

U.S. Patent Number 5,485,575: Automatic Analysis of a Computer Virus Structure and Means of Attachment to its Hosts

David M. Chess, Jeffrey O. Kephart and Gregory B. Sorkin

summarized by Sean Davey

October 20, 1999

1 Introduction

The invention for this patent is a method of automatically verifying the identity of a virus that has attached itself to a host file and reversing, when possible, the transformation enacted by the virus thus returning the host file to its original functionality. The methods contained in the patent are very general in nature and can be applied to other data transformation scenarios but the focus is on viruses. The majority of known viruses contain several properties that make automatic analysis and restoration computationally feasible.

2 Motivation

The proliferation of viruses is rampant today. Many of the viruses cause damage to systems and data either through malicious intent or accidentally. Methods for discovering the presence of viruses and removing them is crucial to the long-term viability of computing and for contributing to a sense of security in a highly networked environment. The vast number of the viruses inspires the desire for tools that can deal with them in an automatic way. The fact that the virus-writing community continues to develop tools for automating the process of creating new viruses certainly increases the value of such defenses.

3 The Invention

The characteristics of a virus that need to be known in order to detect its presence and to remove it include its method of attachment, the form and location of its “invariant” regions and the location and encryption method of

data from the host file that is present in the infected file. Viruses often transform portions of the original file in such a way that the functionality of the original is preserved. This is because the success of a virus is dependent on it remaining undiscovered. At a high level the method of the invention is:

- obtain a set of “sample pairs,” each of which consist of a transformed data sample and a corresponding original, untransformed data sample.
- locate regions in the transformed data sample that contain unaltered data and regions that contain either new data added by the virus or data that has been transformed by the virus in some way.
- match new or transformed regions across different samples to obtain a description of the regions or portions of regions that are “invariant” across samples.
- locate within the new or transformed regions any data from the original data sample that was placed there by the virus, possibly after encryption.
- generate a prescription for determining if any given data sample has been infected by the virus.
- generate a prescription for repairing a data sample that has been transformed in a function-preserving manner into a form that functionally equivalent to the original.

4 Details

4.1 Obtaining Host/Infected-Host Sample Pairs

Determining the exact behavior of a virus is obviously uncomputable. However typical viruses exhibit only one of a small number of behaviors. The invention relies on comparisons between infected and non-infected copies of the data. Uninfected data can be obtained from original sources, backups or other copies in a network assuming that they can be verified to be uninfected by calculating a checksum or some other method.

4.2 Virus Attachment Analysis

There are a few simple methods of attachment that are used by most viruses. A virus can append itself to the end of a file and modify the beginning of the host so that execution starts with the viral code instead of the host's code. A variation on this technique can make it difficult to statically determine the location of the start of the virus at the end of the file (e.g. the beginning of the host is changed to include more than a single branch to the viral code). Another method involves inserting the viral code at the beginning of the file. A third method is copy a section of the host code to the end of the file and then overwrite the beginning of the host with viral code. Even though most viruses

use one of these methods, the invention uses a more general description for infected files which partitions them into regions of data that match the original file exactly interspersed with regions that contain new data or data from the original that has been transformed by the virus. In most cases the number of these regions is small. The description syntax used by the invention is quite flexible and handles such details as viruses that modify different host types (e.g. EXE and COM) differently, region boundaries that may vary by small amount to account for code alignment performed by the virus, etc. The patent presents the description syntax in some detail but it does not describe at all the algorithms used for determining the boundaries of the regions.

4.3 Locating “Invariant” Regions in Transformed Data

Regions that differ from the original sample are further divided into sections that are constant across samples and those that vary across samples. It is important that enough samples are gathered so that sections that appear to remain constant but actual vary across a large enough sample set are uncovered. The method for determining the sections is compared to problems in computational biology. In particular the algorithms used for DNA sequence assembly are referred to but details are not given. A general outline of a technique of finding the largest common substring and then finding recursively smaller substrings is mentioned.

4.4 Dealing with Encrypted Regions

Many viruses encrypt portions of themselves and/or portions of the host. Historically viruses use very simple encryption methods. The patent describes several of the currently used methods including adding a constant to every byte, adding a constant to every word, performing an XOR of every byte with a 1-byte constant, performing an XOR of every word with a 1-word constant, performing $A + iB \bmod 256$ on each bite given two 1-byte keys A and B, rotating each byte by a constant number of bits, and some variations on these. All of these methods are tried in reverse to find regions that are invariant once encryption is taken into account. The regions that were found to “invariant” before decryption was tried are searched and compared to find the locations of the encryption keys. This involves narrowing the choices from many candidates across many samples. It is often the case that the keys are actually modified by some constant value before they are used by the virus to decrypt so this must be taken into account. However the matching regions can be found without finding the key modifier and once the newly found invariant regions are found the key modifier can be by found by comparing the regions with the uninfected regions from the original. The comparisons are done using known algorithms for Minimum Cover, another NP-complete problem. The patent covers quite a bit of the details regarding the methods to find encrypted “invariant” sections and the “key” information needed to decrypt them. In some cases the invention

is unable to determine the required information and so a repair method cannot be automatically constructed.

5 Conclusion

The patent describes a method of automatically determining a description of a virus that can be used to detect data infected with that virus and, in many cases, restore the infected file to its original functionality. Several of the techniques involved are either manifestations of NP-complete problems or extremely time consuming nonetheless. The authors claim the limiting factors on the size of the problems due to the nature of viruses allow the invention to operate in a reasonable amount of time. The limiting factors include the small numbers of attaching methods and the small number of original, “invariant” and variable regions, amongst others. In my opinion, some of the limiting factors that the invention relies on are a product of the history of virus evolution and not inherent in the nature of a virus. For example, current viruses use fairly simple encryption methods because those methods served the purpose of the virus writers at the time but there is no reason that more complex methods cannot be utilized in the future. The automation of virus analysis may merely be another step in the escalating race of sophistication between virus writers and those who try to combat them.