Computer Viruses: The Threat Today and The Expected Future

Master's thesis performed in Information Theory Division

By

Xin Li

LITH-ISY-EX-3452-2003

2003-09-29

LINKÖPINGS UNIVERSITET

Avdelning, Institution
Division, Department

Institutionen för Systemteknik
581 83 LINKÖPING

Datum
Date
2003-09-25

Språk
Language
Svenska / Swedish
X Engelska / English

Rapporttyp
Report category
Licentiatavhandling
X Examensarbete
C-uppsats
D-uppsats
Övrig rapport

ISBN

ISRN LITH-ISY-EX-3452-2003

Serietitel och serienummer **ISSN**
Title of series, numbering

URL för elektronisk version
http://www.ep.liu.se/exjobb/isy/2003/3452/

Titel
Title

Datorvirus: Dagens situation och förväntad utveckling

Computer viruses: The threat today and the expected future

Författare
Author

Xin Li

Sammanfattning
Abstract
This Master's Thesis within the area computer security concerns "Computer viruses: The threat today and the expected future".

Firstly, the definitions of computer virus and the related threats are presented; Secondly, current situation of computer viruses are discussed, the working and spreading mechanisms of computer viruses are reviewed in details, simplistic attitude of computer world in computer virus defence is analyzed; Thirdly, today's influencing factors for near future computer virus epidemics are explained, then it further predicts new possible types of computer viruses in the near future; Furthermore, currently available anti-virus technologies are analyzed concerning both advantages and disadvantages; Finally, new promising trends in computer virus defence are explored in details.

Nyckelord
Keyword
computer virus, computer worm, theoretical suggestion, anti-virus technology, trend

Computer viruses: The threat today and the expected future

Master's Thesis in Computer Security

Linköping Institute of Technology

By

Xin Li

LiTH-ISY-EX-3452-2003

Supervisor: Viiveke Fåk
Examiner: Viiveke Fåk

Department of Electrical Engineering

Linköping University

Linköping 2003-09-29

Abstract

This Master's Thesis within the area computer security concerns "Computer viruses: The threat today and the expected future".

Firstly, the definitions of computer virus and the related threats are presented; Secondly, current situation of computer viruses are discussed, the working and spreading mechanisms of computer viruses are reviewed in details, simplistic attitude of computer world in computer virus defence is analyzed; Thirdly, today's influencing factors for near future computer virus epidemics are explained, then it further predicts new possible types of computer viruses in the near future; Furthermore, currently available anti-virus technologies are analyzed concerning both advantages and disadvantages; Finally, new promising trends in computer virus defence are explored in details.

Acknowledgements

I would like to thank my academic supervisor and examiner, Viiveke Fåk at the Department of Electrical Engineering at Linköping University, for her guide and help. Also, I would like to thank my sister Li Zhihong for her continuous encouragement throughout my study at Linköping University.

Table of Contents

Table of Figures

1 Introduction

In the mid-eighties, so legend has it, the Amjad brothers of Pakistan ran a computer store. Since they were frustrated by computer piracy, they wrote the first computer virus in the world, a boot sector virus called Brain. From those simple beginnings, an entire counter-culture industry of computer virus creation and distribution emerged, leaving us today with several tens of thousands of computer viruses.

In just over a decade, most of us have been familiar with the term computer virus. Even those of us who don't know how to use a computer have heard about computer viruses through Hollywood films such as Independence Day or Hackers (although Hollywood's depiction of viruses is usually highly inaccurate). International magazines and newspapers regularly have virus-scares as leading stories. There is no doubt that our culture is fascinated or frustrated by the potential danger of these computer viruses.

This paper aims to discuss the current situation of computer viruses, suggest theoretically about new types of computer viruses, predict about new possible types of computer viruses in the near future, and explore new promising trends in computer virus defence.

2 The definitions of computer virus and the related threats

It seems that even people who are familiar with computing often have unclear and even controversial understanding of the terms which are associated with computer virus, computer worm, trojan horse, malware,etc. In fact, exact defintions for the terms - computer virus, computer worm, trojan horse, malware,etc. have not been agreed on even among computer anti-virus researchers.

There is no common agreement on exact definitions for the terms - computer virus, computer worm, trojan horse, malware, but the main functions for every type of program code and software have been generally recognized by computer anti-virus researchers. One reason for the difficulties is that it is impossible to say in all given circumstances if a given program is malicious or not. For instance, a program which formats hard disks can be considered

either as harmful or useful, depending on the purpose for which the program is used.

Marko Helenius's definitions are referenced as follows:

Malware: The word 'malware' is an abbreviation of the term 'malicious software'. Refers to a program code which has been deliberately made harmful. This includes such program code classes as computer viruses, computer worms, trojan horses, joke programs and malicious toolkits. This list may not be exclusive[1].

Computer virus: Refers to a program code which has a capacity to replicate recursively by itself. Computer viruses may include operations, which are typical for trojan horses and malicious toolkits, but this does not make such viruses trojan horses or malicious toolkits[1].

Computer worm: Refers to an independent program code which has a capacity to replicate recursively by itself. Independent means that a computer worm does not have a host program which the worm has infected or replaced by its own code. Computer worms are a subgroup of computer viruses. Computer worms may include operations which are typical for trojan horses and malicious toolkits, but this does not make such worms trojan horses or malicious toolkits[1].

Trojan horse: Refers to a self-standing program code which performs or aims to perform something useful, while at the same time it intentionally performs, unknowingly to the user, some kinds of destructive function. Self-standing means that, in distinction to viruses, the program code does not have the capaibility to replicate by itself. The program code may be attached to any part of a system's program code. Trojan horses may include operations which are typical for malicious toolkits but this does not make such trojan horses malicious toolkits[1].

Malicious toolkit: Refers to a toolkit program which has been designed to help such malicious intentions, which are aimed against computer systems. This includes such programs as virus creation toolkits and programs, which have been designed to help hacking[1].

Joke program: Refers to a program which imitates harmful operations, but does not actually accomplish the object of imitation and does not contain any other malicious operation[1].

In this report, Marko Helenius's definitions are adopted, computer worms are considered to be a subgroup of computer viruses. So, both computer viruses and computer worms are discussed with no attempt to treat them separately in this report.

3 Current situation of computer viruses

The main source of the information in the sections 3.1 and 3.2 is the organization -- Internet FAQ Archives[2].

3.1 How do computer viruses work?

A file viruses attaches itself to a file, which is usually an executable application (e.g. a DOS program or a word processing program). Generally, file viruses don't infect data files. But, data files can contain embedded executable code such as macros, which can be exploited by computer virus or trojan horse authors. Recent versions of Microsoft Word and Excel are particularly vulnerable to this type of threat. Text files such as batch files, postscript files, and source codes that contain commands which can be compiled or interpreted by another program are potential targets for malware (malicious software).

Boot sector viruses modify the program which is in the first sector (boot sector) of every DOS-formatted disk. In general, a boot sector infector executes its own code (that normally infects the boot sector or partition sector of the hard disk), then continues the PC bootup (start-up) process. In most cases, all write-enabled floppy disks which are used in that PC from then on will become infected.

Multipartite viruses have some of the features of both the above types of viruses. When an infected file is executed, it typically infects the hard disk boot sector or partition sector, and therefore infects subsequent floppy disks which are used or formatted in the target system.

Many of these types of viruses related to DOS programs were commonly found several years ago when DOS programs were mainly used in the computers, but are rarely found and reported nowdays[3].

Typically macro viruses infect global settings files such as Microsoft Word, Excel templates so that subsequently edited documents are infected with the infective macros. Macro viruses will be discussed in more details at next section.

Stealth viruses are the viruses that go to some length to hide their presence from programs that might notice.

Polymorphic viruses are the viruses that cannot be detected by searching for a simple, single sequence of bytes in a possibly infected file, because they change with each replication.

Companion viruses are the viruses that spread through a file that runs instead of the file the user intended to run, and then runs the original file. For example, the file MYAPP.EXE may be 'infected' by creating a file called MYAPP.COM. Because of the way DOS works, when the user types MYAPP at the C> prompt, MYAPP.COM is run instead of MYAPP.EXE. MYAPP.COM runs its infective routine, then quietly executes MYAPP.EXE. N.B. this is not the only type of companion (or 'spawning') virus. A more modern way to achieve this is to create files that have the same name, but are placed on another branch in the file tree. If the file is cleverly placed, failure to write the full path will result in executing the false file.

Armoured viruses are the viruses which are specifically written to make it difficult for an anti-virus researcher to find out how they work and what they do.

3.2 How do computer viruses spread?

3.2.1 Boot sector virus

A PC is infected with a boot sector virus (or partition sector virus) if it is (re-)booted (normally by accident) from an infected floppy disk in floppy disk drive. Boot Sector/MBR infectors used to be the most commonly found viruses several years ago, and could not usually spread across a network.

These viruses are spread by accident through floppy disks that may come from virtually any source: unsolicited demonstration disks, brand-new software (even from reputable sources), disks which are used on user's PC by salesmen or engineers, new hardware, or repaired hardware.

3.2.2 File virus

A file virus infects other files when the program to which it is attached is run, and so can spread across a network (often very quickly). They can be spread from the same sources as boot sector viruses, but also from sources such as Internet FTP sites and bulletin boards (This applies to trojan horses also).

3.2.3 Multipartite virus

A multipartite virus infects boot sectors and files. An infected file is often used to infect the boot sector, therefore, this is one case where a boot sector infector could spread across a network.

3.2.4 Macro virus

A macro is an instruction which carries out program commands automatically. Many common applications (e.g. word processing, spreadsheet, and slide presentation applications) use macros. Macro viruses are macros which self-replicate. If a user accesses a document that contains a viral macro and unwittingly executes this macro virus, then, it can copy itself into that application's startup files. Now, this computer is infected: a copy of the macro virus resides on this computer.

Any document on that computer which uses the same application can then become infected. If the infected computer is on a network, the infection is highly possible to spread quickly to other computers on the network. Furthermore, if a copy of an infected file is passed on to anybody else (e.g. by floppy disk or email), the virus can spread to the recipient's computer. This process of infection will end only when the virus is noticed and all viral macros are eradicated. Macro virus is the most common type of computer viruses now in the wild[3]. Many popular modern applications allow macros. Macro viruses can be written with very little specialist knowledge,

and these viruses can spread to any platform on which the application is running. Most current macro viruses and trojan horses are specific to Microsoft Word and Excel, but, many applications, not only Windows applications, have potentially damaging and/or infective macro capabilities also. Macro languages such as WordBasic and Visual Basic for Applications (VBA) are very powerful programming languages in their own right. Word and Excel are particularly vulnerable to this threat, due to the way in which the macro language is bound to the command/menu structure in vulnerable versions of Word, the way in which macros and data can exist in the same file, and the eccentricities of OLE-2. But, the main reason for their 'success' is that documents are exchanged far more frequently than executable programs or floppy disks, which is a direct result of email's popularity and web use.

3.2.5 Email worm

An overwhelmingly large proportion of virus infections today is caused by the infected email attachments. The ease with which a user can click on an email attachment and launch an application is a significant factor in the spread of email worms. If the email content is sufficiently inviting (e.g. 'kindly check the attached LOVELETTER coming from me'.) and the visible email attachment extension sufficiently innocent in the eyes of the user (e.g. LOVE-LETTER-FOR-YOU.TXT.vbs - text files cannot carry an infection, can they?), the temptation for a user can become overwhelming.

The danger of computer virus infection through attachments is, of course, not confined to email. Newsgroup postings are capable of carrying attachments also.

Several years ago most viruses spread primarily via floppy disk, but the Internet has introduced new virus distribution mechanisms. With email now used as an essential communication tool, viruses are spreading faster than ever. Viruses attached to email messages can infect an entire enterprise in a matter of minutes, costing companies millions of dollars annually in lost productivity and clean-up expenses.

3.3 Simplistic attitude of computer world in computer virus defence

It is common that computer viruses depend on both the known vulnerabilities of computer softwares and systems, social engineering, and computer user's carelessness for their spreading. Modern information system users tend to rely heavily on technical solutions to the problems caused by computer viruses, with an emphasis on 'symptomatic' response. A system is often considered 'protected' if it is running a virus scanner, and so many companies, organizations and individuals rely much less on other methods such as user education, user awareness, operational procedures, and security policy. It is too simplistic to rely on vendors to provide a 'worry-free' computer virus defence solution in a 'plug-and-play' fashion.

Anti-virus researchers agree that there is a limited set of possible defences to combat computer viruses. A simple dichotomy would be technical versus non-technical methods to a computer virus attack. These methods can then be further viewed from a proactive or reactive perspective. After evaluating these methods, it shows that reliance on reactive technology alone may be a very weak defence against computer virus attacks. The problem of current computer virus defence can be illustrated like this: Standard anti-virus technology looks for the known viruses when the never-seen-before viruses are actually the ones which cause the problems. It does not matter how quickly the anti-virus vendors respond, their fixes are always reactive. The fortunate companies learn about a new virus and get the virus definition update from the anti-virus vendors after another company's systems have already been attacked.

A more rigorous discussion on the problem of technical defence against computer virus attack is provided by Fred Cohen in his book "A Short Course on Computer Viruses". Cohen points out that "there are three and only three things that you can ever do to absolutely and perfectly prevent a computer virus from spreading throughout a computer system or network; limit sharing, limit transitivity, or limit programming. This has been proven mathematically." [4]

The use of a system which has no sharing, no transitivity and no outside programming is difficult to imagine in our common understanding of the role of information systems except in some special situations. We should be prepared to discuss imperfect solutions, and to be willing to admit to some

failures. We should admit the fact that there is no perfect defence. We should choose active methods which include avoidance, detection and cure, rather than only prevention. It means that system users need to adopt a healthier system 'life style' as well as using the best of 'modern medicine' to avoid catching a virus.

3.3.1 Security practices

Some good, basic security practices can really help to reduce the risk of computer virus infections, and decrease the cost (and pain) of cleanup after computer virus infection. Some good examples include keeping good clean backups, reviewing security controls on a regular basis, using access controls for system users, limiting connectivity, audit, limiting/controlling media such as diskettes, cd-r/w, zip drives, etc., user awareness and education, risk analysis and reactive and proactive defences.

Use of good security techniques in many areas can help protect information systems from computer virus attacks. For instance, digital signatures serve to authenticate the sender of a message. Public key (or other) encryption serves to preserve the confidentiality of a message. These same techniques might also reduce the possibility of a computer virus attack. Many computer virus attacks take advantages of email attachments. Users who really rely on 'protected' email services will not trust a message without a digital signature, and will not believe a message which can’t produce a valid checksum.

3.3.2 Security policy

The development of a good security policy is an important first step towards a healthy system. The security policy should state clearly the expectations of the organization; for instance "no floppy disk can be used in any system in the research lab". Also, the policy should be matched with a procedure (in the forgoing example, the best procedure would be to lock or remove the floppy disk drives from all computers in the research lab). A policy without procedure is vacuous at best. It is necessary to assign persons within the organization the roles and responsibility to make the security policy really effective.

One method, also recommended for other reactive IT security matters, is the establishment of a Computer Security Incident Response Capability

(CSIRC), normally implemented as a CERT (Computer Emergency Response Team). CSIRC capability can provide both reactive response and take proactive responsibility for the organization response to the threat of computer virus attacks. "A CSIRC is a combination of technically skilled people, policies, and techniques that constitute a proactive approach to handling computer security incidents."[5] A typical CERT team would be consist of both technical and non-technical users and would be responsible for user support, interaction with senior management, and press liaison besides technical response such as maintainance and repair. The CERT would be capable of providing quick, professional and practiced response.

When the CERT is not in emergency state, it would be also responsible for contingency arrangements, training CERT members, educating, motivating and training organization employees, risk analysis, worst case scenario and recovery, and planning and practicing (live exercises), etc.

CERT would make sure that virus siganature files are constantly kept up to date, promote strong user awareness and education strategies, evaluate backup and recovery techniques, and act as a 'system health monitor' to reduce the overall risk of a successful computer virus attack.

Also, the CERT would work as the coordinator of site alerts in the event of increased risk. This would reduce the threat of the email hoaxes (such as 'pen pals') when users who receive such warnings would send them to the CERT for response (rather than "tell all their friends") and would become ignoring warnings which do not originate from the CERT.

4 Theoretical suggestions about new types of computer viruses.

4.1 Today’s influencing factors for near future computer virus epidemics

Six technological influencing factors have had a huge impact on the variety and complication of computer viruses and worms: computing infrastructural homogeneity, ubiquitous programmability, technology migration to home, and increasing connectivity through a homogeneous communication mechanism.

4.1.1 Broadband in the Home

When more and more users adopt broadband communication technologies in the home, we expect that the incidence of computer worms which are targeting at home users and small businesses will grow rapidly. Also, this will seriously affect telecommuters in both government and industry. Nowadays, home users with modems are assigned dynamic network addresses every time when they login to the Internet. Because users log on and off very frequently, this makes it very difficult for a computer worm to target at such a computer and spread to it.

But, when users migrate to broadband technologies such as cable modems or Digital Subscriber Line(DSL), they will increasingly have constant, static connections to the Internet, which make their computer an easy target. Computer hackers or roaming computer worms will be able to easily enumerate home user Internet addresses and use them to attack these computers. They can rapidly spread through the VPN and onto the corporate or government network once they have a foothold on the home computer. For instance, if home users are infected by a Word for Windows macro virus, it may easily infect their work documents and then transfer these to their company PC through the VPN. In the same way, a computer worm like ExploreZip could potentially spread through the VPN onto other visible computers on the company network.

In addition, it is expected that when more users adopt broadband technologies at home, consumer-oriented connected applications will quickly grow in popularity. Nowadays, products like PointCast TMdot are mainly used as company desktops, but are less appealing for the home user because the home user has a lower-speed connection to the Internet. But, when broadband technologies become more and more popular, these connected applications will grow in popularity; real-time stock tickers, personal web servers, search agents, and “instant message”/chat programs will be running on each desktop. Moreover, just like that we have already seen in the Office application space, vendors will start adding macro/programming support to these applications to extend their abilities for powerful uses.

Although these connected applications will improve the quality of the computing experience, every one also contributes to security risks for the

home user. It is expected that the next generation of computer worms will exploit these connected and often not security-conscious applications and exploit them as back doors into home systems and then into the enterprise. In such an environment, a computer worm like Melissa or Loveletter will easily infect huge numbers of home users.

Since threats which affect the home inevitably find their way into the company, these threats will unfortunately have an impact on the enterprise also. It is expected that when company users bring their connected applications from home into the workplace, this will be a ripe platform for computer worm propagation, and the spread-rate of these computer worms over this new medium will rival that of the popular worms of 2001.

The personal firewall which is in conjunction with anti-virus software will become a must-have application and help to block at least some of the computer worms and viruses which will plague the growing number of connected desktops.

4.1.2 Increasing general sophistication

Although the great majority of the older viruses were written in Assembly language that is a low-level programming language which is arcane and difficult to use, an increasing number of the latest computer worms and trojan horses are designed by using more modern (high level) programming languages and tools. These high level computer worms and trojan horses are more difficult to analyze because optimizing compilers often obscure the code logic to improve efficiency. These high level computer worms and trojan horses utilize more complicated techniques to spread and better lever the operating system and all available exploits of the target platform. Although creating a detection and cure for these threats is still relatively straightforward now, their increased complexity has dramatically increased the amount of time that it takes to fully disassemble and analyze these pathogens. This has, and will continue to put a strain on anti-virus researchers who have already had a full plate to work with.

4.1.3 Computing Infrastructure Homogeneity

The homogeneity of computing hardwares, operating systems, application softwares and communication platforms will become the single largest enabler for the epidemics of computer viruses, worms and trojan horses.

Nowadays, more than 90% of the world's computers are running Microsoft Windows operating system on Intel-based hardware[6]. An equally high percentage of computer users use standard SMTP Internet e-mail, and many large corporations are standardizing on e-mail systems such as Microsoft Outlook and IBM Lotus Notes. In the word processing field, the Microsoft Office suite enjoys a virtual monopoly for home users, business users, and government users[7]. Basically, every of our desktops has a genetically similar software and hardware makeup.

In agriculture, such a homogeneous environment is called a monoculture, and is generally known to have serious negative consequences. If farmers sow a single kind of crop on their lands (for example, in order to increase their yield of that crop), they subsequently increase the vulnerability of their entire crop to disease. If the disease affects one plant, it can rapidly and easily spread to all other genetically similar neighboring plants. Essentially, the standardization of all of the above computing technologies has created a computing monoculture which has subsequently increased our computers' vulnerability to computer-borne disease. With a growing, homogeneous set of hosts, a virus doesn't need to travel far before it finds fertile ground to launch another infection.

We have already seen thousands of macro viruses continously attacking Microsoft Office platform. Several high-profile computer worms (network-aware computer viruses) have exploited the Outlook and SMTP e-mail programs to spread themselves. Moreover, it is known that more than 99% of all computer viruses are designed to spread on the DOS/Windows/Intel platform[8].

Without doubt, software and hardware standardization have given us a huge benefit; it allows companies, government and home users to standardize their software and hardware systems, decrease troubleshooting and technical support costs, and lower replacement costs. For software developers, having a single monolithic platform reduces the costs of software development;

software developers only have to develop for a single platform instead of twelve different platforms. Also, it improves the stability of software.

These benefits have unfortunately transformed our society into one that is highly reliant on a single computing environment. While PCs are widely used as company tools at the beginning, now they pervade the most secure government systems, financial institutions, nuclear power plants, intelligence community, as well as the home. Now, the users of proprietary systems, the government and financial institutions uses the same hardware and software as ones which are found in the home. Since the great majority of government, businesses and home users use the same platform, a digital Armageddon is far from being a fairytale.

4.1.4 Ubiquitous Programmability

The ubiquitous programmability of the Windows operating system has made it possible to write computer viruses and worms without complicated programming. No one would have ever thought that the Word processor or Excel spreadsheet would be the single most successful host for computer viruses and worms. E-mail is just common Office application, however, people did not expect that computer worms could send themselves over e-mail. Unfortunately, software vendors have made the Office products the platform of choice for computer viruses,worms and trojan horses by adding robust programming abilities to the current office applications.

Users can write simple macro programs, and attach them to their Word documents and Excel spreadsheets. These programs, which are written in Visual Basic (an easy-to-use, BASIC-like programming language), can perform useful functions such as spell checking user's document, summing tables in a Excel spreadsheet, or auto-email finished expense reports to the finance department. Moreover, macros can be copied or can copy themselves to other documents. This feature allows users to easily share or install useful macros across the organization. But, the Office platform also becomes extremely vulnerable to computer virus threats by allowing macros to copy themselves from one document to another. Nowadays, more than 80% of all computer virus incidents (actual computer virus infections found by users or corporations) are because of macro viruses in Word and Excel[9].

Unfortunately, these macros not only get access to the features and components of the Office suite, but also to other components of the host computer system. Unfortunately, the marriage of the Office macro programming languages, and a second technology – the Component Object Model (COM)[10] – has had a huge impact on today's computer virus.

In a nutshell, while a programmer designs a new software application, the programmer can make the functionality of the application available to the rest of the software applications which are running on the system (and not just to the user) through the COM system. Subsequently, other programmers can design their software logic to lever the functionality which is provided by the first COM-enabled program. For instance, Microsoft Outlook enables other programs to login to the user's mailbox, inspect messages, extract attachments, enumerate the entries in the address book and send e-mail by using COM. By using this facility, a user could write an expense reporting application in BASIC or C which would make use of the Outlook e-mail program through software APIs. The programmer could program their application to use e-mail functionality of Outlook to send copies of an expense report to the finance department, without knowing anything about how to program an e-mail system, knowing e-mail protocols, etc. Obviously, COM technology has been a huge enabling technology for normal programmers. Now, the typical programmer can design extremely rich software applications by levering other components on the system.

Vendors have made it possible for Office macros to lever powerful features of COM. With this newly added functionality and the simplicity of the BASIC-like macro programming language which is supported by Office products, almost any competent user could pick up a book and develop powerful macro programs which have the capability to do far more than summing tables in a spreadsheet. These COM-enabled macros can inspect and change the entire host computer system, and even more worrisome, they can lever the built-in communication facilities of the computer to spread over the worldwide network of homogeneous computers.

4.1.5 Complete Connectivity

The increasing connectivity and enumerability of the today's communication systems allow computer worms to spread more rapidly, and to more computers than ever before. The spread rate of computer viruses was

confined to how quickly computer users exchanged infected files by e-mail, via file servers, in floppy diskettes, etc. until recently. Traditional computer viruses (which don't intentionally spread over networks) can rapidly infect many files on a single computer system however spread much more slowly from one computer system to another one due to their dependence on user behavior.

Because users share information more frequently than they share programs (at least in company and government environments), user-initiated e-mail has enabled macro viruses to spread far more rapidly than binary viruses(such as DOS and Windows viruses)[11]. But, a typical user sends just a handful of documents to a small set of co-workers during the average week. Thus, when a macro virus might rapidly be transmitted all over the world, it will only spread to a small number of users over a period of days or weeks. Then those target users must open the infected document, edit some other documents, and send them out to their co-workers. This whole human-centric process is cumbersome and limits how quickly these viruses can be spread. Luckly, anti-virus companies can respond with a computer virus siganature update and prevent any further virus spread, by the time a new macro virus can infect even a handful of users.

Unfortunately, with more computers on e-mail and the Internet than ever before, computer worms can now spread more rapidly than any traditional virus. The homogeneous, ubiquitous, COM-accessible communication mechanisms makes writing such a computer worm a piece of cake. Why should a computer virus wait to be sent by the user as an e-mail attachment while it can send itself? Why only send itself to a few of computers while it could send itself to an entire organization? The computer worm doesn't passively wait for the user to send its malicious code in an e-mail. In the contrary, it actively takes matters into its own hands. The computer worm exploits the communication components of the computer system – whether the network or e-mail - to send itself from one computer to another; therefore, it can potentially spread itself thousands of times quicker than a traditional computer virus.

Although e-mail is an ideal communication mechanism for computer worms, it is far from the only viable communication mechanism. Computer worm has begun to exploit peer-to-peer networking, and this trend is changing in the coming years. Windows 95, 98, NT and Windows 2000 support peer-to-peer networks. Users can configure Windows to permit other users on the

Windows network to get access to their files without restriction. Computer worms can rapidly find other computers on the Windows network and copy itself to these computers by exploiting this facility. The Explore.Zip worm exploited just exactly such a mechanism to spread itself over company networks, and was extremely successful. By using two distinct mechanisms, the ExploreZip Worm spread itself to other computers. First, like Melissa, ExploreZip was capable of levering Microsoft Outlook, Outlook Express and Exchange e-mail programs to send itself by e-mail. This worm sends itself to users who have recently sent e-mail to the infected user, instead of sending itself to the first 50 users like Melissa. Besides spreading itself through e-mail, ExploreZip will iterate through all computers which are visible on a peer-to-peer Microsoft network also. The worm will copy itself to all accessible computers and update a configuration file on the target computer to cause the computer to launch the ExploreZip worm during the next boot-up.

Today's computer networks are more connected than ever before. Any user or program can send an e-mail from any computer directly to any other computer in the entire network in seconds. But, a second facet of our communication systems make them even more vulnerable to virus attack: modern software directories allow the enumeration of every node which is connected to the network. For instance,corporate groupware products like Lotus Notes and Microsoft Exchange permit users to view each single e-mail user in the entire corporation, and if they like, the users can send e-mail to every one of these addresses.

In addition to groupware directories, a number of other directory sources exist that would allow a hacker or software agent to obtain a list of targets. For instance, corporate LDAP[12] directories, Internet search engines, and public mailing lists (so-called listservs) all provide the means for enumerating and targeting potentially millions of users.

The capability of users or software programs to enumerate and target specific computer systems makes computer worms quite more troubling. Firstly, a hacker could exploit these publicly available directories to choose an initial distribution list: all CIOs at fortune 500 companies, all CFOs at financial institutions, etc. Secondly, once within a corporation or government network, the computer worm can exploit the same directory mechanisms to enumerate targets and spread itself. Although the corporate e-mail directory may not be available outside of the firewall, a computer

worm can easily get access to this information and exploit it to spread, once inside a corporation. This is exactly how computer worms such as Melissa spread so quickly. It is known that, in many cases, Melissa exploited the corporate directory to spread to hundreds of thousands of mailboxes in hours[13].

4.1.6 Technology Migration to the Home

The migration of the PC from the company to the home, and the further adoption of home networking reduces the bar for computer virus development... and testing. Computer virus writers go with what computer virus writers know. It means that computer virus authors will design their threats to exploit those technologies which they have on their own computer so that they can test their creations. Therefore, those companies which employ worm-enabling technologies which are common to both the company and the home are much more vulnerable to these virus attacks.

For instance, we have already seen a number of computer viruses and worms lever Microsoft Outlook and Outlook Express to send themselves. These e-mail programs are widely used both in the company and at home, and they share the same COM programming interface. In the contrary, we have seen no worm-based attacks that lever Lotus Notes to spread themselves. Lotus Notes, unlike Microsoft Outlook, is used exclusively in companies, and is a less available technology for computer virus/worm writers to play with at home. Although we fully expect to see Lotus Notes worms in the future, the lack of a consumer-oriented Lotus Notes client has undoubtedly slowed down the development of Lotus Notes-centric threats.

When more and more companies adopt products like Outlook, Eudora and other e-mail programs, computer virus authors will have all of the components which are necessary to build and test their viral creations in their own home. Nowadays, it is not uncommon to find that home networks of several machines and all the components can be bought cheaply at the local computer store. Unfortunately, these local networks provide the computer virus author with everything that they need to develop and then test their computer worms. More than ever before, the writers of these computer worms get access to the hardware and software platforms which are employed by businesses and the government.

Also, the popular Linux platform will probably become an increasingly attractive platform for computer virus development. Because Linux is offered free of charge, source code and all, virus authors will get easy access to documentation, operating system source code, and everything that they need to design and test their viruses. Linux runs on the same computers that millions of users already own, and it is well regarded in the programming and hacking communities. In contrast to Solaris; although Solaris is a very popular UNIX platform, it is much less available to the common home user, and therefore, we expect to have fewer virus threats targeting at it. But, this is not to say that a determined attacker won't target at the Solaris platform; based on its availability to today's computer virus authors and its compatibility with existing hardware, we expect computer virus authors to target at this platform in the future.

4.2 The prediction about new types of computer viruses in the near future.

4.2.1 Wireless viruses

The threat from malicious code in the wireless world is still in its infancy. In fact, malicious code has not yet negatively impacted wireless device users. However this will soon change. In much the same way that the Internet changed the way that computer viruses, worms, and trojans were created and distributed, the wireless world represents a fertile breeding ground for hackers and e-vandals who are willing to exploit this expanding medium. When the line between mobile phones and personal digital assistants blurs, the enhanced functionality of the wireless devices which emerge offers a attrative playground for hackers and e-vandals—in much the same way that every new medium which emerged in the last two decades has offered such an opportunity. Traditional approaches to anti-virus security will not provide the necessary security.

The quick spread of wireless communications provides new chances for hackers, disgruntled employees, and others to prove their prowess in spreading computer viruses and malicious code. On the surface, the vulnerability of wireless devices to computer viruses and malicious code threats seems to follow the same patterns of vulnerabilities which the wired world has experienced. However, upon closer examination, the

vulnerabilities are more numerous and complicated. Such threats to the wireless community can be categorized into three classes:

- Application-based threats
- Content-based threats
- Mixed threats (a powerfully-packed combination of application and content-based threats which has not been yet seen in the real world)

4.2.1.1 Application-based Threats

Application-based threats are presented by executable malicious code which latches on to existing, or new, wireless applications in the wireless world. Application-based threats are potentially present anytime when a software program is downloaded to, or executed on, a wireless device, particularly when the software program is downloaded or received from an unknown source. Similarly, in the wired world, these threats are roughly analogous to the early viruses which were borne by executable programs (which were later superceded by the rise in Macro viruses—malicious code borne by non-executable files). The first malicious application-based threat which specifically targeted at the Palm operating system (OS) which is used in Palm Pilot personal digital assistants (PDAs) was called “Liberty Crack.” The free software, which could be downloaded from a Web site or accessed through Internet relay chat (IRC) rooms, pretended to convert the shareware Liberty Game Boy program into a registered version. However, in fact, when the program code was executed, the user was unaware that the program was deleting all executable applications in the handheld device in the background. Liberty Crack did not affect the underlying Palm operating system or the embedded applications.

Liberty Crack and similar “trojan horses” are probable to spread very slowly in the wild and represent a relatively low threat. Liberty Crack is designated a trojan horse because it masquerades with one purpose, while it harbors a surprise purpose (similar to the trojan horse of ancient Greece in which soldiers hid inside a hollow wooden horse presented as a gift to the trojans). Although actual incidences of Liberty Crack have not been encountered in the wild, this trojan horse is significant in its proof of concept which demonstrates that malicious code can be downloaded and may adversely impact PDAs. Many analysts have labeled Liberty Crack, which first made news in late August 2000, as a harbinger of more malwares to come. For instance, future wireless trojans could steal data such as address book information, portal passwords, and other confidential information.

This evolution and proliferation of the trojan horse presents two key aspects of application based threats. Firstly, it shows the potential for proliferation of malicious code, especially in the form of a trojan, when it is disguised as a program with perceived value that is offered for free. Secondly, this early case reminds us that operating systems in the widest use are probable to be the initial playgrounds of writers of malware. The great number of available shareware applications and the growing number of legitimate code developers in the community increases the possibility of malicious behavior. Moreover, the great number of possible affected users presents the potential profile of any malicious activity which is an enticement for those who seek the limelight for destructive activities.

Since the discovery of Liberty Crack, anti-virus researchers have been tracking a number of other application-based, potentially destructive Palm programs, which include Palm Phage—the first known virus designed to affect Palm PDAs. When it was first seen about one month after Liberty, Palm Phage infects all third-party application programs when executed. Instead of running normally, infected executable files infect other third party applications programs. Theoretically, Palm Phage can spread to other machines when the Palm is synchronized with a PC or when a Palm beams data through an infrared link to another Palm. (see Figure 1.)

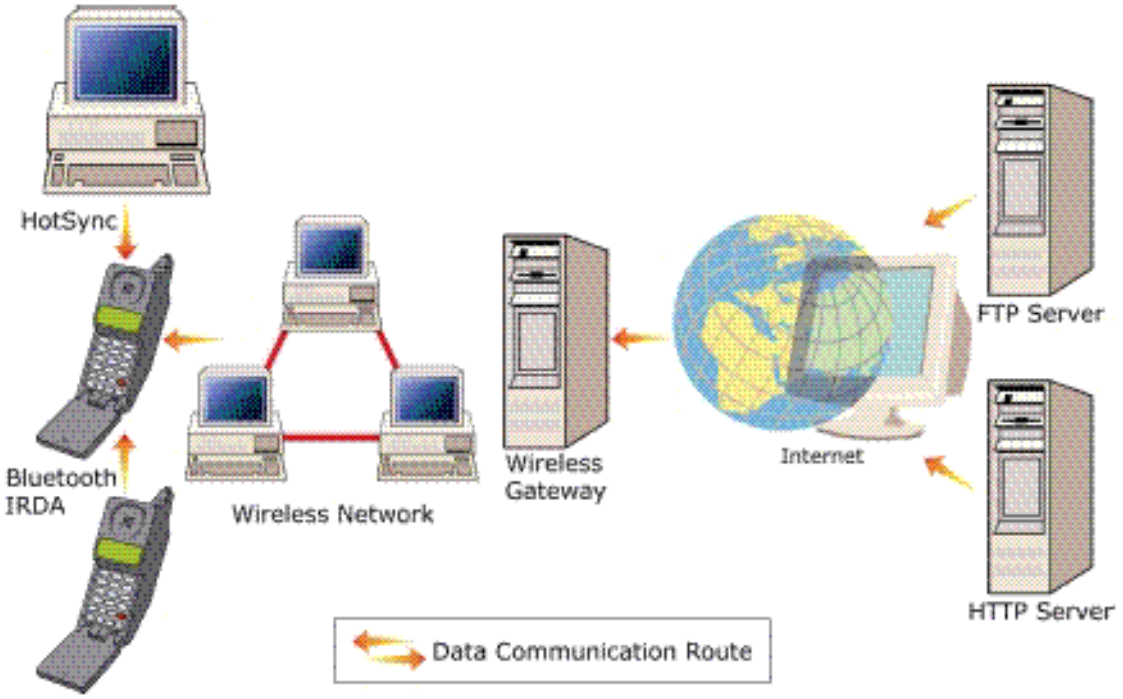

Figure 1[14]. Application-based threats may involve FTP or HTTP downloading of an executable program through the wireless gateway to the device. The virus or malicious code can also spread to other wireless devices through direct beaming via Infrared or RF (e.g. Bluetooth), or by synchronizing the device with a PC.

Simultaneously, several joke programs were observed on PDAs operating on the EPOC operating system. These programs (e.g.,EPOC_Alone.A and EPOC_Ghost.A) disturb users by sounding an alarm or flashing lights on the EPOC-enabled device. Although these programs do not spread from device to device, they show that malicious code can cause worrisome disturbances on wireless devices.

Moreover, the wireless world is viewing the regular birth of new technologies, with more on the horizon. While some of these technologies will expand the functionality of the device, others will dramatically change their connectivity with other devices (e.g., Bluetooth technology, see Figure 1). No users have lost data as a result of Palm Phage and the EPOC joke programs. However, it is demonstrated that the wireless self-replicating viruses are not only possible to develop, but easy to develop. With the expanded functionality of these wireless devices in the coming months and years, so will expand the potential for new application-based threats.

4.2.1.2 Content-Based Threats

The content (e.g., derogatory messages) is the threat, or malicious use of the content is the threat (e.g., spamming of email) in content-based threats. When email has become the primary application of the wireless world, it is one of the most vulnerable to attack also. Thus, the most common content-based threats to the wireless infrastructure happen through infected email or spam mail (see Figure 2).

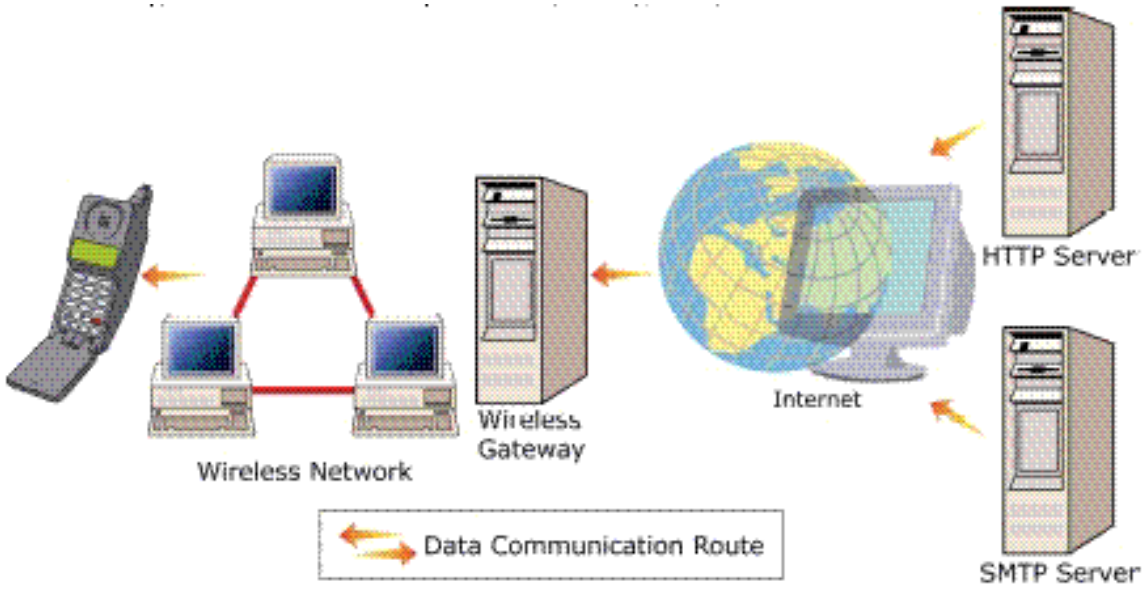

Figure 2[14]. The content threat to the wireless infrastructure involves email messages or spam that flow from SMTP or HTTP servers through wireless gateways to wireless devices.

The first content-based virus to attack wireless devices happened in June 2000 with the appearance, in the wild, of the Visual Basic Script (VBS) Timofonica on the wireless network of Madrid, Spain-based Telefonica SA. Timofonica spread by sending infected email messages from affected computers. As an infected email reached a PC, it exploited Microsoft Outlook 98 or 2000 to send a copy of itself through infected emails to all addresses in the MS Outlook Address Book. This enabled the virus to spread very quickly. In the wired world, this behavior is similar to that of the "ILoveYou" email virus that caused worldwide damage estimated as high as $700 million in May 2000.

However Timofonica was more than an email virus. For each email it sent, the virus also sent an SMS message to a randomly generated address at the

"correo.movistar.net" Internet host (see Figure 3). Because this host sends SMS messages to mobile phones which are operating on the European GSM standard (the phone number is the prefix of the email address in the message), the virus attempted to spam people with SMS messages—in this case a derogatory depiction of Spanish telco provider Telefonica Moviles.

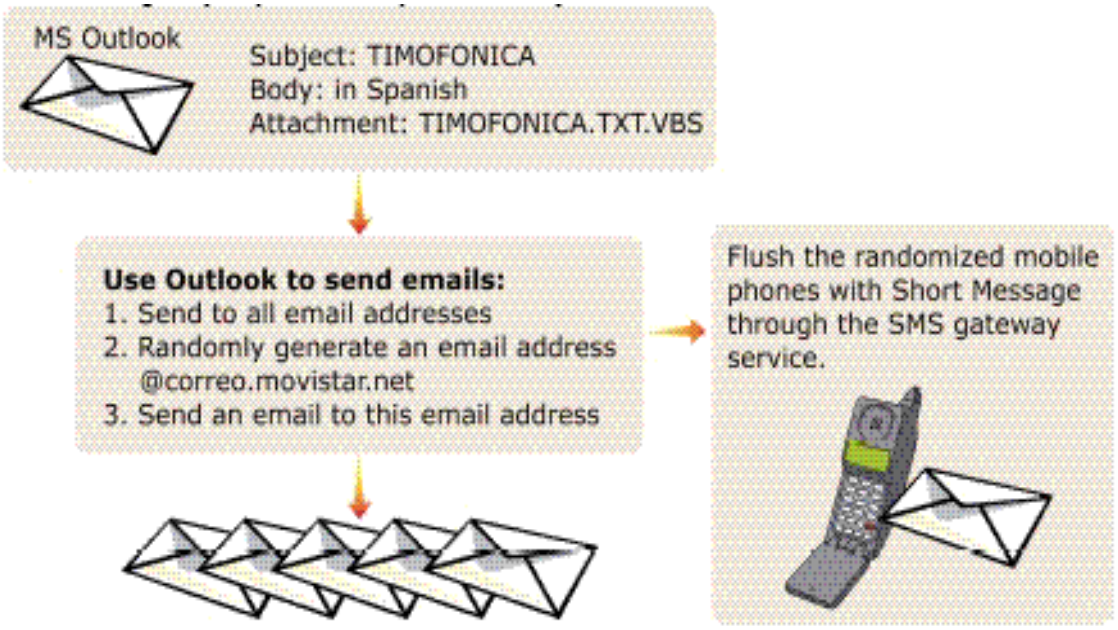

Figure 3[14]. In Madrid, the recent Timofonica virus sent infected emails to all addresses in an infected machine's MS Outlook address book and also sent an SMS message to addresses at the correo.movistar.net Internet host.

Similar to the Liberty Crack trojan, the Timofonica attack was benign and caused little damage. While the program reached out into the wireless world, it propagated through land-based PCs and emails, not from phone to phone directly. However, Timofonica showed in-the-wild, the capability of malicious code to tap into the wireless infrastructure and spread with high speed. By reducing its performance or even impairing its ability to meet load, Timofonica had the potential to flood the wireless network with messages. Even worse, receiving spam costs them money for wireless users who are billed on a per-message basis.

A similar program was observed to happen to Japan's ambitious I-mode system. Japan's largest cellular phone maker, NTT DoCoMo, developed and owns the I-mode system that appears to have successfully got both consumer and business markets for wireless device transactions, wireless Internet

access, and instant messaging in Japan. With more than 10 million users only 18 months after its launch, some analysts see I-mode as a feasible alternative to WAP that is used in Europe and touted in North America. In June 2000, a piece of malicious code started to send a particular message to wireless users on the I-mode system. While the user received the message and clicked on a hypertext link, the program dialed 110—the Japanese equivalent of 112 in Sweden—without the prior knowledge of the user. This loading of emergency service lines with useless calls showed the capability of malicious wireless code to reach out to other key infrastructures and cause serious damage.

When wireless devices become more and more sophisticated over time, another potential content-based threat which may soon enter the wireless world is the embedded script virus. Prior to the first observation of this class of viruses, viruses could be only contracted through email by double clicking on an infected email attachment. With the discovery of embedded script viruses, like the VBS_Kakworm and VBS_Bubbleboy, now wireless viruses can infect a user's system when the email is opened.

4.2.1.3 Mixed Application/Content-Based Threats

Application-based wireless threats, in which an executable program carries some malicious code, affect the receiving device. The spread of this malicious code is slow because the user must download a program with malicious code and execute the program to become infected. Content-based threats which spread relatively benign text messages or generate mobile phone calls are at the other end of the spectrum. However, these threats can spread quickly because of the nature of their propagation medium—entire address books of emails.

The third type of threat is worse than the combination of the previous two types. Although it has not yet been seen in the wild or even in the laboratory, a threat that integrates techniques from both of these previous threat types could be formidable indeed. Let's imagine a virus that involved the unwitting download of sophisticated malicious code which is attached to a shareware program that wiped out wireless device applications and propagated itself quickly across the wireless infrastructure through address books of email. Such a virus could cause damage to each device it encountered and spread across a country, or across the world, in hours. We have seen the reality of the ILoveYou virus and its destructive power.

Without adequate comprehensive wireless infrastructure virus protection, some kinds of highly destructive, quickly spreading wireless virus will inevitably surface.

4.2.2 Malware threats to Peer-to-peer networking

Peer-to-peer networking permits communication between two systems, in which every system is considered to be equal. Peer-to-peer networking is an alternative to the client-server model. In the peer-to-peer model, every system is both a server and a client, which is commonly referred to as a servent. Peer-to-peer networking has existed since the birth of computing networks. But, peer-to-peer networks recently have gained momentum with searchable peer-to-peer network file databases, increased network connectivity, and content popularity.

4.2.2.1 New Vector of Delivery

Peer-to-peer networking introduces an additional vector of delivery. The primary method of contracting a virus in the past was through a floppy disk. The floppy disk drive was the primary vector of delivery. The primary vector of delivery today is email. Malicious software is usually found as email attachments. Peer-to-peer networking presents another method of introducing malicious code to a computer. Currently, this additional vector of delivery is the greatest threat which malicious softwares have on peer-to-peer networking.

Although peer-to-peer networking systems permit user to introduce executable files to a computer, those files still request specific downloading and execution. For instance, a user of Gnutella may search for ExampleVirus and a servent may return a match of ExampleVirus.exe. In order to become infected, the Gnutella user must require a download of that file from the remote servent and execute the file. Thus, classic peer-to-peer unaware viruses could inadvertently be transmitted through a peer-to-peer network. Also, viruses could take advantage of the normal use of a peer-to-peer network. For instance, viruses could specifically try to copy themselves to or infect files in the shared peer-to-peer space.

The first discovered Gnutella worm, VBS.GWV.A, did this by copying itself to the Gnutella-shared directory as a popular file name. For instance, the

worm can copy itself into the Gnutella-shared directory as Pamela Anderson movie listing.vbs. The purpose is to trick users into downloading and executing the file.

Viruses could actually take advantage of the existing peer-to-peer network infrastructure to spread themselves. For instance, a computer worm could set up a servent on an infected system. The user with the infected system does not have to be part of the peer-to-peer network initially. After that, this servent could return the exact matches for incoming search queries, and those which download and execute the file will become infected in turn. An example of such a worm is W32.Gnuman.

4.2.2.2 Malicious Uses of Peer-to-Peer Networks

The use of peer-to-peer networks permits not only the capability for malicious software to spread, but also utilization of the protocols for communication by malicious software. In many organizations, backdoor trojan horses, such as Back.Orifice, are not effective in infiltrating an organization because of a firewall. Such programs open listening connections, while waiting for a client outside of the organization to connect. Since firewalls prevent incoming connections, except for particularly defined computers and ports, the computers remain uncompromised. But, the firewall does not normally block the peer-to-peer software, when it makes outgoing connections to the centralized directory services or other servents. Normally, outgoing connections are not blocked. Once an outgoing connection is made, the centralized directory service or a servent may pass information to the client.

The majority of current backdoor trojan horses do not make such outgoing connections since they would need to connect to a defined awaiting server. While they are discovered, this may lead to the identification of the malicious hacker. By making outgoing connections to IRC or similar centralized services, some backdoor trojan horses avoid this scenario. W32.PrettyPark is an example of a worm which creates an outgoing connection to IRC, and therefore, the common firewall configuration does not block it. Once the computer worm is connected to IRC, hackers can join the same channel and send remote access commands.

Such methods could be conducted by using a peer-to-peer network also. For instance, a malicious threat could register with the Napster centralized server

and pass a specific, unique list of files. After that, a hacker would perform a search on those specific files, and when they are matched, the hacker would be able to identify any infected systems. A request for a certain file would signal the infected machine to perform a particular task, such as taking a screen shot. By bypassing the firewall and ensuring the anonymity of the hacker, information gathering and system control of the system could then be performed in this way.

Moreover, malicious software could easily modify the configurations of existing peer-to-peer clients. For instance, a trojan horse could change the settings so that the entire hard drive could be opened for browsing and downloading, instead of a particular directory being opened for access, such as C:\MyMusic.

Because peer-to-peer malicious threats still need to reside on the system's current desktop, a scanning infrastructure can provide protection against infection. But, desktop protection may not prove to be the best approach in the future. If peer-to-peer networking become standard in home and corporate computing infrastructures, network scanning may become more desirable. Such scanning is not trivial because, by design, peer-to-peer transfer of data does not pass through a centralized server, such as an email server. Systems such as network-based IDS may prove useful, so may gateway/proxy scanning to prevent malicious threats from using peer-to-peer connections that pass inside and outside of organizations. But, peer-to-peer networking models such as Freenet will make networking scanning useless because all data is encrypted. Users will not be able to scan data which resides in the DataStore on a system. Detection of threats which are passed through Freenet type models will only be scanned on the unencrypted file at the desktop just prior to execution. The issue of encryption reinforces the necessity for desktop-based, antivirus scanning.

Although the previous threats request a virus author to create a malicious program, the simple usage of peer-to-peer connections can prove to be the greatest threat to the company. Using peer-to-peer software in computer network environments creates an unforeseen hole in computer network security. Such software easily operates within the restrictions of a configured firewall, since the software usually makes outward connections rather than depending on accepting incoming connections. It is possible that users could easily misuse or configure such software to allow outside systems to browse

and obtain files from their computers. These files can be anything from confidential data in an email inbox to proprietary design documents.

The network should not be used to transfer confidential information even though the peer-to-peer network is configured properly. Data is usually passed unencrypted along the network. Such data can easily be obtained by a network-sniffing program. Systems administrators should consider limiting the usage of peer-to-peer networks just because of privacy concerns alone.

The current peer-to-peer model appears to be moving toward a true peer-to-peer model without a centralized server, which Microsoft Networking uses nowadays. The current peer-to-peer model's advantage over Microsoft Networking is its capability to perform quick searches and exchange data through firewalls. Future models of peer-to-peer networking will combine aspects of both Microsoft Networking and Napster's protocols to permit for easy search abilities and the ability of open DataStores. For instance, users can permit for Full Control in Microsoft Networking, meaning that a remote user can not only download, but also upload and change data which is stored in the shared space. Imagine departmental groups in a corporation which need to share and update each other's files. A peer-to-peer networking model which does not require that a file be downloaded in order to be executed, and permits write-ability to remote shares will increase the capability of a malicious threat to spread.

Threats which infect network shares, such as W32.FunLove, show the difficulty of containment in environments which utilize central file servers (along with personal shares). A peer-to-peer networking model which incorporates both uploading and downloading increases the propagation and difficulty of containment of network infectors. Such a model permits simpler two-way communication of malicious threats. Virus authors may be able to update their threats through a peer-to-peer network. For instance, an infected computer can send an update to all other nearby nodes of a peer-to-peer network.

Clearly, peer-to-peer networks pose a danger as an additional vector of delivery. Their impact on security will rely on the adoption of peer-to-peer networks in standard computing environments. If systems use peer-to-peer networks just like email is used today, then they will be significant approaches of delivery of malicious code. Also, the use of two-way network communication exposes the system to potential remote control. More

importantly, the usage of a peer-to-peer network causes a hole in a firewall and may lead to the exporting of private and confidential information.

4.2.3 Combined attacks

Security exploits, which are usually used by malicious hackers, are being combined with computer viruses resulting in a very complex attack, which in some cases goes beyond the general scope of antivirus software. Such a program is an example of a class of threats which is known as “Combined Threats” – a combination of different threat types. Such viruses have the capability to spread extremely rapidly among a population of vulnerable machines, because many are capable of spreading without any user interaction whatsoever. Combined threats are defined as malware which combines the characteristics of computer viruses, worms, trojan horses, and malicious code with server and Internet vulnerabilities to initiate, transmit, and spread an attack. By utilizing multiple methods and techniques, combined threats can spread quickly and cause widespread damage. Characteristics of combined threats include the following:

(1) Causes harm: Launches a denial of service attack at a target IP address, defaces Web servers, or plants trojan horse programs for later execution.

(2) Propagates by multiple methods: Scans for vulnerabilities to compromise a system such as embedding code in html files on a server, infecting visitors to a compromised Web site, or sending unauthorized email from compromised servers with a malicious attachment.

(3) Attacks from multiple points: Injects malicious code into .exe files on a system, raises the privilege level of the guest account, creates world readable network shares, makes numerous registry modifications, and adds script code into html files.

(4) Spreads without human intervention: Continuously scans the Internet for vulnerable machines to attack.

(5) Exploits vulnerabilities: Exploits known vulnerabilities such as buffer overflows, http input validation vulnerabilities, and known default passwords to get unauthorized administrative access.

The Nimda creator seems to have learned from the characteristics of preceding computer worms and viruses, as showed by the following:

Nimda has four alternate methods of propagation.

(1) Systems which are infected with Nimda will scan the network looking for unpatched Microsoft® Internet Information Server (IIS). Then it tries to use the a specific exploit, which is called Unicode Web Traversal exploit, to gain control of the target server.

(2) Nimda can also propagate through email. It does this by collecting email addresses from any MAPI compliant email program's mailboxes. It can extract email addresses from .html and .htm files also. The worm uses these email address for the To: and the From: addresses. Therefore, the From: addresses will not be from the infected user. The worm uses its own SMTP server to send out emails. When the worm arrives by email, the worm uses a MIME exploit which allows the virus to be executed just by reading or previewing the file.

(3) Users who visit compromised Web servers will be prompted to download an .eml (Outlook Express) email file, which contains the worm as an attachment (readme.eml).

(4) Nimda attacks hard disks of systems which have enabled file sharing over the network. It will create open network shares on the infected computer also, by allowing access to the system. During this process the worm creates the guest account with Administrator privileges. This variety of propagation methods underscores the complexity of the threat and was partially responsible for the speed of its infection rate.

One of the major side effects of Nimda is that it causes localized bandwidth DoS conditions on networks with infected machines also. This is because of a combination of both the infected systems' network scanning and the additional email traffic which is generated by the worm.

From a coverage perspective, Nimda showed a follow-the-sun pattern, which appear first in the United States and then migrate to Asia and Europe. CodeRed is another example of a combined threat, since it was able to launch a DoS attack at a designated IP address (target), deface Web servers, and then, with CodeRed II, leave trojan horses behind for later execution.

The nature of CodeRed, which processes in memory rather than on a hard disk, permited it to slip by detection of some anti-virus products. Its discovery was further hampered by the fact that it presented no outward indications of its presence on the IIS server.

Simple email worms are now considered the last generation threat. Today and the near future will be consist of combined threats and their damage is still yet unseen. A downed mail server is now the least of our worries while threats can now effectively shut down Internet backbones. Hopefully, the appearance of threats such as Win32/CodeRed and Win32/Nimda has given security professionals a wake-up call to prepare for the future, since either threat could easily have been more damaging.

4.2.4 The coming threats to instant messaging

Instant messaging is an up and coming threat as a carrier for computer virus. More and more people are using instant messaging, both for personal and business reasons. Instant messaging networks provide the capability to not only transfer text messages, but also transfer files. Therefore, instant messengers can transfer worms and other malware. Also, instant messaging can provide an access point for backdoor trojan horses. Hackers can use instant messaging to get backdoor access to computers without opening a listening port, effectively bypassing desktop and perimeter firewall implementations. Moreover, finding victims doesn’t request scanning unknown IP addresses, but rather simply selecting from an updated directory of buddy lists. While more functionality are added to instant messaging, such as peer-to-peer file sharing, instant messaging will also become more prone to carrying virus.

In addition, instant messaging is very difficult to block in a company by using conventional security methods such as firewalls. Furthermore, there are normally no anti-virus applications monitoring instant messaging network communications on the server level. This means an instant messaging worm can be caught only at the desktop level.

Preventing the use of instant messaging is difficult. Simple port blocking firewalls will not be effective since clients can use common destination ports such as HTTP port 80 and FTP port 21. Most of the clients will even auto-configure themselves to use other ports than the default one if they are

unable to communicate over the default port. Firewalls with protocol analysis can prevent instant messaging clients from communicating through common destination ports, such as port 80, since instant messaging traffic is different from HTTP traffic. But, the latest versions of all the various clients embed the traffic data in an HTTP request, by bypassing protocol analysis. The client and responses essentially prepend a HTTP header to every packet sent, therefore circumventing any protocol analysis firewall. With some clients, like ICQ and AIM, HTTP headers are added only when a HTTP proxy must be used. But, AOL provides access to such a proxy for free, namely www.proxy.aol.com, and the clients auto-configure themselves to use this proxy if direct access is being blocked on all ports. Even if, in the case of AIM and ICQ, access to the proxy can be prevented by blocking the address, there are many other proxy servers which are freely available on the Internet. A simple search on the Internet will get hundreds of freely available proxy servers. Keeping up with blocking every one is difficult and an administrative nightmare.

This unfortunately makes instant messengers an open door to the computer, because the traffic will pass most server-based security measures unscanned for potential worms. Only the antivirus product running on the computer itself can catch the worms.

It is surprising that more worms and other types of malware are not exploiting instant messaging. When time progresses, we will high-likely see an increase in this area. With time, we can see more interoperability among the various networks also. AOL has already abandoned the ICQ protocol in favor of its own OSCAR protocol. We can see interoperability between AIM and ICQ soon. This interoperability can allow worms to travel among all four networks (The four most popular instant messaging networks and their default clients includes AOL Instant Messenger (AIM), ICQ, MSN Messenger (also known as Windows Messenger), and Yahoo! Messenger.) rather than being confined to a single network. Moreover, the major instant messaging networks still use proprietary protocols. Because they are all different, a worm that spreads by using MSN Messenger will not affect users of the Yahoo! Messenger service. If clients become interoperable, or users primarily utilize one network, instant messaging worms can become more widespread.

But, this does not mean one can ignore the threat that instant messaging poses. Already more than 20 computer worms can spread through instant messaging. Also, there are many exploits available for the various clients. In the future, it appears that exploits will be the predominant way that hackers attack a system. If different instant messengers become interoperable, the security track record of a vendor may decide which instant messenger a company chooses to use.

Email traffic in companies is normally monitored by antivirus software. Thus, once detection is available for a particular worm, infected emails will be stopped at the server. In the case of instant messaging, currently antivirus software does not monitor traffic at the gateway level. If a worm started to spread by using instant messaging, it could not be stopped before it reached the user's computer.

The number of computer worms for instant messaging is increasing every month, and looking at the success of some of these worms, instant messaging is clearly an up and coming platform for malicious threats. We should be careful when using instant messengers and the best way to make sure that we can use them safely is by educating users. Hopefully we will never see an outbreak of a computer worm that can spread bying using instant messengers only. While email became a part of our daily lives, it became a large carrier of computer worms also. Even after many email worm outbreaks, people are still not educated about the potential dangers of email usage. Hopefully, the same story will not repeat with instant messengers.

5 Anti-virus technology analysis

The main source for this chapter is Virus Test Center, University of Hamburg[15].

Currently, there are three main kinds of anti-virus technologies. In essence, they are scanners, monitors and integrity checkers.

5.1 Scanners

Scanners are the programs which scan the executable objects (files and boot sectors) for finding the presence of code sequences which are present in the

known computer viruses. At present, scanners are the most popular and the most widely used kind of anti-virus programs. There are some variations of the scanning technique, such as virus removal programs (programs which can "repair" the infected objects by removing the computer virus from them), resident scanners (programs which are constantly active in memory and scan each file before it is executed), virus identifiers (programs which can recognize the particular virus variant exactly by keeping some kind of map of the non-modifiable parts of the virus body and their checksums), heuristic analyzers (programs which scan for particular sequences of instructions which perform some virus-like functions), and so on. The reason which this kind of anti-virus program is so widely used today is that they are relatively easy to maintain. It is especially true for the programs that just report the infection by a known virus variant, without trying exact identification or removal. They are composed primarily of a searching engine and a database of code sequences (which are often called virus signatures or scan strings) which are present in the known computer viruses. As a new virus appears, the author of the scanner needs just to pick a good signature (which is present in every copy of the virus and at the same time is impossible to be found in any legitimate program) and to add it to the scanner's database. This may be done often very rapidly and without a detailed disassembly procedure and understanding of the particular virus.

Moreover, scanning of any new software is the only way to detect computer viruses before they get the chance to be executed. Bearing in mind that in most operating systems for personal computers the program which is executed has the full rights to access and/or change any memory location (which includes the operating system itself), it is preferable that the infected programs do not have any chance to get executed.

Finally, even though the computer is protected by another (not virus-specific) defence, a scanner will still be needed. The reason is that when the non virus-specific defence detects a virus-like behavior, the user normally wants to identify the particular virus, which is attacking the system - for example, to figure out the possible side-effects or intentional damage, or at least to identify all infected objects.

Unfortunately, the scanners also have several very serious disadvatages.

The primary one is that they must be constantly kept up-to-date. Because they can detect only the known viruses, any new virus presents a danger, as

the new virus can bypass a scanner-only based protection. Acturally, an old scanner is worse than no protection at all - because it provides a false sense of security.

At the same time, it is very difficult to keep a scanner up-to-date. The author of the scanner must obtain a sample of the virus, disassemble it, understand it, pick a good scan string which is characteristic for this virus and is impossible to cause a false positive alert, incorporate this string in the scanner, and ship the update to the users, in order to produce an update, which can detect a particular new virus. It can take quite lots of time. While new computer viruses are created every day - with a current rate of up to 100 per month[16], very few anti-virus producers are able to keep up-to-date with such a production rate. One can even say that the scanners are somehow responsible for the existence of so many virus variants. Acturally, because it is so easy to modify a virus in order to avoid a particular scanner, lots of "wannabe" virus authors are doing so.

But, the fact that the scanners are obsolete as a single line of defence against the computer viruses, became clear only with the appearance of the polymorphic viruses. They are viruses that use a variable encryption scheme to encode their body and even change the small decryption routine, so that the virus looks differently in every infected file. It is impossible to pick a simple sequence of bytes which will be present in all infected files and choose it as a scan string. Simply such a sequence does not exist. By using a wildcard scan string, some polymorphic viruses can be detected, however, more and more viruses appear today, which can’t be detected even though the scan string is permited to have wildcard bytes. The only possible way to detect such viruses is understanding their mutation engine in details. So an anti-virus researcher has to make an algorithmic "scanning engine" which is specific to the particular virus. But, this is a very time-consuming and effort-expensive task, therefore, many of the existing scanners have problems with the polymorphic viruses. We will see more such viruses in the future.

The last disadvantage of the scanners is that scanning for lots of viruses can be very time-consuming. The number of currently existing distinct computer virus, worm and trojan horse strains is about 56,000[17] and is exponentially growing. Despite the scanning methods used, scanning is not cost-effective in the long run.

Estimates of the actual number of computer viruses varies, and is dependant on how the various strains (variations) are counted. NCSA estimated that there are more than 4000 different viruses now 'in the wild'. About 100's of variations of all types of malware appear every day[18]. Concept viruses (Demonstrated attacks which have been not yet found in the wild) push this number even higher.

5.2 Monitors

The monitoring programs are memory resident programs that constantly monitor some functions of the operating system. Those functions are considered to be dangerous and indicative for virus-like behavior. Such functions include changing an executable file, direct access of the disk by bypassing the operating system, and so on. So, when a program attempts to use such a function, the monitoring program intercepts it and either denies it completely or asks the user for confirmation.

The monitors are not virus-specific and thus need not to be constantly updated, unlike the scanners. Unfortunately, they have other very serious disadvantages, which make them even weaker than the scanners as an anti-virus defence and almost unusable nowadays.

The most serious disadvantage of the monitors is that they can be easily bypassed by the so-called tunneling viruses. The reason is the complete lack of memory protection in most operating systems for personal computers. Any program which is being executed (including the virus) has full access to read and/or modify any area of the computer's memory, which includes the parts of the operating system. Thus, any monitoring program may be disabled since the virus could simply patch it in the memory. There are other clever techniques such as interrupt tracing, DOS scanning, and so on, which permit the computer viruses to locate the original handlers of any operating system function. Then, this function can be called directly, therefore bypassing any monitoring programs, which be supposed to watch for it.

Another disadvantage of the monitoring programs is that they try to detect a computer virus by its behavior. In essence, this is impossible in the general case, as proven in Ref. [19]. Thus, they cause many false alarms, because the functions which are expected to be used by the computer viruses usually have pretty legitimate use by the normal programs. If the user becomes used

to the false alerts, it is possible that the user will oversee a real one. Also, the monitoring programs are completely useless against the slow viruses.

5.3 Integrity checking programs

According to Dr. Fred Cohen's definition[20], a computer virus is a program that can `infect' other programs by modifying them to include a possibly evolved copy of itself. Thus, in order to be a computer virus, a program must be able to infect others.The program must cause modifications to the programs which get infected, in order to infect. Thus, a program, which can detect that the other executable objects have been modified, will be able to detect the infection. Normally, such programs are called integrity checkers.

The integrity checkers compute some types of checksum of the executable code in a computer system and keep it in a database. Moreover, the checksums are periodically re-computed and compared with the stored originals. A few of authors point out that the checksums must be cryptographically strong in order to avoid forging attempts from the part of the computer virus. By using some types of trap-door one-way function that is difficult algorithmically to be inverted, this can be achieved. The examples of such functions are DES, MD4, MD5, and so on. However, this is not compulsory, as has been shown by Yisrael Radai[21]. If it is correctly implemented, a simple CRC is sufficient.

Currently, there are several kinds of integrity checkers. The most widely used ones are the off-line integrity checkers that are run to check the integrity of all the executable codes in a computer system. Another one are the integrity modules that may be attached to the executable files with the help of a special program, so that when started the latter will check their own integrity. This is not a good idea, unfortunately, because not all executable objects can be "immunized" in this manner. In addition, the "immunization" itself can be bypassed easily by stealth viruses. The third kind of integrity software are the integrity shells that are resident programs, similar to the resident scanners, which check the integrity of an object only at the moment when this object is going to be executed. These are the least widespread anti-virus programs nowadays, however, the anti-virus researchers predict them a bright future[22].

The integrity checking programs are not virus-specific and thus do not need constantly to be updated like the scanners. They do not attempt to block computer virus replication attempts like the monitoring programs and thus cannot be bypassed by the tunneling viruses. Acturally, as showed by Fred Cohen[23],currently, they are the most cost-effective and sound line of defence against the computer viruses attacks.

Also, integrity checking programs have some disadvantages. For examples, they can't prevent an infection. They are only able to detect and report it after the fact. Secondly, they must be installed on a virus-free system, otherwise they will compute and store the checksums of already infected objects. Thus, they must be used in a combination with a scanner at least before installation. In order to make sure that the system they are being installed on is virus-free, this is absolutely needed. Thirdly, they are prone to false positive alerts. Because they only detect changes, not computer viruses, any change in the programs such as updating the software with a new version is possible to trigger the alert. By using some intelligent heuristics and educating the users, this can be avoided or at least decreased sometimes. Fourthly, normally the integrity checkers can't determine the initially infected object (i.e., the source of the infection), while they are able to detect the computer virus spread and identify the newly infected objects. Fifthly, integrity checkers can't detect slow viruses also. A typical slow virus can infect an executable file when it is copied to a new directory. The slow virus would leave the source file intact; however, it would infect the target file when it is written to disk. It doesn't reside in the integrity checker's database because the target file was just created.

Despite the disadvantages described above, the integrity checking programs are the currently most powerful line of defence against computer viruses and other malwares attacks, and are highly likely to be used more widely in the future.

6 New promising trends in computer virus defence

In the new world of Internet-born viruses, computer viruses can become very widespread shortly after their first infection and in some cases before a cure is available. Virus incidents in the future will spread more than what the Internet Worm and the Melissa virus did. Every new virus will have the potential to rage out of control, if a cure is not made available rapidly and

distributed widely. Even worse, nothing can prevent viruses from being written at a much faster pace than what they are today. Acturally, it is easy to imagine viruses which are written at a quick enough pace that even a dedicated effort to hire and train new human virus analyzers could not keep up with.

Taken together, these two trends draw a disturbing picture: More new viruses than humans can handle, spreading more rapidly than humans can respond. Whatever is done to solve this problem, it will look quite different from the current solution.

In order to solve this problem, IBM has built a pilot computer immune system which can find, analyze and cure previously unknown viruses faster than the viruses themselves can spread. The system solves several important problems. While rapidity of response requests the entire system to be capable of automated operation, customer administrators can control which parts of their system are automated and which parts request manual intervention. A novel active network architecture allows the system to handle a great number of customer submissions rapidly, thus the system is able to handle floods because of an epidemic of a fast-spreading virus. With greater speed and precision than what human analysts can, a virus analysis center is able to analyze most viruses automatically. Both the active network and the analysis center are scaleable, thus the system can accommodate easily ever-increasing loads. End-to-end security of the system permits the safe submission of virus samples and makes sure authentication of new virus definitions.

The main source for this whole chapter is IBM Thomas J. Watson Research Center[24] .

6.1 Immune System Architectural Overview

In order to cure a new virus and be faster than virus spreads, IBM has built a pilot immune system for the world's computers. Similar to the biological immune system, it defends the "body" of computers against viruses which are seen once by any of them. It is able to find, analyze, and create a cure for a new, previously unknown virus, then make that cure available to all of the computers. It is able to do this completely automatically, and quite quick, most importantly, quicker than the virus itself can spread.

In order to see how this immune system works, let’s now step through an example(see Figure 4) of detecting a virus at a client system, sending a sample of the virus to a local administrator, transporting it to a virus analysis center, analyzing it, and distributing the cure. In the example, all of the steps can be done automatically, without human at any of the computers involved.

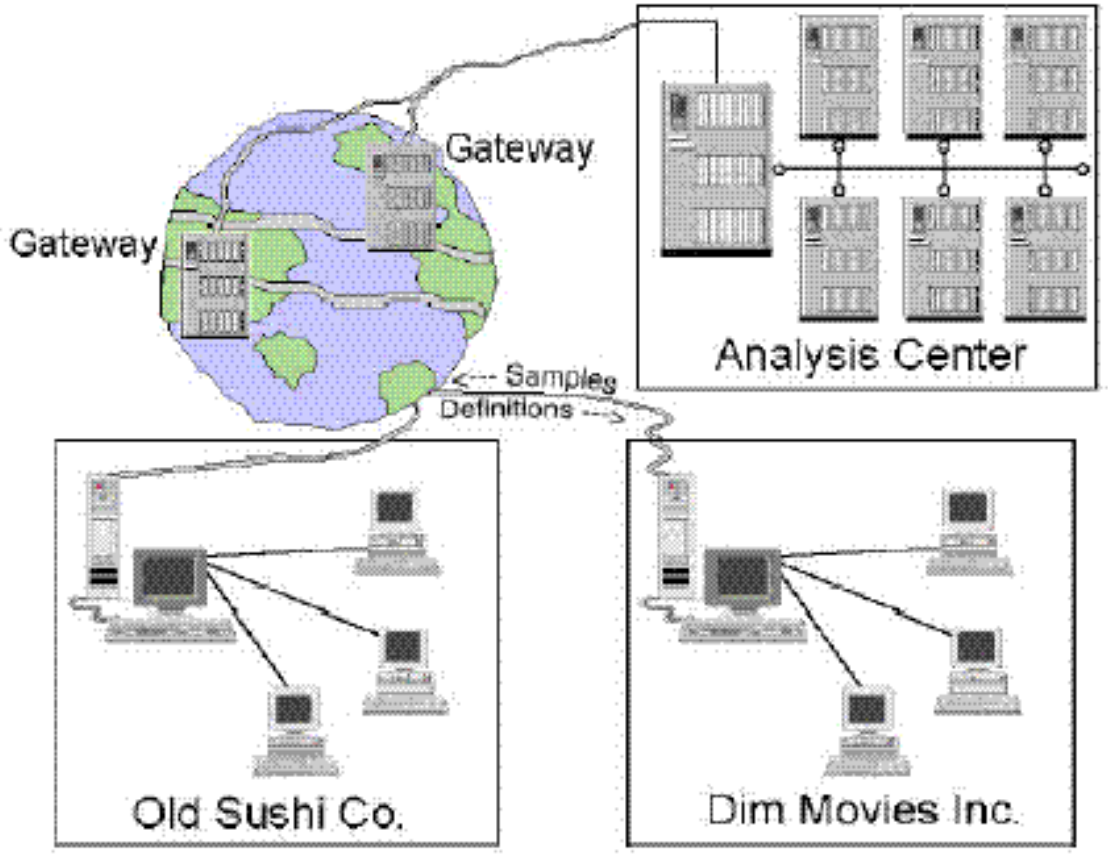

Figure 4[24]: Overview of the Immune System. A new, previously unknown virus is found in a client in one organization. A sample of the virus is transported through the organization's administrator, to the immune system's active network, where it travels through a hierarchy of gateways. If it cannot be handled directly by the gateways, it reaches the analysis center, where it is analyzed and a cure is prepared. The cure is distributed to the infected organization and made available to others who have not yet encountered the virus. The entire process can be done automatically.

6.1.1 Virus Detection

A possible new virus is detected on a client system. The client can’t decide whether the file or other object is actually infected, however the heuristic or

signature detection raised enough suspicion. Further analysis is necessary. A sample of the suspicious object is extracted, packaged in a harmless way, and sent to an anti-virus administrator system over the organization's internal network.

6.1.2 Administrator System

The administrator system allows control and auditing of what leaves and enters the organization's internal network through the immune system. If the administrator system can't handle the sample, it can forward the sample higher in the immune system hierarchy for analysis. Also, the administrative system keeps track of the status of various samples - waiting to be submitted, submitted but not yet analyzed, analysis complete and updated virus definitions ready, etc. All of these functions can be implemented automatically in order to make sure quick response to a new virus. Also, the administrator can configure the system to request human intervention and choice in deciding whether files need to be stripped, in prioritizing samples for submission or in submitting the samples themselves.

6.1.3 Active Network

An active network processes samples and transports samples via the Internet for potential analysis by a central virus analysis center, once samples are submitted from the administrator system. This active network is constructed to handle epidemics or floods by dealing with as many submitted samples as possible in the network. So it leaves the analysis center to focus on a single copy of a new virus rather than its many siblings. Standard Internet transport and security protocols are used to make sure reliable and safe transmission.

6.1.4 Virus Analysis

The virus analysis center analyzes the virus sample, uses the results of this analysis to create and test a cure for the new virus, and packages that cure as a virus definition update that can be distributed to users.

6.1.5 Cure Distribution

The virus definition update must be returned to the client which reported the initial infection once the virus analysis center has created a cure in the form of a virus definition update. Also, it must be made available to other systems in the reporting organization and other systems around the world, so that they can be protected from the virus before the virus spreads to them.

6.2 An Active Network to Handle Epidemics and Floods

6.2.1 Overview

The role of the active network is twofold. In the case of average loads, it supplies a safe, reliable means of transporting virus samples from a customer to the virus analysis center, and transporting the resulting new virus definitions back to the customer. In the case of peak loads like epidemics and floods, it has the critical responsibility of dealing with potentially huge volumes of traffic both ways without clogging up the analysis center with demands to analyze the same virus (or the same clean file) over and over again. In nature, the virus analysis center implements very computationally intensive tasks and can’t feasibly keep up with the millions of potential files which the immune system can receive during an epidemic or flood. The active network must intermediate between these demands and the analysis center(see Figure 5).

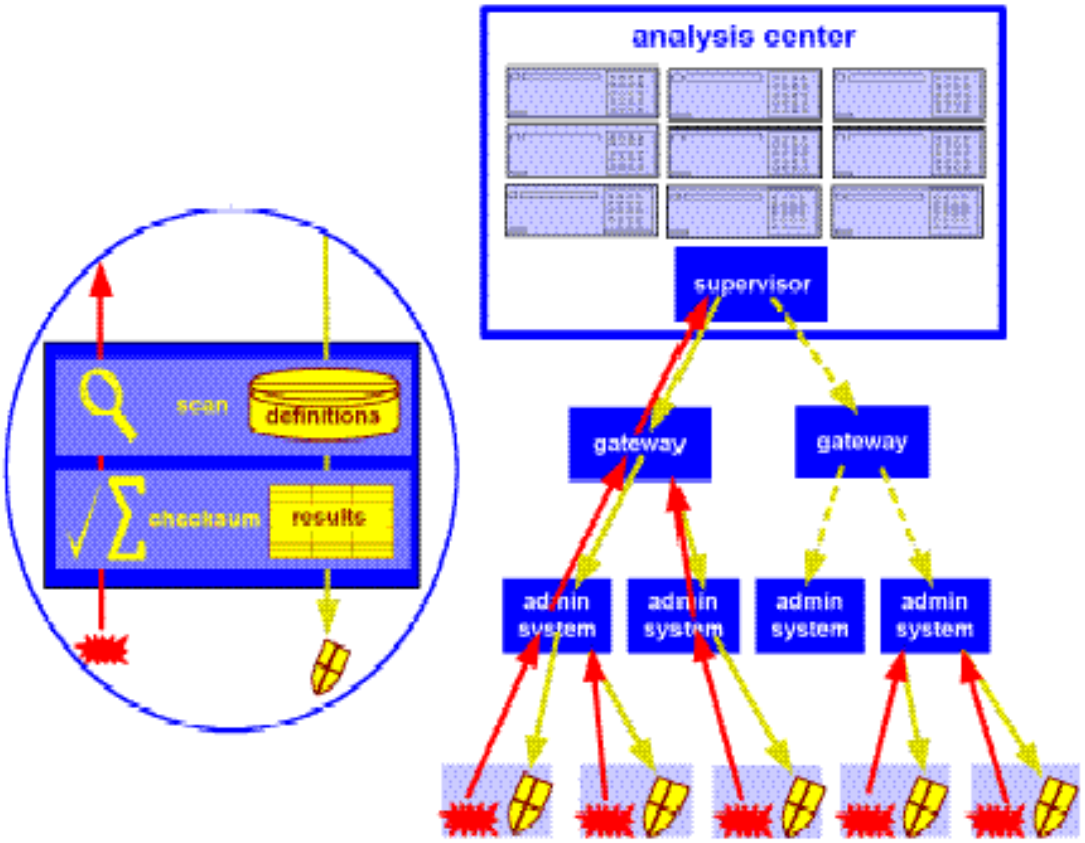

Figure 5[24]: The Active Network. Administrator systems, which send virus samples to the immune system, form the leaves of the active network. Samples travel through a hierarchy of filters, which handle the sample if it has already been analyzed as uninfected or as a known infected file. Otherwise, they forward it to the analysis center for analysis, resulting in updated virus definitions which are distributed downward to the gateways, to the administrator systems, and ultimately to the clients.

6.2.2 Safety and Reliability

A system which is intended to deal with virus emergencies must be reliable, especially in an emergency, and must not expose customers to risks like disclosure of their sensitive information or the delivery of a forged virus definition file from an unscrupulous source. In order to meet the objective of reliability, a system must have a transaction protocol which guarantees delivery of the sample to the appropriate gateway or analysis center, makes sure an appropriate response is generated, and guarantees delivery of the updated virus definitions (or other response) back to the administrator system. In order to meet the objective of security, a virus protection system

must encrypt the virus sample, virus definition files and any information which are sent along with them, to prevent disclosure of potentially sensitive customer information. IBM has created special-purpose transactions for use in the active network. These transactions send samples up, and send back status information and virus definition files(see Figure 6).

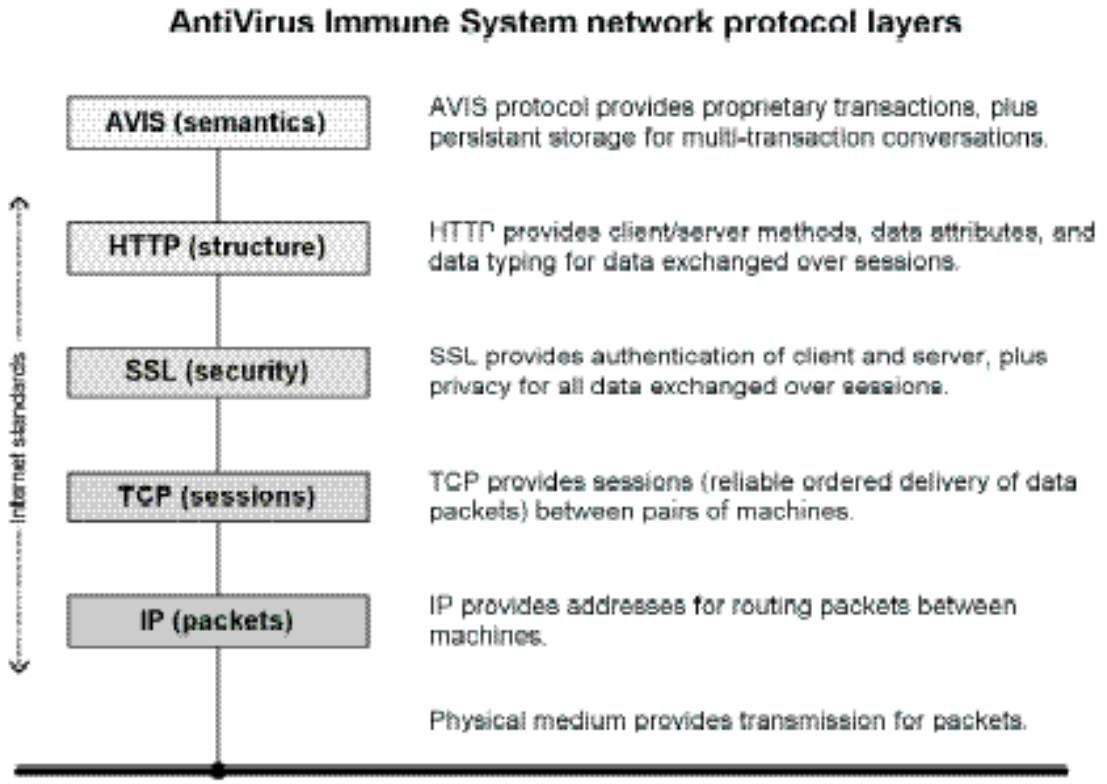

Figure 6[24]: The Active Network Protocol Stack. The special-purpose transaction protocols that implement the active network are built on top of international standards for structured data, reliable transport, and secure communications.

6.2.3 Scaling the Active Network

The active network is constructed to easily scale up to larger transaction volumes when the nature of the virus problem changes. Additional gateways can be added to the tree, and the gateways around them are reconfigured to understand the addition of the new gateways. Nothing else needs to be changed.

6.3 Automated Virus Analysis Center

6.3.1 Overview

The analysis center determines whether the sample contains a virus by actually getting the virus to spread. The analysis center analyzes the virus and makes a virus definition update which can detect, verify and disinfect the virus, if it does contain a virus. This virus definition file is tested to make sure it works correctly on all available sample of the virus. A problem in any phase of this process results in the virus sample to be sent to human analysts for processing. Then, the virus definition file is sent out through the active network to all organizations which submitted samples of this virus, once it finishs testing. As shown in Figure 7, the analysis center is composed of a network of computers, which are isolated from the rest of the world by a firewall for security purposes. A supervisor system is in charge of coordinating all activities inside the analysis center.

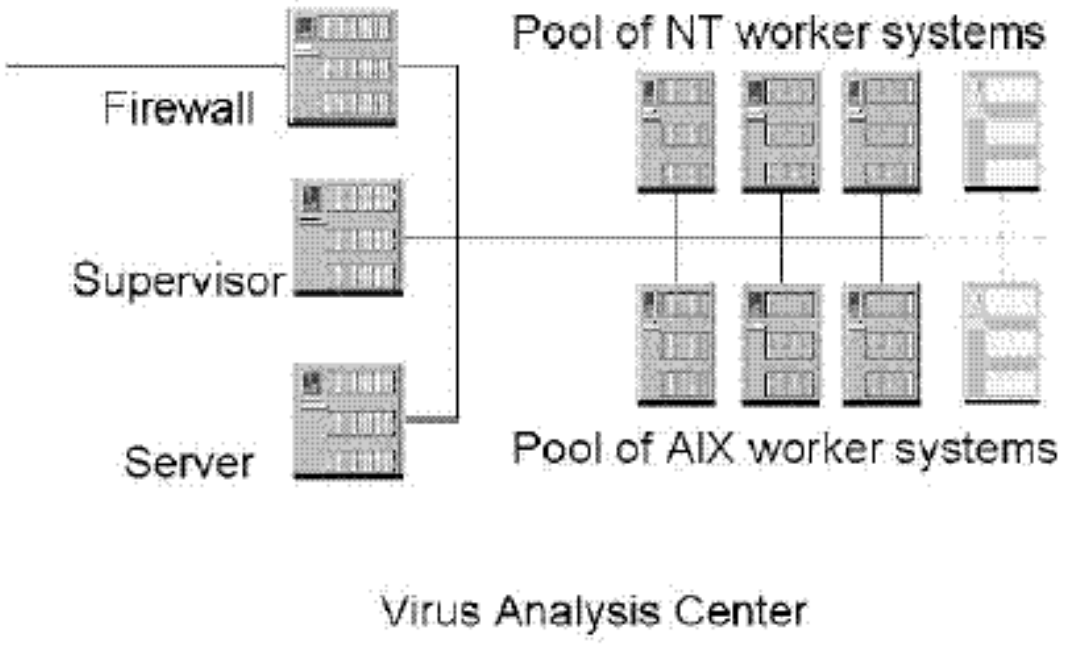

Figure 7[24]: The Virus Analysis Center. Samples come into the virus analysis center from the active network through a firewall, which isolates the virus analysis center from the rest of the Net. Samples are queued for processing under control of a supervisor system, which tracks priorities and status, assigning tasks to pools of worker systems, until analysis is complete and updated virus definition files are returned to the active network through the firewall. A server stores virus replication environments and contains an archive of everything done in the analysis center. The pools of worker systems can be expanded dynamically to scale the analysis center to larger workloads.

6.3.2 The Supervisor

A supervisor system oversees the flow of samples via the system. It's responsibility is to keep track of what worker machines are available, to parcel out work to them, to notice when an assigned task is finished, and to notice whether something goes wrong during a task and intervention is needed.

6.3.3 Integration with Back Office Systems

The virus analysis center is integrated with back office systems which track customer incidents, produce new virus definitions, and maintain a database of virus definitions.

6.3.4 Virus Analysis Tasks

As showed in Figure 8, a sample goes through a number of steps in order to determine whether it is infected and to analyze whatever virus mayt be present. The type of virus that it may contain is firstly classified, then the virus is replicated enough times in order for analysis to be reliable. The virus is analyzed, and information to detect, verify and disinfect the virus is extracted. The information is used to create a virus definition, then, the virus definition is tested against all of the samples of the virus. The updated virus definitions are returned if all of these steps are successful.

These processes are every carried out as modular, isolated tasks. The supervisor can dispatch any of them to any number of worker systems at any time. Several viruses can be analyzed at the same time and several machines can be devoted to the analysis of a single, difficult virus.

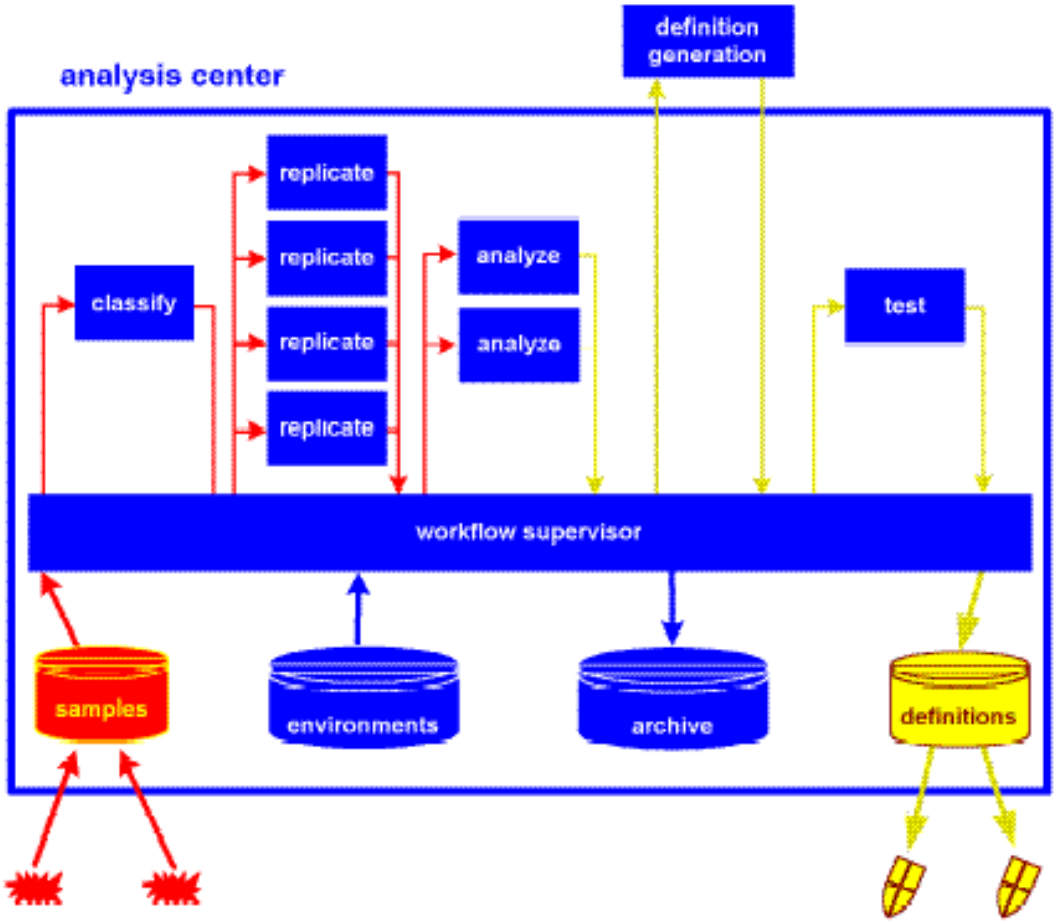

Figure 8[24]: Processes within the Analysis Center. The supervisor directs virus samples through the various processing. Computationally intensive stages, such as replication, can be done in parallel to complete the process more quickly. The fail-safe design of the analysis center defers problematic samples, at any stage, to a human analyst.

6.3.4.1 Classification

The first step in analyzing a virus is to determine what type of virus it is, so that specialized type-specific routines can be used. For Microsoft Word files, currently, the classification task identifies the version of Word and determines the language of the file (English, French, etc.). For Microsoft Excel files, it determines the version of Excel. For DOS file viruses, it determines whether they are COM or EXE files. In order to make sure reliability, this classification is implemented by examining the structure of the file, rather than by looking at the filetype.

6.3.4.2 Creation of the replication environment

Once the sample has been classified, it must be replicated, both to determine that it is acturally a virus, and to create enough samples that it can be analyzed reliably. The first step in replication is to set up a virtual environment where the virus is likely to replicate. When an appropriate replication environment has been set up, an image of that environment is obtained from the server and installed on one or more worker machines of the proper type.

6.3.4.3 Replication

Now, replication tasks are dispatched to the worker machines whose replication environments were set up in the previous step. Replication tasks run the virus in the emulated environment, trying to make it infect "goat" files which are uninfected files of known structure put there for exactly this purpose. In order to try to infect the "goat" files, the system emulates the actions which an expert human virus analyst would try. Key sequences are input to the emulated machine to simulate a user typing. The purpose of replication is to obtain enough virus samples to allow analysis to be done reliably.

6.3.4.4 Analysis

The virus can be analyzed once enough samples have been replicated. In fact, some of this analysis has already been done as part of the replication task, because it had to know enough about the virus to determine whether it had replicated and that there were a sufficient number of good samples. If several different forms of a virus have been generated (e.g. upconversions of macro viruses), each form is analyzed separately and can result in an additional virus definitions. Completing the analysis involves activities such as extracting a good signature string for the virus, constructing a map of all of its regions for verification, and creating disinfection information.

6.3.4.5 Definition generation

The definition generation task starts with the virus definition source which is produced by the analysis task and creates a complete set of virus definition files, which include definitions for all viruses to date. Also, human analysts

can create updated virus definitions as a result of their manual analysis of viruses. In order to make sure consistency between definitions generated by humans and those generated automatically, and to make sure regularity of the sequence numbering of virus definitions, both humans and the analysis center use a single definition generation system. When a new set of definitions is to be generated, the definition generation system is locked by either the human or the analysis center, a new sequence number is created, the definition is generated and tested, then, the definition generation system is unlocked for subsequent use.

6.3.4.6 Test

Once an updated virus definition file is available, the test task uses these definitions to make sure that all of the samples can be detected, and that all of the goat files can be returned to their original form by disinfecting them. The virus definition must properly detect, verify and disinfect all files. No exceptions are allowed. A virus definition file is packaged up and sent out by the supervisor system to the active network as a solution to the submitted virus once a virus definition file has passed test.

6.3.5 Deferring Problematic Samples

The inability to process a sample can occur since it contains a new or complex type of virus which the analysis center cannot currently deal with. Also, the sample might be uninfected, in which case it will not replicate. In the former case, a human analyst will have to examine the virus, create a virus definition update, and help understand how to enhance the analysis center to handle such viruses in the future. In the latter case, a human can verify no virus is present, and update the active network so it recognizes this file as uninfected if seen again.

6.3.6 Safety and Reliability

The analysis center has been designed to operate 24 hours a day. It produces safe, reliable virus definitions. It is fault tolerant. It is transaction based. It is isolated from outside interference by a strict firewall. Viruses are stored in non-executable form whenever possible. They are only executed on virtual machines from which they cannot escape. Detection signatures are extracted

so as to make sure extremely low false positive rates. Full verification information is added to virus definitions, so that disinfection is only tried if a virus exactly matches the one which is analyzed. This eliminates the risk that a (rare) false positive could lead to an improper attempt to disinfect. Before virus definitions are released, virus definitions undergo rigorous testing. Finally, the sample is deferred to human analysts if a problem is encountered in any phase of the analysis. This fail-safe policy makes sure that the analysis center only produces dependable definitions.

6.3.7 Scaling the Analysis Center

The design of the analysis center predicts the necessity of responding to an increased load on the system in the future. Individual virus incidents are dealed with in parallel, and every individual processing step can be done in parallel also. Due to this parallelism, simply reconfiguring the analysis center with more worker machines increases the rate at which an individual virus can be analyzed, and the overall throughput of the analysis center. Adding new worker machines to the analysis center can be done dynamically, without shuting the system down. When a new worker machine is added, the supervisor notices the new resource, adds it to its resource pool, and starts using it immediately. Doubling the number of worker machines roughly doubles the overall throughput of the analysis center, up to the point where definition generation is a bottleneck.

6.4 How does the Immune System Handle Loads

It is suggested that a system which is designed to respond to virus emergencies must handle average loads flawlessly. It must handle very heavy peak loads without denying service to any customer and should, at worst, inflict only slight delays. The system should deal with it gracefully, without requiring any action on the part of customers and, at worst, recover and continue to process requests when any backlog of requests is cleared up, if something cause an overload condition.

6.4.1 Average Loads

In average conditions, the immune system is designed to easily handle the load of new viruses, and demand for virus definition updates to deal with

submitted samples. The active network and analysis center are designed to be robust, fault-tolerant systems which operate 24 hours a day.

6.4.2 Peak Loads

In peak loads, the immune system continues to operate as usual, however concentrate its efforts on urgent customer incidents.

6.4.3 Overload

The immune system is constructed so that overload is a rare exception. Although it could happen during an extremely wide epidemic of a very fast-spreading virus, it is more likely to occur due to a network outage or other failure. The only effect of overload is increased delay in transporting the samples. The same is true on the downward path, when updated virus definition files are sent to customers. No samples are lost, service is never denied to any customer, and customers are not required to intervene to make sure that their samples are processed and their updated virus definitions are returned.

6.5 Current Capabilities and Performance

6.5.1 Active Network

Although the active network is capable of being fully hierarchical, the pilot immune system uses a single gateway as the contact point for all pilot customers. This is consistent with the very small number of pilot customers and the expected peak loads on the pilot system. It is estimated that the active network in the pilot immune system is capable of supporting an upload rate of 100,000 virus samples per day, and a download rate of approximately 10,000 virus definitions per day. The gateway's database of results from previously analyzed samples, which it uses to handle any future submission of these same samples, will hold 10 million results in 1 GB of disk[24].

6.5.2 Analysis Center

In the pilot configuration, the analysis center is equipped with three AIX worker machines (which are used primarily for macro virus replication) and three NT worker machines (which are used for all other analysis tasks). This already provides substantial benefits from parallelism. Additional worker machines can be added dynamically. Currently, the input queue in the analysis center is capable of holding approximately 8,000 samples which are awaiting analysis. It can be expanded easily by increasing the total disk space on the supervisor system.

6.5.2.1 Macro Viruses

Currently, the analysis center can analyze Microsoft Word and Microsoft Excel macro viruses in Office 95, Office 97 and Office 2000 formats. It can handle Microsoft Word documents which are in any of ten languages: English, French, German, Italian, Spanish, Polish, Dutch, Brazilian Portuguese, Japanese and Traditional Chinese. A separate replication environment is used for each format and language, to make sure that viruses in these formats and languages execute and spread properly in the virtual machines. It means that the analysis center can successfully replicate and analyze viruses which are specific to any of these versions of Microsoft Word or Excel, and specific to any of these languages. In tests to date, the analysis center analyzes and produces complete definitions for over 80% of the macro viruses which are in the wild. It can typically complete analysis of a single macro virus from beginning to end in 30 minutes in its current configuration if the analysis center is working on only a single macro virus[24]. The analysis center can complete analysis of four viruses per hour on average if many macro viruses are queued up for analysis simultaneously, so the worker machines are used most efficiently. Although the increase is not linear, when the number of worker machines is increased, the turnaround time will continue to decrease and the throughput will continue to increase.

6.5.2.2 DOS File Viruses

DOS file viruses are replicated in a virtual DOS machine under Windows NT. Although a variety of DOS environments could be tried on a given virus, the current system uses only one virtual DOS environment. In tests to date, the analysis center replicates over 80% of the DOS file viruses which

are in the wild, although complete definitions are produced for only about 50% of the viruses in the wild. It can typically complete analysis of a single DOS file virus from beginning to end in 20 minutes if the analysis center is working on only a single DOS file virus[24]. The analysis center can complete analysis of seven viruses per hour on average if many DOS file viruses are queued up for analysis at the same time, so the worker machines are used most efficiently. Although not linearly, increasing the number of worker machines will continue to increase the throughput. It will not have a significant effect on the turnaround time.

Solving the problem of epidemics of fast-spreading viruses requests a very different approach than the anti-virus industry has taken historically. The computer immune system which IBM has developed solves this problem, and does so safely and reliably, so it can be used by real customer organizations in day-to-day operation.

7 Conclusion

It is expected that the virus problem will continue to evolve, just as it has for the past decade or so, and sometimes in unexpected directions. The explosive growth of the Internet and the rapid emergence of applications that disregard the traditional boundaries between computers threaten to increase the global spread rate of computer viruses by several orders of magnitude. The nature of computer viruses and their ability to propagate is on the cusp of a fundamental, qualitative change -- one that demands an equally fundamental change in the way we must defend against them. The new promising immune system is likely to be an important tool to control their spread for the foreseeable future.

Home users, corporations and government entities need to seriously reconsider their security policies, and anti-virus companies need to start working on the next generation of anti-virus protection. The best computer virus defence strategy is that computer users adopt a healthier system 'life style' as well as using the best of 'modern medicine' to avoid catching a virus!

The game continues.

8 Glossary

Bluetooth: Bluetooth wireless technology is a worldwide specification for a small-form factor, low-cost radio solution that provides links between mobile computers, mobile phones, other portable handheld devices, and connectivity to the Internet.

COM: The Component Object Model (COM) is a software architecture that allows applications to be built from binary software components. COM is the underlying architecture that forms the foundation for higher-level software services, like those provided by OLE. OLE services span various aspects of commonly needed system functionality, including compound documents, custom controls, interapplication scripting, data transfer, and other software interactions.

DSL: Digital Subscriber Line(DSL). DSL is based on the fact that a telephone line, between a telephone exchange and a relatively nearby user, has a much higher bandwidth than the approximately 4 kHz that is used for phone calls. The bandwidth may be as large as some MHz. This technology allows for both a phonecall and high rate data communication simultaneously on the same telephone line. The two uses of the telephone line are in both ends separated by filters. The data communication is in a higher frequency band than the phone call. There are several variations of DSL, e.g. ADSL, HDSL and VDSL.

EPOC: EPOC is the first truly communication centric operating system for mobile information platforms. It is not constrained by being a down sized PC operating system or a 16 bit architecture. It harnesses the full potential of object-orientation to deliver a powerful set of user and developer tools from a small ROM footprint. Its three tier architecture means it can be delivered on a range of chip sets and allow device manufacturers to deliver unique and ergonomic interfaces.

LDAP: The Lightweight Directory Access Protocol (LDAP) is a directory service protocol that runs on a layer above the TCP/IP stack, and provides a mechanism for connecting to, searching, and modifying Internet directories. The LDAP directory service is based on a client-server model. The function of LDAP is to allow access to an existing directory. The data model (data and namespace) of LDAP is similar to that of the X.500 OSI directory service, but with lower resource requirements due to its streamlined features. The associated LDAP API simplifies writing Internet directory service applications.

Lotus Notes: Lotus Notes is the integrated e-mail and e-business software for the Internet and corporate intranets. An intuitive, Web-inspired environment, Notes integrates user's highest priority information sources, including e-mail, calendaring, group scheduling, to do list and more. Notes users can exchange messages via the Internet, work with any Web application, read and post topics to Internet newsgroups, search Web directories, and use X.509 certificates for security.

MAPI: Messaging Application Programming Interface (MAPI), a system built into Microsoft Windows that enables different e-mail applications to work together to distribute mail. As long as both applications are MAPI-enabled, they can share mail messages with each other.

MIME: Multipurpose Internet Mail Extensions (MIME). MIME extends the format of Internet mail to allow non-US-ASCII textual messages, non-textual messages, multipart message bodies, and non-US-ASCII information in message headers.

SMTP: Simple Mail Transfer Protocol (SMTP), documented in RFC 821, is Internet's standard host-to-host mail transport protocol and traditionally operates over TCP, port 25. In other words, a UNIX user can type telnet hostname 25 and connect with an SMTP server, if one is present. SMTP uses a style of asymmetric request-response protocol popular in the early 1980s, and still seen occasionally, most often in mail protocols. The protocol is designed to be equally useful to either a computer or a human, though not too forgiving of the human. From the server's viewpoint, a clear set of commands is provided and well-documented in the RFC. For the human, all the commands are clearly terminated by newlines and a HELP command lists all of them. From the sender's viewpoint, the command replies always take the form of text

lines, each starting with a three-digit code identifying the result of the operation, a continuation character to indicate another lines following, and then arbitrary text information designed to be informative to a human.

VPN: A virtual private network (VPN) allows two or more private networks to be connected over a publicly accessed network. In a sense, VPNs are similar to wide area networks (WAN) or a securely encrypted tunnel, but the key feature of VPNs is that they are able to use public networks like the Internet rather than rely on expensive, private leased lines. At they same time, VPNs have the same security and encryption features as a private network, while taking the advantage of the economies of scale and remote accessibility of large public networks.

9 References

[1] A System to Support the Analysis of Antivirus Products' Virus Detection Capabilities, Marko Helenius, 2002.

[2] http://www.faqs.org/faqs/computer-virus/

[3] http://www.wildlist.org/WildList/

[4] "A Short Course on Computer Viruses"; Fred Cohen; ASP Press Pittsburg 1990.

[5] Cohen, Fred; Establishing a Computer Security Incident Response Capability; 1992 all.net/books/ir/csl02-92.html

[6] Findings of Fact, United States District Court For The District of Columbia, 1999. usvms.gpo.gov/findfact.html

[7] USA Today, Jan 22, 1998.

[8] Symantec AntiVirus Research Center, 1999. www.symantec.com/avcenter/venc/data/mailissa.html, www.symantec.com/avcenter/venc/data/worm.explore.zip.html

[9] ICSA 2001 Computer Virus Prevalence Survey, 2001.

[10] http://www.microsoft.com/com/tech/com.asp

[11] ICSA 1998 Computer Virus Prevalence Survey, 1998.

[12] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/ldap/lightweight_directory_access_protocol_ldap_api.asp

[13] C. Nachenberg, 'Computer Parasitology', Proceedings of The Ninth International Virus Bulletin Conference, 1999. p. 7.

[14]Virus and Malicious Code Protection for Wireless Devices, Trend Micro, February 2001.

[15]Possible Virus Attacks Against Integrity Programs And How To Prevent Them, Vesselin Bontchev, Virus Test Center, University of Hamburg.

[16] NIPC CyberNotes Issue #2001-03; Feb. 12, 2001.

[17] ICSA 2001 Virus Prevalence Survey.

[18] NCSA ibid. pages 822 – 832.

[19] Fred Cohen, Computer Viruses - Theory and Experiments, Computer Security: A Global Challenge, Elsevier Science Publishers B. V. (North-Holland), 1984, pp. 143-158.

[20] Fred Cohen, Computer Viruses - Theory and Experiments, Computer Security: A Global Challenge, Elsevier Science Publishers B. V. (North-Holland), 1984, pp. 143-158.

[21] Yisrael Radai, Checksumming Techniques for Anti-Viral Purposes, Proc. 1st Int. Virus Bulletin Conf., September 1991, pp. 39-68.

[22] Fred Cohen, A Cost Analysis Of Virus Defenses, A Short Course On Computer Viruses, ASP Press, 1990, ISBN 1-878109-01-4, pp. 155-160.

[23] Fred Cohen, A Cost Analysis Of Virus Defenses, A Short Course On Computer Viruses, ASP Press, 1990, ISBN 1-878109-01-4, pp. 155-160.

[24] Anatomy of a Commercial-Grade Immune System, Steve R. White, Morton Swimmer, Edward J. Pring, William C. Arnold, David M. Chess, John F. Morar, IBM Thomas J. Watson Research Center.