A

Seminar report

On

Computer Viruses

Submitted in partial fulfillment of the requirement for the award of degree
Of CSE

SUBMITTED TO:

www.studymafia.org

SUBMITTED BY:

www.studymafia.org

Preface

I have made this report file on the topic **Computer Viruses**, I have tried my best to elucidate all the relevant detail to the topic to be included in the report. While in the beginning I have tried to give a general view about this topic.

My efforts and wholehearted co-corporation of each and everyone has ended on a successful note. I express my sincere gratitude to ……..who assisting me throughout the preparation of this topic. I thank him for providing me the reinforcement, confidence and most importantly the track for the topic whenever I needed it.

Acknowledgement

I would like to thank respected Mr…….. and Mr. ……..for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a seminar report. It helped me a lot to realize of what we study for.

Secondly, I would like to thank my parents who patiently helped me as i went through my work and helped to modify and eliminate some of the irrelevant or un-necessary stuffs.

Thirdly, I would like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Next, I would thank Microsoft for developing such a wonderful tool like MS Word. It helped my work a lot to remain error-free.

Last but clearly not the least, I would thank The Almighty for giving me strength to complete my report on time.

Introduction

Computer viruses are called viruses because they share some of the traits of biological viruses. A computer virus passes from computer to computer like a biological virus passes from person to person.

There are similarities at a deeper level, as well. A biological virus is not a living thing. A virus is a fragment of DNA inside a protective jacket. Unlike a cell, a virus has no way to do anything or to reproduce by itself -- it is not alive. Instead, a biological virus must inject its DNA into a cell. The viral DNA then uses the cell's existing machinery to reproduce itself. In some cases, the cell fills with new viral particles until it bursts, releasing the virus. In other cases, the new virus particles bud off the cell one at a time, and the cell remains alive.

A computer virus shares some of these traits. A computer virus must **piggyback** on top of some other program or document in order to get executed. Once it is running, it is then able to infect other programs or documents. Obviously, the analogy between computer and biological viruses stretches things a bit, but there are enough similarities that the name sticks.

Types Of Viruses

When you listen to the news, you hear about many different forms of electronic infection. The most common are:

- **Viruses** - A virus is a small piece of software that piggybacks on real programs. For example, a virus might attach itself to a program such as a spreadsheet program. Each time the spreadsheet program runs, the virus runs, too, and it has the chance to reproduce (by attaching to other programs) or wreak havoc.

- **E-mail viruses** - An e-mail virus moves around in e-mail messages, and usually replicates itself by automatically mailing itself to dozens of people in the victim's e-mail address book.

The latest thing in the world of computer viruses is the **e-mail virus**, and the Melissa virus in March 1999 was spectacular. Melissa spread in Microsoft Word documents sent via e-mail, and it worked like this:

Someone created the virus as a Word document uploaded to an Internet newsgroup. Anyone who downloaded the document and opened it would trigger the virus. The virus would then send the document (and therefore itself) in an e-mail message to the first 50 people in the person's address book. The e-mail message contained a friendly note that included the person's name, so the recipient would open the document thinking it was harmless. The virus would then create 50 new messages from the recipient's machine. As a result, the Melissa virus was the fastest-spreading virus ever seen! As mentioned earlier, it forced a number of large companies to shut down their e-mail systems.

The ILOVEYOU virus, which appeared on May 4, 2000, was even simpler. It contained a piece of code as an **attachment**. People who **double clicked** on the attachment allowed the code to execute. The code sent copies of itself to everyone in the victim's address book and then started corrupting files on the victim's machine. This is as simple as a virus can get. It is really more of a Trojan horse distributed by e-mail than it is a virus.

The Melissa virus took advantage of the programming language built into Microsoft Word called **VBA**, or Visual Basic for Applications. It is a complete programming language and it can be programmed to do things like modify files and send e-mail messages. It also has a useful but dangerous **auto-execute** feature. A programmer can insert a program into a document that runs instantly whenever the document is opened. This is how the Melissa virus was programmed. Anyone who opened a document

infected with Melissa would immediately activate the virus. It would send the 50 e-mails, and then infect a central file called NORMAL.DOT so that any file saved later would also contain the virus! It created a huge mess.

Microsoft applications have a feature called **Macro Virus Protection** built into them to prevent this sort of thing. With Macro Virus Protection turned on (the default option is ON), the auto-execute feature is disabled. So when a document tries to auto-execute viral code, a dialog pops up warning the user. Unfortunately, many people don't know what macros or macro viruses are, and when they see the dialog they ignore it, so the virus runs anyway. Many other people turn off the protection mechanism. So the Melissa virus spread despite the safeguards in place to prevent it.

In the case of the ILOVEYOU virus, the whole thing was human-powered. If a person double-clicked on the program that came as an attachment, then the program ran and did its thing. What fueled this virus was the human willingness to double-click on the executable.

- **Worms** - A worm is a small piece of software that uses computer networks and security holes to replicate itself. A copy of the worm scans the network for another machine that has a specific security hole. It copies itself to the new machine using the security hole, and then starts replicating from there, as well.

A worm is a **computer program** that has the ability to copy itself from machine to machine. Worms normally move around and infect other machines through computer networks. Using a network, a worm can expand from a single copy incredibly quickly. For example, the **Code Red** worm replicated itself over 250,000 times in approximately nine hours on July 19, 2001. A worm usually exploits some sort of **security hole** in a piece of software or the operating system. For example, the Slammer worm (which caused mayhem in January 2003) exploited a hole in Microsoft's SQL server. This article offers a fascinating look inside Slammer's tiny (376 byte) program.

Worms use up computer time and network bandwidth when they are replicating, and they often have some sort of evil intent.

- A **worm** called **Code Red** made huge headlines in 2001. Experts predicted that this worm could clog the Internet so effectively that things would completely grind to a halt.

The Code Red worm slowed down Internet traffic when it began to replicate itself, but not nearly as badly as predicted. Each copy of the worm scanned the Internet for Windows NT or Windows 2000 servers that do not have the Microsoft security patch

installed. Each time it found an unsecured server, the worm copied itself to that server. The new copy then scanned for other servers to infect. Depending on the number of unsecured servers, a worm could conceivably create hundreds of thousands of copies.

The Code Red worm was designed to do three things:

- Replicate itself for the first 20 days of each month
- Replace Web pages on infected servers with a page that declares "Hacked by Chinese"
- Launch a concerted attack on the White House Web server in an attempt to overwhelm it

The most common version of Code Red is a variation, typically referred to as a **mutated strain**, of the original **Ida Code Red** that replicated itself on July 19, 2001. According to the National Infrastructure Protection Center:

The Ida Code Red Worm, which was first reported by eEye Digital Security, is taking advantage of known vulnerabilities in the Microsoft IIS Internet Server Application Program Interface (ISAPI) service. Un-patched systems are susceptible to a "buffer overflow" in the Idq.dll, which permits the attacker to run embedded code on the affected system. This memory resident worm, once active on a system, first attempts to spread itself by creating a sequence of random IP addresses to infect unprotected web servers. Each worm thread will then inspect the infected computer's time clock. The NIPC has determined that the trigger time for the DOS execution of the Ida Code Red Worm is at 0:00 hours, GMT on July 20, 2001. This is 8:00 PM, EST.

- **Trojan horses** - A Trojan horse is simply a computer program. The program claims to do one thing (it may claim to be a game) but instead does damage when you run it (it may erase your hard disk). Trojan horses have no way to replicate automatically.

Examples: NetBus and BackOrifice, Subseven

- **Boot Sector Viruses** - As virus creators got more sophisticated, they learned new tricks. One important trick was the ability to load viruses into memory so they could keep running in the background as long as the computer remained on. This gave viruses a much more effective way to replicate themselves. Another trick was the ability to infect the **boot sector** on floppy disks and hard disks. The boot sector is a small program that is the first part of the operating system that the computer loads. The boot sector contains a tiny program that tells the computer how to load the rest of the operating system. By putting its code in the boot sector, a virus can **guarantee it gets**

executed. It can load itself into memory immediately, and it is able to run whenever the computer is on. Boot sector viruses can infect the boot sector of any floppy disk inserted in the machine, and on college campuses where lots of people share machines they spread like wildfire.

In general, both executable and boot sector viruses are not very threatening any more. The first reason for the decline has been the huge size of today's programs. Nearly every program you buy today comes on a compact disc. Compact discs cannot be modified, and that makes viral infection of a CD impossible. The programs are so big that the only easy way to move them around is to buy the CD. People certainly can't carry applications around on a floppy disk like they did in the 1980s, when floppies full of programs were traded like baseball cards. Boot sector viruses have also declined because operating systems now protect the boot sector.

Both boot sector viruses and executable viruses are still possible, but they are a lot harder now and they don't spread nearly as quickly as they once could. Call it "shrinking habitat," if you want to use a biological analogy. The environment of floppy disks, small programs and weak operating systems made these viruses possible in the 1980s, but huge executables, unchangeable CDs and better operating system safeguards have largely eliminated that environmental niche.

Examples: Form, Disk Killer, and Michelangelo

- **Program viruses** - These infect executable program files, such as those with extensions like .BIN, .COM, .EXE, .OVL, .DRV (driver) and .SYS (device driver). These programs are loaded in memory during execution, taking the virus with them. The virus becomes active in memory, making copies of it and infecting files on disk.

Examples: Sunday, Cascade

- **Multipartite viruses** - A hybrid of Boot and Program viruses. They infect program files and when the infected program is executed, these viruses infect the boot record. When you boot the computer next time the virus from the boot record loads in memory and then starts infecting other program files on disk.

Examples: Invader, Flip, and Tequila

- **Stealth viruses** - These viruses use certain techniques to avoid detection. They may either redirect the disk head to read another sector instead of the one in which they reside or they may alter the reading of the infected file's size shown in the directory listing. For instance, the Whale virus adds 9216 bytes to an infected file; then the virus subtracts the same number of bytes (9216) from the size given in the directory.

Examples: Frodo, Joshi, Whale

- **Polymorphic viruses** - A virus that can encrypt its code in different ways so that it appears differently in each infection. These viruses are more difficult to detect.

Examples: Involuntary, Stimulate, Cascade, Phoenix, Evil, Proud, Virus 101

- **Macro Viruses** - A macro virus is a new type of computer virus that infects the macros within a document or template. When you open a word processing or spreadsheet document, the macro virus is activated and it infects the Normal template (Normal.dot)-a general purpose file that stores default document formatting settings. Every document you open refers to the Normal template, and hence gets infected with the macro virus. Since this virus attaches itself to documents, the infection can spread if such documents are opened on another computer.

Examples: DMV, Nuclear, Word Concept.

- **Active X** - ActiveX and Java controls will soon be the scourge of computing. Most people do not know how to control there web browser to enable or disable the various functions like playing sound or video and so, by default, leave a nice big hole in the security by allowing applets free run into there machine. There has been a lot of commotion behind this and with the amount of power that JAVA imparts, things from the security angle seem a bit gloom.

Origin Of Viruses

People create viruses. A person has to write the code, test it to make sure it spreads properly and then release the virus. A person also designs the virus's attack phase, whether it's a silly message or destruction of a hard disk. So why do people do it?

There are at least three reasons. The first is the same psychology that drives vandals and arsonists. Why would someone want to bust the window on someone else's car, or spray-paint signs on buildings or burn down a beautiful forest? For some people that seems to be a thrill. If that sort of person happens to know computer programming, then he or she may funnel energy into the creation of destructive viruses.

The second reason has to do with the thrill of watching things blow up. Many people have a fascination with things like explosions and car wrecks. When you were growing up, there was probably a kid in your neighborhood who learned how to make gunpowder and then built bigger and bigger bombs until he either got bored or did some serious damage to himself. Creating a virus that a spread quickly is a little likes that -- it creates a bomb inside a computer, and the more computers that get infected the more "fun" the explosion.

The third reason probably involves bragging rights, or the thrill of doing it. Sort of like Mount Everest. The mountain is there, so someone is compelled to climb it. If you are a certain type of programmer and you see a security hole that could be exploited, you might simply be compelled to exploit the hole yourself before someone else beats you to it. "Sure, I could TELL someone about the hole. But wouldn't it be better to SHOW them the hole???" That sort of logic leads to many viruses.

Of course, most virus creators seem to miss the point that they cause real damage to real people with their creations. Destroying everything on a person's hard disk is real damage. Forcing the people inside a large company to waste thousands of hours cleaning up after a virus is real damage. Even a silly message is real damage because a person then has to waste time getting rid of it. For this reason, the legal system is getting much harsher in punishing the people who create viruses.

1981 - The First Virus In The Wild

As described in Robert Slade's history, the first virus in the wild actually predated the experimental work that defined current-day viruses. It was spread on Apple II floppy disks (which contained the operating system) and reputed to have spread from Texas A&M. [*Side note*: Thanks to a pointer from anti-virus pioneer Fridrik Skulason we know the virus was named Elk Cloner and displayed a little rhyme on the screen:

It will get on all your disks

It will infiltrate your chips

Yes it's Cloner!

It will stick to you like glue

It will modify ram too

Send in the Cloner!

1983 - The First Documented Experimental Virus

Fred Cohen's seminal paper *Computer Viruses - Theory and Experiments* from 1984 defines a computer virus and describes the experiments he and others performed to prove that the concept of a computer virus was viable. From the paper...

On November 3, 1983, the first virus was conceived of as an experiment to be presented at a weekly seminar on computer security. The concept was first introduced in this seminar by the author, and the name 'virus' was thought of by Len Adleman. After 8 hours of expert work on a heavily loaded VAX 11/750 system running Unix, the first virus was completed and ready for demonstration. Within a week, permission was obtained to perform experiments, and 5 experiments were performed. On November 10, the virus was demonstrated to the security seminar.

1986 - Brain, PC-Write Trojan, & Virdem

The common story is that two brothers from Pakistan analyzed the boot sector of a floppy disk and developed a method of infecting it with a virus dubbed "Brain" (the origin is generally accepted but not absolutely). Because it spread widely on the popular MS-DOS PC system this is typically called the first computer virus; even though Cohen's experiments and the Apple II virus predated it. That same year the first PC-based Trojan was released in the form of the popular shareware program *PC-Write*. Some reports say Virdem was also found this year; it is often called the first file virus.

1987 - File Infectors, Lehigh, & Christmas Worm

The first file viruses started to appear. Most concentrated on COM files; COMMAND.COM in particular. The first of these to infect COMMAND.COM is typically reported to be the Lehigh virus. At this time other work was done to create the

first EXE infector: Suriv-02 (Suriv = Virus backward). (This virus evolved into the Jerusalem virus.) A fast-spreading (500,000 replications per hour) worm hit IBM mainframes during this year: the IBM Christmas Worm.

1988 - MacMag, Scores, & Internet Worm

MacMag, a Hypercard stack virus on the Macintosh is generally considered the first Macintosh virus and the Scores virus was the source of the first major Macintosh outbreak. The Internet Worm (Robert Morris' creation) causes the first Internet crisis and shut down many computers. CERT is created to respond to such attacks.

1989 - AIDS Trojan

This Trojan is famous for holding data hostage. The Trojan was sent out under the guise of an AIDS information program. When run it encrypted the user's hard drive and demanded payment for the decryption key.

1990 - VX BBS & Little Black Book (AT&T Attack)

The first virus exchange (VX) BBS went online in Bulgaria. Here virus authors could trade code and exchange ideas. Also, in 1990, Mark Ludwig's book on virus writing (*The Little Black Book of Computer Viruses*) was published. While there is no proof, hackers are suspected of taking down the AT&T long-distance switching system.

1991 - Tequila

Tequila was the first polymorphic virus; it came out of Switzerland and changed itself in an attempt to avoid detection.

1992 - Michelangelo, DAME, & VCL

Michelangelo was the first media darling. A wordwide alert went out with claims of massive damage predicted. Actually, little happened. The same year the Dark Avenger Mutation Engine (DAME) became the first toolkit that could be used to turn any virus into a polymorphic virus. Also that year the Virus Creation Laboratory (VCL) became the first actual virus creation kit. It had pull-down menus and selectable payloads.

1995 - Year of the Hacker

Hackers attacked Griffith Air Force Base, the Korean Atomic Research Institute, NASA, Goddard Space Flight Center, and the Jet Propulsion Laboratory. GE, IBM, Pipeline and other companies were all hit by the "Internet Liberation Front" on Thanksgiving.

1995 - Concept

The first macro virus to attack Word, Concept, is developed.

1996 - Boza, Laroux, & Staog

Boza is the first virus designed specifically for Windows 95 files. Laroux is the first Excel macro virus. And, Staog is the first Linux virus (written by the same group that wrote Boza).

1998 - Strange Brew & Back Orifice

Strange Brew is the first Java virus. Back Orifice is the first Trojan designed to be a remote administration tool that allows others to take over a remote computer via the Internet. Access macro viruses start to appear.

1999 - Melissa, Corner, Tristate, & Bubbleboy

Melissa is the first combination Word macro virus and worm to use the Outlook and Outlook Express address book to send itself to others via E-mail. It arrived in March. Corner is the first virus to infect MS Project files. Tristate is the first multi-program macro virus; it infects Word, Excel, and PowerPoint files. Bubbleboy is the first worm that would activate when a user simply opened and E-mail message in Microsoft Outlook (or previewed the message in Outlook Express). No attachment necessary. Bubbleboy was the proof of concept; Kak spread widely using this technique.

2000 - DDoS, Love Letter, Timofonica, Liberty (Palm), Streams, & Pirus

The first major distributed denial of service attacks shut down major sites such as Yahoo!, Amazon.com, and others. In May the Love Letter worm became the fastest-spreading worm (to that time); shutting down E-mail systems around the world. June 2000 saw the first attack against a telephone system. The Visual Basic Script worm Timofonica tries to send messages to Internet-enabled phones in the Spanish telephone network (later in 2000 another Trojan attacked the Japanese emergency phone system). August 2000 saw the first Trojan developed for the Palm PDA. Called Liberty and developed by Aaron Ardiri the co-developer of the Palm Game Boy emulator Liberty, the Trojan was developed as an uninstall program and was distributed to a few people to help foil those who would steal the actual software. When it was accidentally released to the wider public Ardiri helped contain its spread. Streams became the first proof of concept NTFS Alternate Data Stream (ADS) virus in early September. As a proof of concept, Streams has not circulated in the wild (as of this writing) but as in all such cases a circulating virus based on the model is expected. Pirus is another proof of concept for malware written in the PHP scripting language. It attempts to add itself to HTML or PHP files. Pirus was discovered 9 Nov 2000.

2001 - Gnuman, Winux Windows/Linux Virus, LogoLogic-A Worm, AplS/Simpsons Worm, PeachyPDF-A, Nimda

Gnuman (Mandragore) showed up the end of February. This worm cloaked itself from the Gnutella file-sharing system (the first to specifically attack a peer-to-peer communications system) and pretended to be an MP3 file to download. In March a proof of concept virus designed to infect both Windows and Linux (and cross between them) was released. Winux (or Lindose depending on who you talk to) is buggy and reported to have come from the Czech Republic. On 9 April a proof of concept Logo Worm was released which attacked the Logotron SuperLogo language. The LogoLogic-A worm spreads via MIRC chat and E-mail. May saw the first AppleScript worm. It uses Outlook Express or Entourage on the Macintosh to spread via E-mail to address book entries. Early August, the PeachyPDF-A worm became the first to spread using Adobe's PDF software. Only the full version, not the free PDF reader, was capable of spreading the worm so it did not go far. September, the Nimda worm demonstrated significant flexibility in its ability to spread and used several firsts. While not new in concept, a couple of worms created a fair amount of havoc during the year: Sircam (July), CodeRed (July & August), and BadTrans (November & December).

2002 - LFM-926, Donut, Sharp-A, SQLSpider, Benjamin, Perrun, Scalper

Early in January LFM-926 showed up as the first virus to infect Shockwave Flash (.SWF) files. It was named for the message it displays while it's infecting: "Loading.Flash.Movie...". It drops a Debug script that produces a .COM file which infects other .SWF files. Also in early January Donut showed up as the first worm directed at .NET services. In March, the first native .NET worm written in C#, Sharp-A was announced. Sharp-A was also unique in that it was one of the few malware programs reportedly written by a woman. Late May the Javascript worm SQLSpider was released. It was unique in that it attacked installations running Microsoft SQL Server (and programs that use SQL Server technology). Also in late May the Benjamin appeared. Benjamin is unique in that it uses the KaZaa peer-to-peer network to spread. Mid-June the press went wild over the proof-of-concept Perrun virus because a portion of the virus attached itself to JPEG image files. Despite the hype, JPEG files are still safe as you must have a stripper program running on your system in order to strip the virus file off the image file. On 28 June the Scalper worm was discovered attacking FreeBSD/Apache Web servers. The worm is designed to set up a flood net (stable of zombies which could be used to overwhelm one or more systems).

2003 - Slammer, Sobig, Lovgate, Fizzer, Blaster/Welchia/Mimail

Sobig, a worm that carried its own SMTP mail program and used Windows network shares to spread started the year. Sobig variants continued to multiply throughout the year. Slammer, exploiting vulnerabilities in Microsoft's SQL 2000 servers, hit Super Bowl weekend. Its spreading technique worked so well that for some period of time all of South Korea was effectively eliminated from the Internet (obscured). It received significant media coverage. The unique entry that February saw was Lovgate. This was unique as it was a combination of a Trojan and a worm; two pieces of malware that generally don't get combined. Starting in early May Fizzer spread via usual E-mail methods but also used the KaZaa peer-to-peer network to spread. While generally not unique types, August is (in)famous for a combination of Sobig.F, Blaster (also known as Lovsan and MSBlast), Welchia (or Nachi), and Mimail; all spreading rapidly through a security vulnerability in a Windows Distributed Component Object Model (DCOM) Remote Procedure Call (RPC) interface. 2003 also saw what appeared to be a use of worm-like techniques used in the spreading of spam. Sobig dropped a component that could later be used by spammers to send mail through infected machines. The social engineering techniques used by virus/worm writers improved dramatically as well. Some of the malware this year was accompanied by very realistic graphics and links in an attempt to make you think the mail actually came from the likes of Microsoft or Paypal.

2004 - Trojan.Xombe, Randex, Bizex, Witty, MP3Concept, Sasser, Mac OS X, W64.Rugrat.3344, Symb/Cabir-A, JS/Scob-A, WCE/Duts-A, W32/Amus-A, JPEG Weakness

Year 2004 started where 2003 left off with social engineering taking the lead in propagation techniques. Trojan.Xombe was sent out to a wide audience. It posed as a message from Microsoft Windows Update asking you to run the attached revision to XP Service Pack 1. (This, and like messages that "phish" for personal information, are expected to take a lead role in 2004 -- and, yes, phish is the correct term for a message designed to "fish" for personal information; the technique is called phishing.) In February it was demonstrated that virus writers were starting to ply their craft for money. A German magazine managed to buy a list of infected IP addresses from a distributor of the virus Randex. These IP addresses were for sale to spammers who could use the infected machines as mail zombies. The end of February saw Bizex go after ICQ users through an HTML link that downloaded an infected SCM (Sound Compressed Sound Scheme) file. The weekend of 20/21 March introduced Witty, the first worm to attack security software directly (some Internet Security Systems' RealSecure, Proventia and BlackICE versions). The worm was malicious in that it

erased portions of the hard drive while sending itself out. A Mac OS X scare in the form of MP3Concept was announced 8 April. Said to be a benign Trojan, MP3Concept turned out to be nothing more than a bad proof-of-concept that never made it into the wild. The end of April saw the Sasser worm which is the first to effectively use the LSASS Windows vulnerability; a vulnerability that allowed the worm to spread via an open FTP port instead of through E-mail (even though Microsoft had already issued a patch for the vulnerability -- yet another example of people not paying attention to operating system security updates). Toward the end of May Apple issued critical patches to OS X when a vulnerability that could spread via E-mail and mal-formed Web pages was found. The vulnerability would allow AppleScript scripts to run unchecked; even to the point of deleting the home directory. The proof-of-concept Worm W64.Rugrat.3344 showed up the end of May. This is claimed to be the first malware that specifically attacks 64-bit Windows files only (it ignores 32-bit and 16-bit files). It was created using IA64 (Intel Architecture) assembly code. In June Symb/Cabir-A appeared to infect Nokia Series 60 mobile phones. The worm is designed to spread to nearby Bluetooth-enabled devices. JS/Scob-A appeared in the last half of June. It was special in that it used Javascript to infect Microsoft's IIS Server HTML files through an unpatched vulnerability. User's visiting infected sites were then infected via a download from a Russian site (which was quickly closed down) using an unpatched vulnerability in the IE browser. Mid-July WCE/Duts-A showed up. This was another crude proof-of-concept virus relating to the PocketPC. The virus writer was apparently trying for attention as this text is in the virus: "This is proof of concept code. Also, i wanted to make avers happy.The situation when Pocket PC antiviruses detect only EICAR file had to end ..." Early September saw W32/Amus-A show up. The only thing that qualified this beast to even be mentioned here was that it uses the Microsoft Speech engine in Windows to read out loud: "hamsi. I am seeing you. Haaaaaaaa. You must come to turkiye. I am cleaning your computer. 5. 4. 3. 2. 1. 0. Gule. Gule." where "Gule" is Turkish for "Bye" and "Hamsi" is a small fish found in the Black Sea. On 14 September that paragon of virus-free file type, the JPEG image, came under attack. To be accurate, the image file itself is not so much to blame as a Microsoft common .DLL file that processes the image file type and has a buffer overrun error that could allow someone to add malicious code to a JPEG image which can then open holes in an attacked system. Shortly after, some Trojan exploits started to appear.

Execution Of Viruses

Early viruses were pieces of code attached to a common program like a popular game or a popular word processor. A person might download an infected game from a bulletin board and run it. A virus like this is a small piece of code embedded in a larger, legitimate program. Any virus is designed to run first **when the legitimate program gets executed**. The virus loads itself into memory and looks around to see if it can find any other programs on the disk. If it can find one, it modifies it to add the virus's code to the unsuspecting program. Then the virus launches the "real program." The user really has no way to know that the virus ever ran. Unfortunately, the virus has now reproduced itself, so two programs are infected. The next time either of those programs gets executed, they infect other programs, and the cycle continues.

If one of the infected programs is given to another person on a floppy disk, or if it is uploaded to a bulletin board, then other programs get infected. This is how the virus spreads.

The spreading part is the **infection** phase of the virus. Viruses wouldn't be so violently despised if all they did was replicate them. Unfortunately, most viruses also have some sort of destructive **attack** phase where they do some damage. Some sort of trigger will activate the attack phase, and the virus will then "do something" -- anything from printing a silly message on the screen to erasing all of your data. The trigger might be a specific date, or the number of times the virus has been replicated, or something similar.

Impact and Effects

- Nuisance
- Spoofing
- Denial of Service
- Overwriting and Data diddling
- Destruction
- Psychological
- "Netspionage"
  - Siphoning data
  - Exposing vulnerabilities
- Compromise or Loss of Data
- Loss of Productivity
- Denial of Service
- Data Manipulation
- Loss of Credibility
- Loss of Revenue
- Embarrassment

Protection Against Viruses

You can **protect** yourself against viruses with a few simple steps:

➢ Secure Operating System

If you are truly worried about traditional (as opposed to e-mail) viruses, you should be running a more secure operating system like UNIX. You never hear about viruses on these operating systems because the security features keep viruses (and unwanted human visitors) away from your hard disk.

➢ Antivirus Protection

If you are using an unsecured operating system, then buying **virus protection software** is a nice safeguard.

Symantec Corporation

Symantec Corporation helps make users productive and keep their computers safe and reliable anywhere and anytime. Symantec offers a broad range of solutions and is acclaimed as a leader in both customer satisfaction and product brand recognition. The company is focused on addressing customer needs in three main application areas: the Norton Product line of anti-virus and PC-assistance products; the pcANYWHERE, WinFax, and ACT! product lines that cater to remote user productivity; and the Café product lines in Internet development tools.

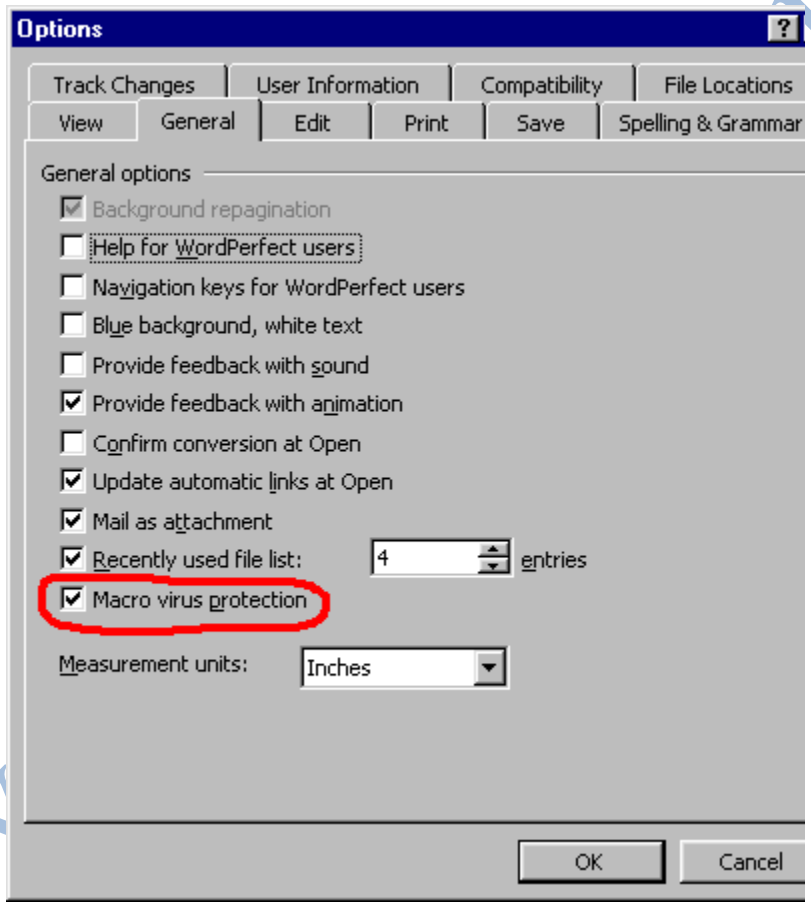
Founded in 1982, the company's global operations span North America, Europe, Japan, and several fast-growing markets throughout Asia Pacific and Latin America. Traded on Nasdaq under the symbol SYMC, Symantec Corporation is based in Cupertino, California, and employs more than 2,000 people.

➢ Unknown Sources Protection

If you simply **avoid programs from unknown sources** (like the Internet), and instead stick with commercial software purchased on CDs, you eliminate almost all of the risk from traditional viruses. In addition, you should **disable floppy disk booting** -- most computers now allow you to do this, and that will eliminate the risk of a boot sector virus coming in from a floppy disk accidentally left in the drive

- **Macro Virus Protection**

You should make sure that **Macro Virus Protection** is enabled in all Microsoft applications, and you should NEVER run macros in a document unless you know what they do. There is seldom a good reason to add macros to a document, so avoiding all macros is a great policy.



Open the Options dialog from the Tools menu in Microsoft Word and make sure that Macro Virus Protection is enabled, as shown.

- **Email-Virus Protection**

You should **never double-click on an attachment that contains an executable that arrives as an e-mail attachment.** Attachments that come in as Word files (.DOC), spreadsheets (.XLS), images (.GIF and .JPG), etc., are data files and they can do no damage (noting the macro virus problem in Word and Excel documents mentioned above). A file with an extension like EXE, COM or VBS is an executable, and an executable can do any sort of damage it wants. Once you run it, you have given it permission to do anything on your machine. The only defense is to never run executables that arrive via e-mail.

MOBILE VIRUSES

❖ WAP THREATS

The use of WAP-enabled mobile phones is booming. Cellular phones with support for WAP (Wireless Application Protocol) allow users to access a wide variety of services.

WAP enables users to do on-line banking, monitor stock markets, use email, access the Internet – all from their mobile phones. Future WAP services with positioning support will enable even more advanced services – for example, you could ask your phone to find the closest restaurant in a strange city and your phone would answer back with map and directions.

When it comes to WAP security, why worry? From the outset, vendors of mobile phones and WAP servers have ensured that much consideration was given to on fidentiality and privacy issues for WAP data, as well as to user authentication. Add this to the fact that data integrity checking has been taken into account, and you could be forgiven for thinking that the WAP infrastructure is already secure enough. However, we believe that there are still a number of security issues to be resolved. Firstly, there is no content security for the WAP infrastructure, and yet this is where one of the biggest threats typically lies. As we have already seen in the desktop-PC world, content-related security is the single biggest security issue for home and corporate users alike. Even now, we receive an average of seven new PC virus samples every day, with actions that range from benign to potentially catastrophic. In the telecommunications world, content has traditionally been speech – with no security risks involved. Now the content is code, and the whole picture changes. The WAP infrastructure has not taken executable mobile content – such as downloadable programs into account from a content-security point of view. The WAP content requested by the mobile device and returned by the origin server can, for example, contain WML cards, which may display text or pictures, working similarly to HTML pages on the Web. The pages can also contain script written with WML Script language – which is a close relative to the JavaScript scripting language. As a side note, several PC viruses written with JavaScript were discovered during 1999 and 2000.

The WLAN weak link

A security weakness in the encryption standard used within IEEE-based WLANs

has been uncovered. Three cryptographers have described a practical way of attacking the key scheduling algorithm of the RC4 cipher, in a paper entitled Weaknesses in the key scheduling algorithm of RC4.

The RC4 cipher forms the basis of the WEP encryption that is used in IEEE 802.11b wireless networks. The paper's authors discovered several ways to uncover patterns in packets of information passing over WLANs.
These patterns can be used to figure out the WEP encryption "key" and the number used to scramble the data being transmitted. Once the key is recovered, it can be used to decrypt the messages. According to the authors, using a longer key-128 bits instead of the current WEP standard of 40 bits-does not make it harder for attackers to uncover the process. The paper provides a more practical approach to breaking RC4 than previous publications and lends fresh urgency to the work of two IEEE groups grappling with the 802.11 vulnerabilities.

However, the Wireless Ethernet Compatibility Alliance said enterprise users should continue to use WEP because only skilled crypto analysts would be able to exploit the weakness. Enterprises could also use several existing tools for additional security, such as VPNs, IPSec, and RADIUS authentication servers.

In addition, many WLAN vendors have introduced proprietary encryption schemes because of the known weaknesses in WEP. However, these schemes are not interoperable with each other. There have been
other problems uncovered in the WEP structure but the latest discovery is more significant because an attack could be carried out faster and with fewer resources.

One emerging solution is from the 802.1x group that is focused on overall network security and authentication. Another is the 802.11i group that is making use of some of the 802.1x work to overhaul the identified WEP vulnerabilities. These initiatives are scheduled to be finalized by year end and vendors are likely to have products out soon

❖ Potential PDAs Problems

What about palmtop computers and PDAs-can they be infected by computer viruses? PDAs run specially written scaled-down operating systems, such as EPOC, PalmOS or PocketPC. They are often connected to home or office PCs to synchronize the data between the two machines. This presents an opportunity for viruses to spread onto them.

Yet, no viruses currently exist for the PocketPC and EPOC operating systems, although there is no technical reason why they could not be written. There is a virus called Palm/Phage, which is able to infect Palm OS, but it is not in the wild and poses little threat.

Nonetheless, it is sensible to keep backups of any Palm applications and data. There is also a Trojan horse known as Palm/Liberty-A, which is able to infect the Palm OS. It deletes Palm OS applications and was distributed in the 'warez' community. Like Phage, it is low risk and you are unlikely to ever encounter it.

❖ Bluetooth Bugs

Bluetooth is a standard for low-power radio data communication over very short distances. Computers, mobiles, fax machines and even domestic appliances, like video recorders, can use Bluetooth to discover what services are provided by other nearby mobile devices and establish transparent links with them.

Software that utilizes Bluetooth is currently emerging. For example, Sun's Jini technology allows devices to form connections, exchange Java code automatically and give remote control of services. The worry is that an unauthorized user, or malicious code, could exploit Bluetooth to interfere with these services.

However, Bluetooth and Jini are designed to ensure that only trusted code from known sources can carry out sensitive operations. For now, this means that it is highly unlikely for a virus outbreak to occur.

❖ CABIR

As recently as this spring, the idea of a virus that infects mobile phones was a scary bedtime story for the wireless industry, viewed in a similar vein as the threat of global warming: important but not imminent. All that changed in June when the world got a glimpse of the first mobile phone virus, Cabir. Since then, the industry has been scrambling to prepare itself for Cabir's offspring, hoping to divine the best defense strategies before the scary bedtime story becomes reality.

Cabir is the first ever computer virus that can infect mobile phones and has been discovered by the French arm of Kaspersky Labs, a Russian security software developer.

Cabir can spread via cell phones and is the first malicious code with such ability. Anti-virus software developers, however, have yet to detect any harmful effects of the virus on cell phones.

Kaspersky Labs said that Cabir seems to have been developed by some global group that specializes in creating viruses to demonstrate that 'no technology is reliable and safe from their attacks.'

The developers of Cabir apparently have not designed the virus -- or worm -- to propagate on a massive scale, but to demonstrate that cell phones and PDAs can be infected by malicious code.

This malicious code spreads to devices that run under Symbian OS, which is used in many models of phones, including some manufactured by Nokia, Siemens and Sony Ericcson.

Cabir spreads in a file called 'Caribe.sis,' which installs itself automatically on the system when the user accepts the transmission. It displays a message on the screen with the text: 'Caribe' and then starts a continuous search for other devices to send itself to, although these must be connected via Bluetooth technology.

Bluetooth's transmission range is 30 feet. The virus is only able to jump from phone to phone within that range. Also the phone must have the correct OS installed and the appropriate settings -- that is, it has to be set for a known number.

It is able to scan for phones that are also using the Bluetooth technology and is able to send a copy of itself to the first handset that it finds.

On the other hand, it is possible that the Caribe.sis file copies itself to other devices using Bluetooth, such as some printers.

Cabir uses Symbian Series 60 phones to replicate it, sending a clone to the first Bluetooth-enabled device it can find in the area (even a printer) when a user OKs two installation prompts. It was launched as a “proof of concept” by a member of 29A Labs, a group of Eastern European hackers who develop innocuous viruses with the benevolent aim of exposing security weaknesses. Their incentive to create Cabir was likely the notoriety of boasting the world's first mobile phone virus. (Security nerds call Cabir a worm, not a virus, because it does not attach itself to a host program. Even bigger security nerds point out Cabir is not a worm because it cannot propagate itself; it relies on the user to do so by actively installing it. Symbian refers to it as malware.)

Cabir had no real payload — no harmful effect other than the word “Caribe” displayed on infected devices — and it was sent directly to security experts rather than the general population, but it proved its concept as planned and sparked a wave of fear that a less scrupulous group of hackers would build on Cabir's design to unleash something far more sinister. It was a mid-summer wake-up call to the mobile phone industry, said Richard Wong, general manager of messaging and anti-abuse at software vendor Open wave Systems.

According to the anti-virus software developer F-Secure, the discovery of Cabir is proof that the technologies are now available to create viruses for mobile phones and that they are now known to the writers of computer viruses.

Anti-virus experts have been warning for months that mobile phone viruses are set to multiply, given the increasingly diverse uses of mobile phones.

MOTIVES

Viruses are written for a variety of reasons, such as curiosity, a challenge, or to gain wider attention. Some virus writer groups are known to target any new platform, just be able to say they were the first to write a virus for this platform. At the time of writing, the WAP infrastructure is still emerging and the uptake of WAP devices is still increasing. Currently therefore, WAP devices do not present a big enough target and so no WAP-specific viruses have yet been seen.

However, a growing threat is coming in from the horizon as the power of WAP devices is set to increase dramatically with future WAP protocol versions.

As WML also increases in sophistication, so do the opportunities for creating more advanced, malicious code. When the first WAP virus hits, it could spread as fast or faster than similar PC viruses. The implications for the WAP infrastructure as a whole are ominous if this were to occur. For example, public confidence for an activity such as wireless banking would deteriorate if the threat of WAP viruses loomed large.

VIRUS ACTIONS

The only way to deal with these threats is to secure all remaining gaps in WAP security, before such attacks are mounted. The industry is in a unique position to benefit from past experience and proactively prevent the type of weaknesses in infrastructure that caught us unaware with past computer incidents. There are no viruses on WAP yet – we still have time to react.

Before we consider the key security issues, and solutions that will help identify and meet WAP content security risks, it is best to understand how a basic WAP network is composed. There are three logical components: the WAP client (or mobile terminal), the WAP gateway and the origin server. The origin server is located in the traditional

Internet domain and functions like an ordinary Web server by providing storage for WAP content. The WAP gateway interconnects the Internet domain with the mobile network domain by providing the mobile terminal with Internet access. The mobile terminal roams in the mobile network and sends encoded content requests to the origin server via the WAP gateway. WAP needs more functionality in order to be useful and for it to really take off the ground.

Unfortunately, more functionality means more risks. The power of WAP devices is set to increase dramatically with future functions set to be included in the WAP specification in the near future. Such functions include making phone calls, accessing and modifying phone book data, and sending Short Messaging Service (SMS) messages.

With such functionality available to WML scripts, it is not difficult to imagine a virus, which would spread by accessing your phone book and sending a link to itself in SMS text messages to all the phone numbers, found within. Subsequently, the virus could do damage by either deleting or modifying your phone book, or by starting to make phone calls to pay-per-minute numbers –in the middle of the night. With such a feature, virus writers could easily make money with their viruses – thus providing an obvious motivation. As WML increases in sophistication, so do the opportunities for creating more sophisticated, malicious code.

PROTECTION AGAINST VIRUS

The Symbian OS (operating system) smart phones will provide on-device protection, similar in fashion to antivirus protection programs for PCs, with automatic over-the-air antivirus updates for a monthly fee. The software will not come loaded into the device, but can be downloaded from the F-Secure Web site, according to Nokia .The Nokia 6670 will be the first mobile phone in its Series 60 line to offer the mobile virus protection, though users of other Series 60 mobile phones will also be able to purchase the antivirus protection software.

F-Secure is also in talks with other handset manufacturers about offering similar antivirus protection. He declined to name any companies or set out potential dates for availability. This announcement is a starting point for us and we have been testing the service with a variety of handsets from different vendors and in several operator networks.

Nokia, based in Espoo, Finland, already offers antivirus software through F-Secure for its Communicator line of mobile devices, but the protection offered for the Nokia 6670 is a greatly improved version in terms of both features and pricing options.

"The first general offering for the mobile antivirus software came a couple of years ago, but this version has a whole new infrastructure. "For example, it has a patented SMS (short message service) update mechanism and HTTPS (Hypertext Transport Protocol Secure) connections. Plus, there is a big difference in the actual client.

The monthly pricing plan is also a first for F-Secure. The first month of the service will be free trial period and thereafter, users will be charged a licensing fee that will include the cost of updates, he said. "Before you paid on a yearly basis, but by paying monthly, you just buy the protection that you need"

The final decision about pricing has yet to be made but will be finalized by the time the phone ships "some time in October. According to the company's current estimates, the antivirus mobile protection licence will cost about €2.95 ($3.62) per month, but early buyers will most likely be offered a discounted price of €1.95 per month.

The handset will have an estimated retail price of €500 without taxes, according to Nokia "That price will vary from market to market.

The Nokia 6670 will come in two tri-band versions, optimized for GSM (Global System for Mobile Communications) networks in the EMEA (Europe, Middle East and Asia) markets (on 900MHz, 1800MHz and 1900MHz bands), and in the Americas (on 850MHz, 1800MHz and 1900MHz bands). Both versions will be able to roam in GSM networks across regions.

Nokia is also offering addition security through its mobile VPN (virtual private network) client and SSL (Secure Sockets Layer) encryption for Web-based applications. Lehmusvirta stressed that there is nothing about the Nokia 6670 that makes it particularly susceptible to viruses and that Nokia knows of no capabilities within any of its devices that a virus might exploit.

The rational behind the phone is as a smart phone targeted at business users who use data in their daily work, and we want to offer them some security for that data. There has been a common perception for many years by the entire industry that mobile devices will become a target of viruses, though to date this kind of threat is small. We want to begin protecting against it now.

After a series of three malicious program targeting wireless devices were discovered in between June and August, security specialists stepped up their warnings of the pending possibility of serious attacks against mobile phones and PDAs (personal digital assistants).

In June, Antivirus company Kaspersky Labs Ltd. said it discovered Cabir, a network worm infecting phones running the Symbian mobile phone operating system by Symbian Ltd. At the time, the company characterized Cabir as the first-ever computer virus capable of spreading over mobile phone networks.

Cabir was followed in August by the discovery of the so-called Backdoor.Bardor.A virus, a Windows CE Trojan horse program designed to give attackers control over Pocket PC mobile devices. A few days later, a Symbian Trojan program infecting phones using the Series 60 user-interface platform cropped up with the ability to make the phones send text messages without the knowledge of the user.

The threats we saw for the first time this summer have not been big ones, but it was a proof of concept in a way. It shows the point that hackers and virus writers are targeting all types of mobile handsets. There is no reason to panic, but it is good to be ready, to prepare for the future with protective insurance. We learned that from the PC world.

F-Secure claims its mobile anti virus software service is the first commercially available product for protecting Symbian OS smart phones but similar programs can be expected in the very near future.

With the convergence of both the fixed and the wireless worlds, comes the increasing need to monitor not just for malicious code but also for an influx of Spam that could clog up networks. It isn't an issue today, but it's a potential issue that could exist down the line.

❖ SHUTTING THE STABLE DOOR

The first, and biggest, step in delivering content security into the WAP world is a gateway-level solution for protecting the WAP infrastructure. A WML script scanner is integrated with the WAP gateway, which detects and removes malicious code before it is passed to users' devices. Gateway protection will also ensure that when a new virus is found, counter-measures to provide protection can be developed quickly and distributed over the Internet-based framework to WAP servers worldwide. This type of solution has the advantage for the current WAP infrastructure (see Figure 2) of requiring no client software and leaving no 'footprint' on the client device, making it suitable for the current generation of WAP phones, for example. The WAP subscriber simply receives virus free content in a way that's transparent to them, and the solution is centrally managed by the content provider to ensure optimum control of content.

Conclusion

The current trend seems to be for people to worry about the potential threats of tomorrow, which may never come to fruition, as opposed to the real risks of today. The best advice to follow is to remain alert to what the dangers are right here, right now and to protect against them. While you're concerning yourself about the future, you could be missing what's right under your nose.

References

- www.google.com
- www.wikipedia.com
- www.studymafia.org