself to all of the users' listed correspondents.

While probably the first newsworthy virus, it was by no means the first, and knowledge of the computer virus phenomenon had already been in existence for several years.

The term 'computer virus' was formally defined by Fred Cohen in 1983, while he performed academic experiments on a Digital Equipment Corporation VAX system. The first computer viruses were developed in the early 1980s, the first found in the wild was the Apple II virus, *Elk Cloner*, which was reported in 1981. The first IBM-PC virus appeared in 1986. This was the *Brain* virus, a boot sector virus that remained resident. In 1987, *Brain* was followed by *Alameda (Yale), Cascade, Jerusalem, Lehigh,* and *Miami (South African Friday the 13th)*. These viruses expanded the target executables to include COM and EXE files and *Cascade* was encrypted to deter disassembly and detection. Variable encryption appeared in 1989 with the *1260* virus. Stealth viruses, which employ various techniques to avoid detection, also first appeared in 1989, such as *Zero Bug, Dark Avenger and Frodo* (4096 or 4K). In 1990, self-modifying viruses, such as *Whale* were introduced. The year 1991 brought the *GP1* virus, which is 'network-sensitive' and attempts to steal Novell NetWare passwords.

While some of the most commonly detected viruses vary according to continent, *Stoned, Brain, Cascade* and members of the *Jerusalem* family, have spread widely and continue to appear. In fact, a disk scanning 'fest' in any large organization could still encounter at least one of these. This implies, according to Lawrence E. Bassham & W. Timothy Polk, that highly survivable viruses tend to be benign, replicate many times before activation, or are somewhat innovative — utilizing some technique never used before in a virus.

Vagaries of location aside, the virus payload also comes in different sizes. David Chess likens it to natural phenomena, “Small earth tremors are much more common than large earthquakes. Slight rises in the level of a river are much more common than floods.” So, accepting the fact that small things happen often and big things happen rarely, our management of the smaller events is consequently better than our management of the catastrophes, which, by their very nature and sometimes their unpredictability, are probably impossible to fully prepare for anyway. But accepting the infrequent occurrence of the great unknown, and with the knowledge that the probability of large computer virus incidents increases annually, there are less and less excuses to be caught totally unawares. Unlike the victims of the 1988 Internet Worm, which only God could have foreseen, we are now in a much better position to postulate, prophesy and prepare. Requiring no human intervention, the Internet Worm’s automatic spread was facilitated by some Unix bugs and features that simply allowed it to hop from machine to machine, replicating and executing in what seemed an unstoppable demonic cycle. This is the type of automatic spread that can also be enabled by mobile code. There are enough analysts and anti-virus resources out there to keep us aware of the threats and to advice us on practising 'safe computing'. There are enough uploads, downloads, updates, patches, free seminars, bulletins, news alerts, 'in the wild' and hoax lists to give even the most lethargic of us some level of protection.

If we can understand the changing technology we can understand and appreciate the emerging threats, and be aware of the products that are attempting to circumvent them. Imbedded macros in Microsoft Word and Excel documents (although the first macro virus, WM/Concept.A has being doing the rounds since 1995), mobile code in the form of Java, ActiveX and Visual Basic Script, and Trojan worms (with their many zipped and packed permutations) are the obvious threats that spring to mind. We should see new threats with every new development. Unlike the 1988 Internet Worm victims, we do have a chance to develop methods that can handle the majority of pandemic virus outbreaks, even if only to a modest extent. There is also a bigger and more established network of anti-virus experts than there was in 1988. Despite the march of time, it is still hard to believe that the main success of *Melissa* was that it needed human intervention to propagate and spread. Just as if the clock had been turned back to December 1987, there were still far too many unquestioning recipients of suspect E-mail. A simple user awareness campaign could have gone such a long way towards minimizing *Melissa.*

Bearing this in mind, David Chess poses some appropriate questions, “How will the continued growth of the Internet change the pattern of virus spread and similar threats. Is our current paradigm of virus containment sufficient for the future?” In 199,7 Chess foresaw integrated mail systems and mobile code having an impact on virus spread. There is no denying that this in fact has been the case, and while ease of use and functionality is paramount for the average Internet user, the technology used to achieve this is facilitating the active spread of network-aware viruses. “We need to be prepared for a world in which this sort of attack is common, by employing security systems that can deal with them routinely and automatically, reducing them to the level of ‘little tremors', so that we can free up our experts to deal with whatever ‘Big Quakes’ are coming down the wire behind them.”

How does this practically manifest itself? Well, in 1997, Chess was talking about real-time detection and removal, including analysis and interim action before the forwarding of data, and this has come to pass. A new development was reported by Robert Lemos on September 29th in *ZDNet* news whereby Symantec Corp. have developed a technology to stop viruses that use scripting — to infect, manipulate and destroy programs and data on computers — by blocking the use of certain commands commonly used by malicious code. This is a case of fighting the virus writers with their own tools, since scripting languages are also used by them to create viruses and Trojan Horses that run on a variety of platforms. Lemos quotes one consultant who sees a weakness in the Symantec defence, observing that each scripting language (e.g. JavaScript, Perl) will need unique software. When you compare the number of scripting languages with the number of computer viruses, this argument dies a swift death. As Lemos points out, “keeping up with the changes in scripting services could be far easier than keeping track of the more than 50 000 virus variants in the wild today.”

There is no doubting that the *Melissa* virus, while causing untold grief across the world, was also a valuable wake-up call, in that while updating their anti-virus software against *Melissa*, users automatically protected themselves from a host of other viruses that had not hit yet, some of which were actually more malevolent than *Melissa.* This was highlighted with the case of the *Chernobyl* strain *CIH*, which hit at least 540 000 computers in South Korea and Turkey in April 1999, with a payload of reformatting hard drives and in some instances, zapping a key chip on the computer motherboard. Discovered in May 1998 and believed to have originated in Taiwan, it is aimed at Windows 95/98 platforms. The more widespread variant strikes every 26th April, the anniversary of the Chernobyl nuclear disaster, a lesser strain striking on the 26th of every month while a third strikes on 26th June. It would be true to say that *CIH* wreaked a certain amount of havoc in both South Korea and Turkey, but interestingly, nowhere else. The Korean Supreme Court had to delay some rulings because evidence saved on computers was lost. The article speculates why *CIH* hit South Korea and Turkey in particular. One theory is that the virus piggy-backed on pirated software which is quite common in Asia. Another theory suggests that unlike the US, Asia was largely unaffected by the *Melissa* virus, thus they did not benefit from the *Melissa* victims wake up call.

Melissa and *VBS/LoveLetter* apart, the virus hall of fame had many infamous members throughout the 1990's, each responding or adapting in its own way to the changes in computing environments, machine types and operating systems.

Michelangelo got the award for hype and hysteria in 1992. Discovered in New Zealand in 1991, an infected system would have parts of its hard disk overwritten with random data if booted on the 6th of March of any year. With such a potentially damaging payload, the *Michelangelo* virus caught the attention of the media in the run up to 6th March 1992.

Disastrous predictions of doomsday proportions were being flouted on television and print media. When the fateful day dawned, there were no horsemen to be seen riding across the sky. All was quiet on the north, south, east and western front. White, Kephart and Chess even say that it was difficult to find a verified incidence of destruction of data anywhere.

But is this because the build up was mere hype, or because the people caught up in the hype ensured that they had the appropriate anti-virus software in place before the trigger date, and/or many IT managers prevented the trigger date appearing on their PCs? Like *Melissa* eight years later, it provided the, by now passé, wakeup call to the computer world.

Along with the hype, we also have the hoaxes, and they have indeed become an intrinsic part of the virus world. The hoax payload is wasting peoples' time and using up precious bandwidth.

Hoaxes depend upon ignorance and innocence, as well as peoples' enthusiasm for being good citizens and colleagues, by passing on the dire warnings to as many people as they can. Their misplaced good deeds are merely playing into the hands of those strange beings who instigate the hoaxes.

HoaxKill (www.hoaxkill.com) currently cites seven hoaxes on its active list and eight on its rare list. Some of these have been around for such a long time (*It Takes Guts to Say 'Jesus'* and *Penpal Greetings*) that it is hard to believe that there is anyone left in the universe that can be fooled by them.

www.macafee.com have fifty nine hoaxes listed on their site. It gets better with Vmyths.com (www.vmyths.com), who have one hundred and eight virus hoaxes listed.

Sarah Gordon, in her paper *Hoaxes and Hypes*, likens the prevalence of the computer hoax to peoples' continued belief in corn circles, even though in 1991, two hoaxers admitted to creating them over a fifteen year period.

She also cites the hoax carried out in Australia in 1988 by Jose Luis Alvarez, who claimed to be a channel for the spirit Carlos. Again, when a Sixty Minutes programme proved the feat to be a hoax, Gordon points out that prior to this, the Australian media had made no serious efforts to check the accuracy of the story, even though there were some glaring inconsistencies.

To this day, people still believe in corn circles and Carlos. In fact, the next time you receive a virus hoax warning from a friend or colleague, you might include in your response your fears concerning the spread of corn circles, and await their reply! Among her reasons cited for buying into computer virus or other hoaxes are: trust in authority, excitement, lack of appropriate scientific skepticism, sense of importance or belonging, and finally, furthering our own goals and self interests.

"the next time you receive a virus hoax warning from a friend or colleague, you might include in your response your fears concerning the spread of corn circles, and await their reply!"

There are two important terms used in 'virus speak' and they need to be understood, especially if we are to put the figures and statistics pertaining to computer viruses into a realistic perspective. If a newspaper article tells us that there are 50 000 computer viruses in existence, we may first be shocked at the figure, and secondly, we may ask ourselves, how come my computer is not infected every other day if there is that amount of malicious code working its way around cyberspace?

The terms I refer to are, 'In the Wild' and 'Zoo', and you will perhaps be less familiar with the second.

The phrase, 'In the Wild', was first coined by David Chess in 1990/91 to describe real-world virus incidents or those which pose an active threat to a real world PC.

'Zoo' viruses, while obviously existing, are not an active threat. When considering the effectiveness of anti-virus software, it is important to be aware of the meaning of these terms, because it is the product's ability to counteract the real world threat that you are interested in.

If we did not have a de facto standard like 'In the Wild', simply comparing products by detection rates would be meaningless. As perhaps the product with the higher detection rate might include many 'Zoo' viruses, which are not a threat to us, and the product with the lower detection rate might include most of the 'In the Wild' viruses that we should be concerned about.

According to Sarah Gordon, the *WildList* represents the most organized effort to catalogue those viruses which are spreading, although it may not contain all 'In the Wild' viruses, and should more appropriately considered as a subset of the 'In the Wild' viruses, albeit, a core set that every anti-virus product must be able to detect.

In 2001 the computer virus will be twenty years old. We are now concerned with stealth, polymorphic and armoured viruses.

Dr. Fred Cohen's definitive definition of the computer virus, "Any software which can modify other programs to include a (possibly evolved) version of itself," still stands, but the complexity which has ensued as a result of the constant battle of wits between the virus writers and the anti-virus software developers may require an updated definition.

Medium and low risk viruses simply replicate or deliver an innocuous payload. High risk viruses, those that corrupt, delete, reformat, crash or flood come in several guises.

The three broad categories, system (boot record), file and multi-partite infectors still remain, but they increasingly utilize the stealth and polymorphic approaches, rendering them far more difficult to trap and kill.

So, how has the stealth and polymorphic medley made life more difficult for the anti-virus software industry? Polymorphic viruses have been crafted to vary themselves when infecting new disks or machines. One of their tricks is to imbed a do-nothing instruction in their own source code to prevent a virus scanner from looking for strings of code which match known viruses.

The more sophisticated polymorphic viruses actually encrypt virus code with randomly generated numbers, and decrypt the virus each time it is executed. This negates the traditional signature matching approach of anti-virus software and requires specially constructed search engines, which can identify encryption schemes. Programs known as mutation engines allow almost any virus to be made into a polymorphic virus, the *MtE* mutation engine being the best known.

Stealth, on the other hand, is a technique employed by viruses to conceal themselves, rather than a virus type itself. Often, these programs monitor disk accesses, and if a request is made to read the boot record, the virus redirects the disk read to the original boot record, stored by the virus, negating any attempts to spot an altered boot record. The anti-virus workaround here is that most of these viruses can be spotted by checking to see if there are any of these illegal 'hooks' into the interrupt used for disk access.

With the discovery of the first Palm OS virus this September, one can see the amphitheatre widening.

The battle of wits continues. Simple replicating viruses were caught by signature matching. Encrypted viruses that hid the fixed signature were caught by virus scanners searching for tell tale sequences of bytes that identified a specific decryption routine.

In the case of that formidable adversary, the polymorphic virus, the response of the anti-virus writers was to develop generic decryption techniques that would trick it into decrypting and revealing itself.

But it is the virus authors who lead the way, as the anti-virus industry treads the Ferris wheel of reacting to their malicious code. They will continue to challenge the anti-virus industry, although the industry is looking forward itself, and is getting better at predicting new threats.

Carey Nachenberg predicts several even more complex permutations of the polymorphic virus. This is a constant challenge for the anti-virus industry, but surely they have more technical resources and man power than the virus writers?

It is also the case of many against a few, but hopefully this will always be enough to keep the birth rate of viruses down and the death rate up, thus avoiding too many future epidemics.

Notes

[1] *hreat Assessment of Malicious Code and Human Threats*, National Institute of Standards and Technology Computer Security Division, 1994

[2] Only spread on 5.25 diskettes. Introduction of 3.5 diskettes in the late 1980's saw its demise.

[3] Improved technology helped to eliminate this, its conduit being boot diskettes, which were increasingly replaced by hard disks from the mid 1980's.

[4] *Threat Assessment of Malicious Code and Human Threats*, National Institute of Standards and Technology Computer Security Division, 1994

[5] *The Future of Viruses on the Internet* — originally presented at the *Virus Bulletin International Conference*, San Francisco, October, 1997

[6] *The Future of Viruses on the Internet* – originally presented at the *Virus Bulletin International Conference*, San Francisco, October, 1997

[7] Dan Schrader, Trend Micro Inc.

[8] Robert Lemos, *ZDNet News*, April 28, 1999

[9] Steve R White, Jeffrey O Kephart, David M Chess. *Computer Viruses: A Global Perspective*, 1995

[10] Sarah Gordon, *Hoaxes and Hypes*, 1997

[11] Sarah Gordon, *What is Wild?* IBM TJ Watson Research Centre

[12] Sarah Gordon, *What is Wild?* IBM TJ Watson Research Centre

[13] These attach themselves to the system files stored on a disk. Any formatted disk can be infected. Examples include Stealth viruses, Stoned and AntiCMOS

[14] Attach themselves to COM or EXE files. When infected files run, the virus code executes first and infects other executables or makes itself resident before continuing with the actual program. This class includes the Hitchcock, Shake, and Dark Avenger viruses.

[15] These infectors have been programmed to be able to infect both files and boot records. Examples include the BootEXE and EXEBug viruses.

[16] Created by *Dark Avenger*

[17] Mark Hazen, *Infection Detection, Virus-Proofing Your LAN*, Office of Information and Instruction technology, 1995.

[18] PalmOS/Phage.963

[19] *Understanding and Monitoring Polymorphic Viruses*, Carey Nachenberg, Symantec Enterprise Papers, Volume XXX

[20] *Understanding and Monitoring Polymorphic Viruses*, Carey Nachenberg, Symantec Enterprise Papers, Volume XXX

[21] Explained in Steve R White, Jeffrey O Kephart, David M Chess. *Computer Viruses: A Global Perspective*, 1995