Submission of Paper for National Information Systems Security Conference

THE MACRO VIRUS AND VIRUS SCANNING SOFTWARE AN ANALYSIS OF THEIR INTERACTION

ABSTRACT

In order to protect a computer system from being infected by a computer virus, many individuals install anti-virus software assuming a shield of immunity has been put between them and these potentially devastating weapons. This paper will attempt to clarify this situation by illustrating potential limitations and capabilities of current virus scanning software as well as some of the limitations and capabilities of a particular breed of computer viruses, the Macro Viruses.

KEYWORDS: Computer Viruses, Computer Security, Anti-Virus

LT William A. Macchione
Naval Air Systems Command Air 4.5.1.1
PEO(T) PMA-241
Attn: LT Macchione / Bldg IPT / Suite 452
47123 Buse Road Unit # IPT
Patuxent River, MD 20670-1547
(301) 757-7013
FAX (301) 757-7354
E-Mail: macchionewa.jfk@navair.navy.mil

Daniel Warren
Computer Science Department
Naval Postgraduate School
Monterey, CA 93943-5118
(408) 656-2353
Fax (408) 656-2814
EMAIL: warren@cs.nps.navy.mil

INTRODUCTION

An important consideration regarding the results of this paper is the fact that this exercise in investigating viruses and anti-virus software is by no means exhaustive or complete. It is simply a pursuit to gain an insight into the workings of these programs. All work performed here was conducted by a single individual on one computer system. As testing is extremely limited in such an environment, the most common breed of virus in existence today was chosen as the subject; specifically, the Microsoft Word Macro Virus.

The objective here was designed to approach the different types of virus scanning programs objectively and see what could be found out in terms of detection capability, signature identification, and triggering mechanisms. According to Dr. Fred Cohen, one of the early pioneers in the field of computer viruses and the man who coined the term, the best way to learn about a computer virus is to write your own. Importantly, he pointed out to only make use of the pseudocode he presented in his book so as not to be influenced by other programmer methods or techniques.[1] The true strength of his argument will be apparent when virus scanner performance is presented. Armed with that information and the help file for Microsoft Word 95's programming language[2], WordBasic, the virus was written.

It is very important to realize that all results here were obtained on a particular computer system. While the configuration of the test platform is fairly common, the possibility of differing results does exist across different machines. Finally, due to the possible dangers associated with computer viruses and the implications associated with the Department of Defense (DoD) providing the source code for them, this information will not be included in this presentation. If the source code is desired for an approved reason it may be obtained with proper authorization of the DoD.

Virus Development

The most complicated issue here is where to begin. Having never embarked on writing something with little more than a notion of what to do, the author set out to identify what constitutes a computer virus and what tools were available in the Microsoft Word Macro Language. According to Dr. Alan Solomon, "A virus is a program that copies itself."[3] This definition is in agreement with the formal definition provided by Dr. Cohen.[4] While these definitions might describe a virus in the strictest sense, Dr. Cohen also more loosely describes a virus through pseudocode in his book as basically a four part program; 3 subroutines and a main

---

[1] F.B. Cohen, A Short Course on Computer Viruses, Second Edition, John Wiley & Sons, Inc., New York, 1994, page 4-5.

[2] Microsoft Word Ver 7.0 for Windows 95 was chosen to write the virus for it is the most advanced version available which is document compatible with Version 6.0 for Windows and 6.0 for Macintosh. Word 97 uses Visual Basic, which is more advanced, however commands are not necessarily compatible with the earlier versions. Additionally, due to document compatibility, Version 6.0/7.0 offer platform independence capability. Some functionality is different, however the Macro Language can determine the platform it is running on and execute the correct commands accordingly. None of the programs enclosed make specific use of this feature. They all operated on Windows 95 and Windows NT machines however Macintosh capability has not been analyzed.

[3] A. Solomon, *Dr Solomon's Virus Encyclopedia*, Dr. Solomon's Software Ltd, Burlington, MA, 1996, Page 1.

[4] Cohen, page 1.

program.[5] The subroutines consist of an infection routine, which is the one that formally declares the program as a virus, a damage routine, and a trigger-pull routine. The main program simply calls these routines. It infects the program, checks for a trigger-pull condition, and if it exists (the trigger is pulled), does damage, else it exits. Using this methodology and the tools available in WordBasic, the program was written.

The Virus program written was designed to target a system containing a specific trigger and hence it will be called TARGETING MACRO VIRUS. This virus contains a main program, which includes an infection section, and a trigger-pull routine, which calls a damage routine if the trigger condition is satisfied.

The design of the program is based on an Information Warfare aspect of Computer Viruses and their propagation. This program actually targets a computer system. If the computer system has a particular file located on it, then the damage mechanism is triggered. Here, the damage chosen is minimal as it simply renames the document, however there is no limitation that would restrict the implementation of a malicious payload.

Virus Scanners

For the analysis, two variations (Windows 95 and Windows NT) of four virus scanning software packages were used. The programs were Dr. Solomon's, FPROT Professional's, Norton's, and McAfee's Antiviral toolkits. These programs were chosen for two reasons. First, they are ones that were available to obtain due to DoD contracts and purchasing; and second, they are usually rated among the very best in the field. In fact, in the January 1998 issue of SC Magazine (Info Security News), Dr Solomon's was their number one pick for anti-virus programs with McAfee's at number two.[6] With this understood, if a limitation is identified in one of these programs, it is likely to be evident in others.

All of the programs include two parts to them, one part is a scanner and the other is what they call dynamic virus protection (DVP). DVP involves identifying virus action as it happens where scanner software checks memory and storage devices for viruses.

What exactly is a virus scanner? Dr. Solomon puts it "A scanner is a program that knows how to find a particular repertoire of viruses."[7] This poses a particular problem in the virus game. A scanner can only detect viruses that exist in their library. New viruses will not be detected. Every time a new virus is written, the library must be updated. Several of these programs included easy to use "live updates" via the Internet. This said, what can be done about unknown viruses (the real problem)? Many of the programs now incorporate a method of scanning called advanced heuristics whereby they can check for probable new viruses. Heuristics consist of an educational method in which learning takes place through discoveries that result from investigations.[8] Basically, they are claiming that the programs can check for viral activity and not just viruses that exist in a library. In many ways these are the same methods that are used by gaming programs to decide the next move.[9] Since the results are only plausible, there is chance of

---

[5] Cohen, pages 4-5.

[6] SC Magazine, *1998 Anti-Virus Review*, West Coast Publishing Inc, Framingham, MA, Page 44.

[7] Solomon, Page 17.

[8] *The American Heritage® Dictionary of the English Language, Third Edition* copyright © 1992 by Houghton Mifflin Company. Electronic version licensed from INSO Corporation; further reproduction and distribution restricted in accordance with the Copyright Law of the United States. All rights reserved

[9] R. W. Hamming, *The Art of Doing Science and Engineering: Learning to Learn*, Gordon and Breach Science Publishers, The Netherlands, 1997, Page 75.

false alarm (the recognition of a virus when, in fact, there is no virus present, Type I Error). The goal of the analysis here is to examine all eight of the programs to see if they can discover the virus and if so, what triggers recognition.

The Design and Redesigning of the Test Procedure

The first program tested was Dr. Solomon's Find Virus for Windows 95 and Anti-Virus Toolkit for Windows NT. Fully expecting both systems to identify the virus, the plan was to use the modular capability of the program to remove certain procedures until the system no longer detected it as a virus. When this point was reached, the process would be repeated with the complete program and remove lines of the critical section until the program was no longer recognized as a virus. Through the iterative process, the hope was to discover what elementary items were causing virus recognition. Some element or elements must be responsible. To ensure maximum likelihood of detection, it was verified that the necessary options for heuristic detection were selected, and wherever possible, the most sensitive setting. It was preferred to attain a false alarm rather than miss detection of the virus. This would give the anti-virus software the maximum advantage and detection capability. After inserting the first disk in the drive to begin the test analysis, the floppy scan initiated, and the results were staggering!

Neither the Windows 95 nor the NT version could detect the virus. Next the attempt was made to try and infect Microsoft Word. DVP was enabled and again no detection! Clearly this virus was doing things it should not. It copied itself appropriately, scanned the disk for the target file, and when found, renamed it. It is hard to believe that the "advanced heuristic" nature of these programs did not detect the virus. In order to be sure that the anti-virus toolkits were operating correctly, it was decided to incorporate a variant of the WAZZU virus into the testing process.

A brief analysis of the WAZZU virus indicates a different method of infection and a different trigger and damage mechanism, however the end result is the same, namely a program that qualifies both under the loose and formal definition of a computer virus. WAZZU is notorious in the world because if a user is editing a document or simply opening it to print, they might never realize that the document has been changed. That is of course, until they send it out and it gets returned because of gross wording errors and possibly a comment, "What's this wazzu thing!" In fact, do not be surprised if such an error exists in this document. The only precaution this document has against such activity is that it was written in Microsoft Word 97, which uses Visual Basic, and therefore the WAZZU variant used here is not compatible with it. Now, since this virus has been around for some time, all the scanners should be able to identify it. The procedure now was to create two test files and two test disks. Each disk had one file on it. The WAZZU virus was on one and the TARGETING MACRO VIRUS was on the other. Testing could once again be continued.

Testing and Triggering Analysis

Of course, the scanners (Dr. Solomon's) again failed to identify the TARGETING MACRO VIRUS, but they did correctly identify the WAZZU virus. Now what triggered detection? Through the iterative process described earlier, five lines were discovered to cause this. They are:

Obtain File Information
Obtain Macro Name
Obtain File Name containing Macro
Copy Command from Word to File
Copy Command from File to Word

The particular sequence of which lines were removed is important. If any two of the first three or the last two were removed, the program would not be identified as a virus. Additionally, the NT Version required only one of the first three or sometimes the fifth line only, however these results were not constantly reproducible. It is suspected that the heuristics algorithms are responsible for this "fuzzy" behavior. Clearly removing any of these lines restricts the program's performance and thus it fails to function correctly. Heuristically, it is not doing what one would expect a virus to do so it should not be detected! Now since this virus is in the library as well, there must be some combinations of the above commands that are listed as a signature of the virus file. No other lines made a difference if they were removed or not.

Next, Dr. Solomon's was removed from both systems and FPROT was installed. Interestingly, the results were extremely similar. Again the TARGETING MACRO VIRUS checked clean. In the WAZZU virus, once again, those five lines were the critical items. This time only one of the first three lines was necessary and in the 95 version of FPROT, only the fifth line needed to be removed to cause failure of detection. The results here were more stable indicating the probability of a lower capability to detect the unknown and to detect by behavior. With a sense of confidence that the system was understood, the next program, Norton Anti-Virus (NAV), was installed and tested.

NAV proved to be a little different but centered on the same elements. Both 95 and NT versions failed to detect the test virus, however, they exhibited slightly different behavior between themselves on the WAZZU variant. They still focused detection on the main portion of the program. However, this time the 95 version required removal of a group of six lines together including the ones indicated above. Again using advanced heuristics, sometimes the first Copy line was not necessary. In NT, if either the first, third, or fourth and fifth lines in the grouping above was removed, it would fail to detect but sometimes the fourth line was not necessary. Basically it appeared that all virus scanning programs were created, more or less, equal. That hypothesis appeared true, until testing McAfee.

The only similarities in these tests are that it too failed to detect the test virus and the Copy command was once again a critical line. That is where the similarity stops. In the previously listed group of five lines, the fifth was always more critical then the fourth. Copying the macro from the file to the program was viewed as bad activity. McAfee's product views the fourth line as the only critical one in that set. However removing this line alone does not stop detection, the line used for error trapping must be eliminated as well. Quite a different signature to look for! In fact the strangest behavior was noted here in that there was different performance between the scanner software and the DVP. In previous testing, all that was necessary to test was remarking out the lines that were suspect. In McAfee's product, the scanner software read the whole data file, ignoring the fact that a line might be remarked, and identified the remarked WAZZU as the real thing. The DVP portion of the program could do this not because the Macro would not run and hence no virus exists. Additionally, the DVP keyed onto the error handling line. This is evident from the fact that the following code was identified as a virus:

```
Sub MAIN
On Error Goto errCaught
Goto bye
errCaught:

bye:
        On Error Goto 0

End Sub
```

Clearly a false alarm, but some combination of these statements made the DVP believe it was the WAZZU virus.

Summary

What does all of this information mean? There is a strong belief by many individuals in the relative uselessness of virus scanning software to the individual user. This is not to say that this will always be the case, but as Dr. Richard Hamming[10] points out in his book, Artificial Intelligence is far from the realities that were once projected for it and heuristic detection is a form of AI. With its current implementation, this method of detection does not seem to provide the protection required against the unknown. Is virus scanning against the known viruses even necessary? This is a question dependent on individual needs and the use of the computer system. For a corporate system where the data is crucial, it helps. However, it is only one layer of defense. As it has been demonstrated time and again, only multi-layered defensive systems can provide any level of assurance against such activity. Additionally important, and a fact that must not be overlooked, is the decrease in performance of a computer using such software. In an article written last year in PC Magazine[11], it stated that the scanners that performed the best caused the greatest decrease in system performance, up to eight percent degradation. In some applications, that is acceptable, however many users are unwilling to sacrifice that level of performance. After all, it is an awfully high insurance premium to pay.

Clearly from the results of this testing, currently (Dec 1997) these programs provide a very limited level of protection against new Macro viruses. This position is not taken without justification. The author is an inexperienced programmer and it was a first attempt to write a virus, and it was not detectable by any of the current generation of virus scanning programs. Since it appears that the method of infection is the only signature/activity that is checked for, it was decided to create a hybrid virus to test this theory. This virus utilizes the method of infection from the TARGETING MACRO VIRUS and the trigger and damage mechanism of the WAZZU virus. Effectively created is a new WAZZU variant. Now running this against the scanner programs there was no surprise; again it was not detected. This fact illustrates a failure of these programs to effectively employ a method to detect viral activity.

An acquaintance with whom the author had some discussion of this topic stated that she believed there was nothing currently that could be done about these macro viruses. She said they had repeatedly infected her company and she did not seem to have any faith in the virus scanning

---

[10] Hamming, R. W., *The Art of Doing Science and Engineering: Learning to Learn*, Gordon and Breach Science Publishers, 1997.

[11] PC Magazine, *What the Numbers Mean*, Ziff-Davis Publishing, April 8, 1997.

software. Today only one virus prevention program catches every instance of a macro virus, however its false alarm rate is exceedingly high. This is because is not even a real virus scanner. The program that is being referring to it the Macro Virus protection offered in Word Version 7.0a or later or by download from Microsoft's Web site for installation on version 6.0 or 7.0. This program is non-discriminating and simply identifies the existence of word macros. Therefore, it must be coupled with training to be used effectively. In an organization, if a new document is ever discovered with a macro, it should not be allowed to run. It should be taken to a stand alone workstation where it can be analyzed by a competent programmer for viral activity. With educated users and controlled inputs to the system, this problem can be made manageable. However, this is far from the automated solution that scanners offer.

Acknowledgements

One final note; this discovery and educational experience would not have occurred had the I failed to follow the advice of Dr. Cohen. This program was written and programmed individually using my own style, therefore it was truly unique. The uniqueness is why the scanners failed. Sometimes beating your head against the wall can lead to startling discoveries. To that end I owe Dr. Cohen a debt of gratitude for this task. Without so much as blinking he told me to write a virus. I felt overwhelmed and worked for a long time until I got it to work without error. Testing proved the greatest reward, not one of these programs looked for my method of thinking. I had, indeed, performed something truly unique.