Open Problems on Generic Computer Virus Recognition

VarunLaxman Bhandarkar#1
#1: Student, Master of Computer Applications
Sardar Patel Institute of Technology
Andheri (West) Mumbai, India
bhandarkarvaroon@gmail.com

Prof. Harshil T. Kanakia#2
#2: Associate Professor, Master of Computer Applications
Sardar Patel Institute of Technology
Andheri (West) Mumbai, India
email2harshil@gmail.com

Abstract**—** There has been a lot of technological improvement on dealing with known viruses and it has also helped in extending the issue of dealing with unknown viruses. But, however there are still problems related to dealing with viruses for which if we research and find solutions now will help us deal with unknown viruses in the future. In this research paper we are briefly going to discuss the open problems in computer virology and review some of the techniques used to solve these problems. Also, we will try to extend the already present solutions by researching on how we can improvise certain aspects of these solutions to generate more efficient virus recognition and detection techniques. The aim of this research paper is to encourage people to work on the computer virus problem.

Keywords**—***Computer, Virus, Virus Recognition, Anti-virus,Heuristic Analysis, Epidemology.*

*****

I. Introduction

In the expressions of Frederick B. Cohen, PC infection as a" system that can contaminate different projects by adjusting them to incorporate a conceivably developed duplicate of it" [1]. PC infection spreads starting with one PC then onto the next by duplicating itself to a current executable code. With the disease property, an infection can spread in a PC framework or system utilizing the approvals of everyone, along these lines influencing the client's project. Each system that gets influenced may likewise go about as an infection.

This conviction tells why there are simply less research groups in colleges and examination associations those arrangements with PC virology. An inadequate spread and information of the few known hypothetical results that have been acquired till now in this field mostly represents this. Upon profound examination, these outcomes exhibit that, , a nearer concentrate on in PC virology is totally required and crucial. Be that as it may, numerous open issues still exist similarly as PC virology is worried, in both hypothetical and specialized connection. Numerous different issues will develop later on, because of the creativity of malware spreaders. Interim, registers turn out to be increasingly mind boggling, increasingly touchy, making more seasoned infection assurance and protection models insufficient. The reason of this paper is to present what I think to be the most intriguing open issues in PC virology. I chose the issues whose determination, or profound study, is prone to add to the level of the field, furthermore to enhance the level of discovery and assurance applications.

Additionally, I attempted to concentrate on issues in PC virology, which can support researchers in software engineering or in arithmetic to do the exploration in this field. PC virology is not only a non-finishing chase between malware journalists and antivirus labs, yet gives a considerable measure of hypothetically profound issues to understand. The natural similarity between the infection and its partner in software engineering, as appeared by its name, infers that "the numerical strategies which have been produced for the investigation of the spread of irresistible ailments may be adjusted to the investigation of the spread of PC infections" [7].

The counter infection industry is a receptive industry as it just finds a cure of the infection when its assault. The work of battling PC infections is limited to the "catch, dissect, convey marks" cycle regular of the counter infection industry. Be that as it may, once the infection taints the PC system it will prompt a considerable measure of loss of monetary increases for commercial ventures which very hand-off on PCs. Hence, the expense to amend the misfortune because of an infection assault is too damn high. PC infection coders use numerous systems to pass recognition, for example, space filling, compacting and encryption in another hand; the antivirus attempt to distinguish the infections by the utilization of variation static and element techniques. Notwithstanding, all the current strategies are not satisfactory [2].

Along these lines, subsequently investigate paper will essentially concentrate on talking about the issues in PC virology which needs consideration of researchers to discover answers for up and coming development infections. In the second segment we will take a gander at the writing study of the rate at which infection assaults have occurred ever. How serious these assaults were and what was the budgetary misfortune caused by these infection assaults. In the accompanying segment we will talk about the five most common open issues in PC virology. Segment 4 will depict the Heuristic Analysis Approach of infection location and how we could enhance it with help of neural systems. Additionally in the segment 5 we will quickly talk about the Epidemiology of an infection. We will likewise examine the restrictions of these methodologies in short.

II. Literature Survey

In order, to understand the impact of a virus attack in terms of financial loss we will have a look at the various virus attacks which had taken place in the past. This survey will give us an idea of how dangerous a virus attack can be if it is triggered strategically. This will help us to understand the urgent need for finding solutions for efficient virus recognition.

I. Top costly viruses

	Top Ten Costly Viruses			
	Year	***Virus Name***	***Damage***	***Feature***
1	2004	MyDoom	$38 billion	25% emails Fast moving, ability to open random programs
2	2003	SoBig	$37.1 billion	Viral spam Copied and emailed itself
3	2000	ILOVEYOU	$15 billion	500000 Computers First one to attach itself to emails
4	2007	Conficker	$9.1 billion	Millions of computers logged keystrokes and downloaded code from hacker-selected websites
5	2001	Code Red	$2 billion	Had the ability to break into computer networks and exploit weaknesses in Microsoft software.
6	1999	Melissa	$1.2 billion	Micro Virus, infiltrated your Outlook
7	2001	SirCam	$1 billion	compromise confidential information, delete items or use up space
8	2003	SQL Slammer	$750 million	Affected banks and caused Internet speed to lag significantly across the globe
9	2001	Nimda	$635 million	The virus caused traffic and Internet speeds to slowdown.
10	2004	Sasser	$500 million	Devastated the British Coast Guard mapping system and caused numerous canceled flights.

Data in this table is referred from [4]
Some viruses employ different types of deception such as the following [8], [9]:

- Overwriting Virus: These viruses overwrite files with their own copy. This is a very old technique, but it is the easiest approach. Overwriting viruses cannot be disinfected from a system. Infected programs must be deleted from the disk.
- Companion Infection: One way to become a companion to an EXE program is to give the virus the same name as the targeted program, but use the .COM extension instead of .EXE. This method was deployed by the Globe virus, first detected in 1992. When the victim try to launch an EXE program, he usually types its name without the extension. In these cases, Windows gives priority to a file with the .COM extension over a file with the same name but with the .EXE extension.
- Appending Virus: In this method, a jump (JMP) instruction is inserted at the front of the host to point to the end of the original host. An example of this virus is Vienna. The appended technique can be implemented for any other type of executable file, such as EXE, NE, PE formats and so on. These files have a header section that saves the address of the entry point, which will be replaced with a new entry point to the start of the virus code appended to the end of the file.
- Prepending Virus: Such virus inserts its code at the front of programs. This is a easy kind of infection, and it is often successful. Computer Virus Strategies and Detection Methods writers have deployed it on various operating systems, causing virus outbreaks. Example of a COM prepended virus is the Hungarian virus Polimer.512.A, which prepends itself, 512 bytes long, at the front of the executable and shifts the original program stuff to follow itself.
- Cavity or space filler Virus: These viruses attempt to install itself in the empty space while not affecting the actual program. The advantage is that the virus then does not grow the length of the program and can avoid the need for stealth techniques. The Lehigh virus was an old example of a cavity virus. Because of the problem of writing this type of virus and the limited number of possible hosts, cavity viruses are less.
- Compressing Virus: A special virus infection technique uses the approach of compressing the content of the program. Sometimes this method is used to hide the

program's size growth after the infection by packing the program sufficiently with a binary packing algorithm.

- Encrypted Virus: It consists of a constant decryptor, followed by the encrypted virus body. It is relatively easy to detect because decryptor is constant. The first known virus that implemented encryption was Cascade on DOS. The simplest way to change the decryptors is to use a set of decryptors than a single one. The first known virus to use this technique was Whale. Whale carried a few dozen different decryptors, and the virus picked randomly.
- Boot Sectors Virus: This takes the advantage of the executable nature of master boot record (MBR) and partition boot sector (PBS). A system infected with a boot sector virus will execute the virus's code when the system boots up. Michelangelo virus is a type of a Boot Sectors Virus.
- Macro virus: This infects a Microsoft Word or similar application and causes a set of actions to be performed automatically when the program is started or something else triggers it. Macro viruses are surprising but relatively harmless. A typical effect is the undesired insertion of some comic text at certain points when writing a line.
- Malicious mobile code (MMC): Mobile code is a small program that is downloaded from a remote machine and executed locally with less or no user intervention. Java applets, JavaScript scripts, Visual Basic Scripts (VBScripts), and ActiveX controls are most famous examples of mobile code that we encounter while browsing the Web or reading e-mail. An attacker may use mobile code for a variety of nasty activities, including monitoring our browsing activities, obtaining unauthorized access to our file system, infecting our machine with a Trojan horse, hijacking our Web browser to visit sites that we did not intend to visit, and so on.

III. OPEN PROBLEMS ON GENERIC VIRUSES

In this section, we will discuss five problems which are as follows:

1. As more viruses are written for new platforms, new detection techniques must be developed. But we often have no way of knowing, whether these techniques will provide efficient results. That is, we don't know how well they will work or how many problems they will cause.

2. We have a reasonable, qualitative understanding of the epidemiology of computer viruses, characterizing their spread in terms of birth rate, death rate, and the patterns of program transfer between computers. Even if evidence suggests that viruses are uncommon, according to current theories this can only happen if virus birth rate is higher than their death rate. We discuss effects that might be responsible for this observation.

3. We are in the process of deploying digital strong system technology that searches new viruses, analyses them, and distributes remedies worldwide. But a more distributed technique, perhaps even a massively distributed approach, has pros as well. We outline the system issues that must be taken into consideration, and what simulation results would be useful, in understanding the trade-offs.

4. There have been few types of worms - freestanding virus-like programs that grow themselves and never be present in the file system. Yet most of our antivirus technology relies on detecting and removing viruses from a file system. We discuss the problems that worms engender, and suggest some of the new technology that may be needed to deal with them.

5. Current anti-virus technology is mainly reactive, based on finding a particular virus before being able to deal with it well. This approach is not effective for modern viruses that spread faster. We review the history of pro-active approaches, illustrating why traditional access controls are useless here, and describe newer techniques that show promise.

- The implementation of Heuristic
- Detection Analysis

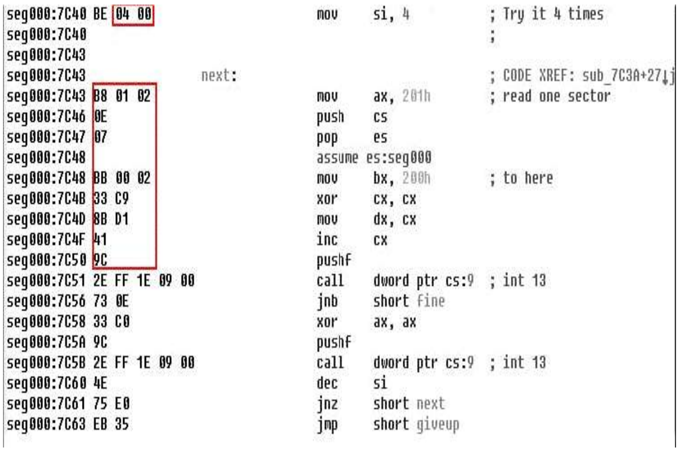
Heuristic investigation is a method conveyed by numerous PC antivirus programs developed to distinguish beforehand obscure PC infections, and additionally new sorts of infections as of now in the "wild"[3]. Most antivirus projects that use heuristic examination perform this capacity by executing the programming summons of a faulty program or script inside a specific virtual machine, accordingly giving the counter infection programming to inside do what might happen if the suspicious system were to be executed while keeping the incredulous system segregated from this present reality machine. It then investigations the orders as they are done, observing for normal viral exercises, for example, replication, document overwrites, and endeavors to shroud the presence of the incredulous record. In the event that one or more infection like activities are identified, the suspicious record is hailed as a potential infection, and the client educated.

Another normal procedure of heuristic investigation is for the counter infection project to decompile the suspicious code, and after that assess the source code. The source code of the suspicious project is contrasted with the source code of regular infections and infection like exercises. On the off chance that a specific rate of the source code matches with the code of basic infections or infection like exercises, the record is hailed, and the client alerted"[5].

Before we set to comprehend Heuristic examination, we should comprehend a fundamental case of infection discovery which incorporates the example coordinating and string checking strategy. An appropriate learning of this conventional system will help us comprehend the Heuristic Analysis better.

- *String scanning or pattern based detection*

The most famous method in anti-virus scanners is pattern based detection. It is not as efficient as some other methods but it can be performed faster. This method includes extracting a unique sequence of bits from a known virus and this sample is subsequently used like a fingerprint to match against while scanning for existence of the virus. Care has to be taken when selecting the bit sequence to minimize the number of false positives and at the same time match the virus and (ideally) possible types. Sometimes statistical methods are also used to extract these types. Figure 1 [6] shows the example of a search pattern for the "Stoned" boot sector virus. In this, the bit sequence selected was chosen by observing a behavioral peculiarity of the virus.



```
seg000:7C40 BE 04 00                    mov     si, 4           ; Try it 4 times
seg000:7C40                                                     ;
seg000:7C43
seg000:7C43                 next:                               ; CODE XREF: sub_7C3A+27↓j
seg000:7C43 B8 01 02                    mov     ax, 201h        ; read one sector
seg000:7C46 0E                          push    cs
seg000:7C47 07                          pop     es
seg000:7C48                             assume es:seg000
seg000:7C48 BB 00 02                    mov     bx, 200h        ; to here
seg000:7C4B 33 C9                       xor     cx, cx
seg000:7C4D 8B D1                       mov     dx, cx
seg000:7C4F 41                          inc     cx
seg000:7C50 9C                          pushf
seg000:7C51 2E FF 1E 09 00              call    dword ptr cs:9  ; int 13
seg000:7C56 73 0E                       jnb     short fine
seg000:7C58 33 C0                       xor     ax, ax
seg000:7C5A 9C                          pushf
seg000:7C5B 2E FF 1E 09 00              call    dword ptr cs:9  ; int 13
seg000:7C60 4E                          dec     si
seg000:7C61 75 E0                       jnz     short next
seg000:7C63 EB 35                       jmp     short giveup
```

- Stoned virus showing the search pattern 0400 B801 020E 07BB 0002 33C9 8BD1 419C [6]
- *Heuristic Scanning and Analysis*

Heuristic filtering is another strategy of infection identification that is neither mark based nor honesty based. A heuristic hostile to infection program examines an objective system (executable document, boot record, or potentially report record with a large scale) and investigates its project code to determine if the code shows up infection like. At the end of the day, a heuristic motor recognizes the summons inside a system that projects, for example, the replication instrument of an infection, the dissemination routine of a worm or the heaviness of a Trojan. On the off chance that the objective system's code shows up infection like, then scanner reports a conceivable disease. As the heuristic system does not utilize infection marks it can recognize new and obscure infections that have not yet been broke down by antivirus specialists. The purpose for this is the heuristic system does not utilize respectability data; it needn't bother with the fingerprints of projects to be taken and kept up when the PC is in a known clean state.

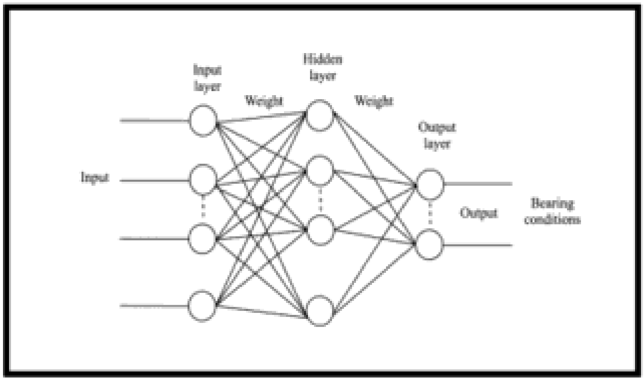
The counter infection industry likewise created systems for distinguishing beforehand unfamiliar infections. These procedures are normally called "heuristic" strategies since they are, by their tendency, imprecise.1 Heuristics are on the horns of the accurate issue as whatever other infection identification strategy: coming down with however many infections as could be allowed while having as couple of false positives as would be prudent. Scholars of heuristics have managed these issues in two distinctive ways. Some have led top to bottom beta tests of new heuristics, and attempted to incorporate their Open Problems in Computer Virus Research, by Steve R. White, Virus Bulletin Discussion, Oct 22, 1998, Munich Germany 3 heuristics to have tolerable false negative and false positive rates. Others have abandoned picking an individual decent exchange off and have given clients a chance to attempt to make this exchange off themselves by altering properties in the counter infection program. Be that as it may, the infection situation is evolving. Never again are we taking care of with basic DOS document or boot infections. Exceed expectations and Word full scale infections are at present the most incessant sorts of infections. Win NT infections are beginning to be composed. We have watched the principal test at a Java application infection. What's more, upcoming are all new sorts of infections that will take influence of the Internet to spread them. Future sorts of infections will emerge and get to be across the board much quicker than before. It is essential that we have approaches to discover new examples of these infections ahead they spread all inclusive. We might not have the recreation of protracted beta periods to tune our heuristics to dispense with false positives. What's more, we completely can't anticipate that clients will be sufficiently complex to tune many distinctive, entangled heuristics if the creators of the heuristics can't do as such.

One plausible beginning stage is a heuristic taking into account conventional marks, however marks that are regular to extensive classes of officially well-known infections. Mixes of these marks can recognize variations of infections in these classes. Probabilities that individual string marks will produce false encouraging points in non-tainted records can be evaluated with strategies that have as of now been created. Figuring false negative probabilities depends on portraying new infections in these classes as they show up after some time.

A second conceivable opening point is to utilize neural systems to endeavor to recognize tainted from uninfected documents. This strategy has officially demonstrated extremely effective for DOS boot infections [8].Neural systems have been concentrated on as general classifiers for a long time. Components are accessible for assessing the false positive probabilities of neural systems prepared on a given conveyance of positive and negative case.

Expounding on the usage of neural systems to endeavor separation between the infection classes the real difficulties to be face will be planning a proper info representation plan; managing the lack of accessible preparing information; finding a significant exchange off point between false positives and false negatives to - fit in with client desires; and making the product fit in with requirements on memory and rate of calculation expected to keep running on PCs.

Moreover, this neural system module could be independently utilized as a PC infection investigation and acknowledgment module by a framework that can help clients to get and examine infections in the event of assault [12]. Artificial neural networks (ANNs)- in the wake of preparing by presenting some traits of infection conduct can identify a wide assortment of infections and determination whether an arrangement of activities set up infection conduct. A artificial neural network (ANN) can be characterized [13] as a relationship of neurons, such that neuron results are associated through weights to all others, including some of the time themselves. A neuron is a direct robot which perceives a weighted aggregate of a few inputs as indicated by an arrangement of weights, and afterward makes sense of enactment or expels capacity to get a yield esteem, called actuation of the neuron. Weights measure the level of interrelationship between action levels of the neurons they interface [13]. Neural systems are frequently sorted out in layers of neurons.

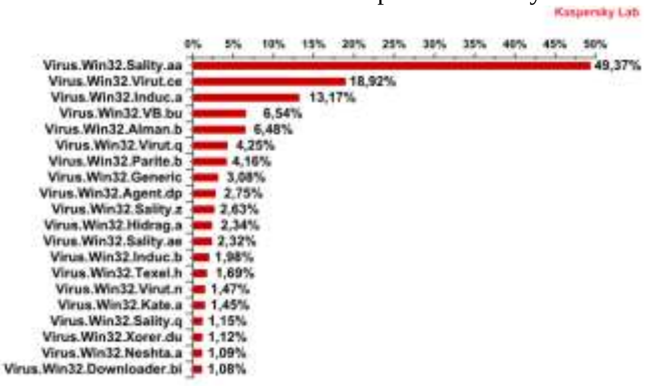


- Graphical Representation of a Neural Network[13].

A neural network can be skilled to recognize the behavior of a virus. This behavior can be characterized by a set of attributes that resemble with or are the same as the attributes that depict activity on a computer system in audit records. In the recall stage, when a set of attributes of system activity is given in the input of the neural network, the latter should be able to perceive whether the behavior portrayed in this set is the same or identical to the behavior of a definite virus that the neural network has been trained with. Thus, such a neural network can be used to analyze (probably in association with an trained system) the audit trails that a technique which monitors the operations on a computer system running a DOS operating surrounding generates, and to decide about the survival of computer viruses.

- Epidimology of virus

Using a biological analogy, virus epidemiology develops a theoretical model to predict the rate and extent of propagation of a computer virus infection. In order to develop the efficient and effective anti-virus policies and heuristics, we should not only learn how virus works, but also the factors that affect its prevalence, such as the new mechanisms and new transmission media. It is also needed to be aware of the different forms computer virus may take.



- Most frequently detected viruses(January 2009 – May 2010)[10]

In this part, we list definitions of several terminologies that are used to describe a viral epidemic [7]:

Birth rate – the rate at which a virus tries to replicate itself from one system to another (also known as infection rate). It is shown as .

Death rate – the rate at which a virus is removed from infected machines, usually when the user finds it and cleans it up (also known as cure rate). It is denoted as .

Topology – the patterns of contact along which diseases spread among individuals in a population.

Epidemic threshold – the link between the viral birth and death rates at which a disease will blow and become widespread. Above this threshold, the disease becomes a constant, recurring infection in the population. Below it, the disease dies out. It is denoted as .

The models of computer virus epidemiology give a reasonable qualitative understanding of the conditions under which viruses grow and why some viruses spread better than others [11]. In 1991, Jeffrey Kephardt and Steve White published “Directed-Graph Epidemiological Models of Computer Viruses,” [7] the first paper that adapt the mathematical epidemiology to the new problem of computer viruses. In this paper, they proposed an epidemiological model, SIS (susceptible -> infected -> susceptible) model, on directed graph.

Epidemiological models display that, by reducing the birth rate and expanding the virus death rate sufficiently, one can drive viruses below the epidemic threshold. “The birth rate of a computer virus is influenced by anything that hinders or promotes its duplication, including intrinsic mechanisms by which the virus infects programs, the rate of software transfer among computers machines, and precautions taken by users such as the use of a write-protect tab on a diskette or preventive anti-virus program. The virus’s death rate is influenced by the intrinsic characteristics that might disguise or acknowledge its presence, by user awareness and vigilance, and by its detection and subsequent removal” [14]. Even though we cannot have exact technology again viral attacks, we still can utilize the available traditional anti-virus technologies to accomplish the better effects. Virus scanners are an effective way to increase the death rate, particularly if they are structured such that they scan periodically without any prompting from the user, like a resident scanner. Scanners can also behave as filters to lower the viral birth rate [14].

- Limitations of models
- *Limitation of Heuristic Analysis*

Although stagnant heuristic detection programs can be relatively faster, they may detect only some of the numerous different methods of performing different viruses like operations. For example, various viruses may open files by using different codes and different techniques. In these cases the static heuristic detection system must look for a huge number of various ways each virus-like action may be implemented in order to accurately detect a virus-like behaviour. As a static heuristic detection requires a database covering huge number of desirable permutations of such action the situation may eventually become chaotic. This problem will be especially acute if a virus coder writes a "virus generator" code which provokes thousands of viruses at a time, permuting the order of its sections of code, but not changing its compelling behaviour. Such a multitude of viruses would be very hard to deal with static heuristic detection programs.

- *Limitation of Epidemology*

Since the epidemiological models are somewhat simple. There are still many of rooms for advancement. We address a number of limitations in this part [7, 11]:

• In SIS model, a corrupt node is retrieved from viral attack, and becomes vulnerable again. This is conflicting to the actuality. We know that individuals will become forever immune or at least to some degree once they recovered from the infection in real world. It should apply to computer science as well.

• In current models, all of the systems are assumed to have the equal birth and death rates. In the real scenario, this expectation will not hold true. How can we entertain these variants into the epidemiological model?

• When infection rate is higher than cure rate, the number of infections grows exponentially at initial stage, and eventually infuses at equilibrium. However, statistics displays that few viruses can stay at such an increasing infection rate. They will be abolishing of the map finally.

- Conclusion

It is difficult to measure the results of computer viruses. It is not only "hard cost", e.g. employee hours, but also "soft cost", as the dos (denial of service), lost staff yield etc. As pointed out in [8], "the virus risk continues to get poor despite corporate efforts. … Companies are experiencing more and more virus situations which result in higher and higher virus incident costs each year". Anti-virus software is reactive, not efficient and only works properly with the known viruses. But as one of the efforts to solve the problem of virus, adapting and applying mathematical epidemiology in virus allows the qualitative analysis of the conditions under which the viruses grows, and the relation between the infection rates, the number of infected computer machines along the time scale.

We have scrutinized a few open problems in virus research. My hope is to convince those who would like to do and in-depth research in this area that there are significant problems yet to be examined, and that these require dedicated work by smart people. The field is by no means complete, and easily predictable problems in the relatively near future will need substantial new invention to avoid considerable problems with new viruses. Beyond the near future lies an extensive and ever-changing landscape, in which today's programs will evolve into complex ecologies of interrelated threads that span the planet, collaborating with each other in ways we cannot anticipate. And undoubtedly, lurking within these ecologies will be the conceptual descendants of today's computer viruses, spreading wherever they can. The fight between microbes on the one hand, and plants and animals on the other, has grown in the biological world for millions of years. There is every reason to believe that the fight in cyberspace will also be long, and will continue to require insight and invention for a long time to come.

References

[1] Frederick B. Cohen. A Case for Benevolent Viruses, Pittsburgh, ASP Press, Pittsburgh, PA, 1991.

[2] Essam Al Daoud, Iqbal H. Jebril and Belal Zaqaibeh, Computer Virus Strategies and Detection Methods, Int. J. Open Problems Compt. Math., Vol. 1, No. 2, 2008

[3] Wong, W.; Stamp, M. (2006). "Hunting for metamorphic engines". *Journal in Computer Virology* **2** (3): 211–229

[4] http://www.investopedia.com/financial-edge/0512/10-of-the-most-costly-computer-viruses-of-all-time.aspx

[5] https://en.wikipedia.org/wiki/Heuristic_analysis.

[6] Peter Szor, "The art of computer virus research and defence", February 2005, Symantec press.

[7] J. O. Kephart and S. R. White. "Directed-Graph Epidemiological models of Computer Viruses". In Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy. Oakland, California. 343-359. May 20-22, 1991.

[8] J. cock, Computer Viruses and Malware, Springer (2006)

[9] E. Skoudis and L. Zeltser, Malware: Fighting Malicious Code, Prentice Hall (2003)

[10] https://securelist.com/analysis/publications/36305/review-of-the-virus-win32-virut-ce-malware-sample/

[11] S. R. White. "Open Problems in Computer Virus Research". Presented at Virus Bulletin Conference. Munich, Germany. October, 1998.

[12] D. Guinier, Computer 'virus' identification by neural networks, ACM SIGSAC Rev., 9(4) (1991) 49-59.

[13] B. Kosko, Neural Networks and Fuzzy Systems, PrenticeHall, 1992.

[14] J. O. Kephart, S. R. White and D. M. Chess. "Computers and Epidemiology". IEEE Spectrum, 20-26. May 1993.

[15] J. Lyman. "In Search of the World's Costliest Computer Virus". NewsFactor Network. 2002. http://www.newsfactor.com/perl/printer/16407/.