Return to Table of Contents Return to last document
Query Query results Find Find again

Independent AV Testing

Ferenc Leitold, Ph. D.
Veszprém University, Hungary

About the Author

Ferenc Leitold graduated from Technical University of Budapest in 1991. He received his Ph.D. at Technical University of Budapest too, in 1997 in the theme of computer viruses. Currently he teaches in the Department of Information Systems at Veszprem University. He teaches computer programming, computer security, and computer networks. His research interest is based on computer viruses: mathematical model of computer viruses, automatic methods for analysing computer viruses, and testing anti-virus software.

Mailing Address: Ph.D. Ferenc Leitold, Kupa str. 14. H-8200 Veszprem, HUNGARY; Phone: +36 30 9599-486; Fax: +36 88 413-241; E-mail: fleitold@veszprog.hu; URL: www.checkvir.com

Descriptors
computer virus, anti-virus, anti-virus testing, disinfection testing, quality assurance, quality engineering

Reference to this paper should be made as follows:
Leitold, F. (2002). Independent AV Testing. In U.E. Gattiker (Ed.), EICAR Conference Best Paper Proceedings (ISBN: 78-987271-3-3) pp. 120-128. Copenhagen: EICAR.

Independent AV testing

Abstract

Software testers and quality engineers have to test their programs in as many various environments as possible with a lot of input combinations. In the case of anti-virus products, this task is more difficult because the product changes continuously, newer and newer procedures are being built into them. Anti-virus software usually includes several thousands of detection and disinfecting algorithms, which should be tested on a great number of virus samples and of course on non-virus files as well. In this paper, a new independent anti-virus testing procedure is presented. Some of the new ideas in this test are:

- *Testing should be executed on a large number of virus samples of the same virus body.*
- *Testing should be extended to disinfection capability.*

Introduction

Software testers and quality engineers have to test their programs in as many various environments as possible with a lot of input combinations. In the case of anti-virus products, this task is more difficult because the product changes continuously, newer and newer procedures are being built in them. Anti-virus software usually include several thousands of detection and disinfecting algorithms, which should be tested on a great number of virus samples and of course on non-virus files as well.

Usually the following problems occur using anti-virus software:

- The anti-virus software can detect a virus but does not deal with special cases (e.g., too small or too big infected files where the virus usually makes mistake).
- The behaviour of at least two versions of an anti-virus software developed by the same company and working with the same engine are different(e.g., Win95 version of an anti-virus product can detect but the Win2K version of the same product is unable to detect the same virus in the same sample).
- The anti-virus software is able to detect a particular virus but only in some samples (e.g., usually in the case of polymorphic and macro viruses).
- The anti-virus does not correctly wipe all virus-related macros from a document and after infecting this document with another one virus, a totally new macro virus may appear.
- The anti-virus program is unable to distinguish between similar but different viruses. In some of these cases, the program makes mistakes during the disinfecting procedure.
- The anti-virus program is able to correctly disinfect a particular virus but from some samples, however sometimes the result is bad and the disinfected file it cannot be executed.
- Other functional problems (e.g., the anti-virus software hangs up during the disinfection procedure for a particular virus).

In 2001, Veszprog Ltd. started to develop new automatic and semi-automatic methods solving this problem in the course of a new project. This short documentation highlights the first results of testing in the real environments executed during this project.

Note: The purpose of this test is not the ranking of anti-virus software. The results of this test do not claim that an anti-virus product is better than another. Of course it cannot claim that any anti-virus product works totally correctly and that it has not made any mistake. The tests claim only that there are a number of cases where the tested anti-virus software fails. The main goals of this test are that that the results of this and further tests help end users in the choice of anti-virus software, help anti-virus developers in their work, and the anti-virus products become better and free from bugs.

Testing algorithms

Testing of anti-virus software is a very delicate matter. Bearing this in mind at the beginning of the project a number of rules were accepted. Some of them are as follows:

- Infected files have to be made by breeding the virus. In this case, the "virus property" of infected files is proved.
- No other application will be installed on the test platform except the anti-virus software.
- Every test has to be repeatable.

According to the mentioned rules, the testing process included three steps:

1. Replicating viruses,
2. Running AV software to detect or disinfect viruses, and
3. Analysing the results.

The testing process was designed for old (e.g. DOS or Boot) viruses too. A big percentage of DOS/Boot viruses can spread under the Windows operating systems. According to Vesselin Bontchev (Bontchev, 2001), viruses that started spreading some years ago may still be active nowadays and infection reporting services or the technical support departments of the anti-virus companies do not provide a realistic picture of how widespread a virus really is.

Virus replication

During the project some automatic and semi-automatic replication (breeding) algorithms were designed and developed. These procedures relate to three different virus types. There are various breeding systems for the replication of each virus type.

DOS program viruses (COM/EXE)

DOS program viruses are usually bred using DOSEMU or VMWare under the Linux operating system. In the case of some viruses, the native operating system is required. It was provided by the extended version of the Automatic Virus Analyser System (Leitold, 1995).

Boot viruses

Breeding of boot viruses can be automated only if the virus does not require the native environment. (Supposing that there is not any robot for floppy disk changes.) It can be easily solvable using DOSEMU or VMWare under the Linux. The rest of the boot viruses are bred manually but, of course, the number of required activities is minimised.

Macro viruses / 32 bit program viruses

Macro viruses and 32 bit program viruses are bred using VMWare under Linux with the aid of Automate, WinTask or MacroExpress, which can automate activities under Windows. For breeding macro viruses, all possible applications (MSOffice95, MSOffice97, MSOffice2000) should be used because the results (the infected files) can be different in the case of each of these applications.

Viruses infecting other computers

These viruses can infect no other code (executable, macro, or other) on the same computer; they can infect another application on another machine using a TCP/IP connection. For breeding these types of viruses, VMWare is a great tool. Using VMWare, two or more virtual computers can be installed on a single computer in a virtual TCP/IP network. With the aid of Automate, WinTask or MacroExpress on each virtual machine, these types of viruses can breed.

Running AV software on viruses

AV software eas executed only in its native environment except in the case of boot viruses. For the testing of boot viruses, VMWare under Linux and Automate were used. In each case, an attempt was made to create a report file but in some cases it was not possible (i.e., the AV crashed or the AV was unable to handle the big report file).

Analysing results

At this writing, there are only procedures for analysing the results for DOS program viruses and for macro viruses. The development of algorithms for analysing the test results on other virus types is under development.

DOS program viruses (COM/EXE)

In the case of DOS program viruses, the analysing procedure includes two main parts:

- Using the report file of the anti-virus software inform us how many different virus were found in how many files. It can help only if the software generates the report file correctly.
- On the other hand, the files after disinfecting should be checked. In this case the files after disinfecting and before disinfecting are checked. If they are the same then it means that the AV did not do anything with this file. It can mean that the

AV does not know this particular virus or the AV is unable to disinfect it. This distinction can be determined using the report file (if available). If the file after the disinfecting and the original goat file are the same or there are only some minor changes then it means that the AV disinfected the file correctly. In other cases, the AV failed.

Comparing to the infected file is an exact problem but comparing to the original file is not so trivial. During the test an "intelligent compare" was used. This "intelligent compare" can check the following cases and display them by flags:

q The cleaned file is bigger than the original goat file.
S The cleaned file is smaller than the original goat file. (!)
z Some 0 values has been changed.
G The cleaned file is bigger than the original goat file and the difference is more than 16 bytes.
Z A non-zero byte in the program area has been changed. (!)
h At least one byte of the header which is never used has been changed.
H At least one byte of the header that is often used has been changed.
e The EXE file starts with 'ZM' instead of 'MZ'.
E The EXE file does not start with 'MZ' or 'ZM'. (!)
p The value in the EXE header indicating the EXE file size has been changed.
m The values in the EXE header indicating required memory have been changed.
Y The CS:IP and/or CS:SP values in the EXE header have been changed. (!)
R Some values in the relocation table have been changed. (!)
D The size of the header has been changed.
c The checksum field has been changed.
o The overlay flag has been changed. (!)

Note: (!) means that they are critical flags.

Macro viruses

In the case of macro viruses the analysing procedure includes two main parts also:

- Using the report file of the anti-virus software inform us how many different virus were found in how many files. It can help only if the software generates the report file correctly.
- And on the other hand, the cleaned files can be checked easily: only the name and the content of the macros in it should be displayed.

The current status of the project

The project is currently running and the following table shows the actual status (January 2002) of the project depending on the type of tested viruses.

Table 1: The status of the project (January 2002)

	DOS executable	WIN32 executable	Macro (DOC, XLS, …)	Boot	Script, mail, mail-mass
Breeding procedure developed	Ö	Ö	Ö	Ö	Ö
Analyser developed	Ö	Ö	Ö	**currently running**	**currently running**
Initial set of viruses breed	Ö	Ö	Ö	**currently running**	
Big set of viruses breed	Ö		**currently running**		
AVs have run on initial set	Ö	**currently running**	Ö		
AVs have run on big set	**currently running**				
Analyser has run on initial set	Ö		Ö		
Analyser has run on big set					

Preliminary results

The preliminary tests were run from July to August of 2001. The following systems were used for testing:

DOS program viruses:
    Intel Celeron processor @333MHz , 64Mb of SDRAM
    Microsoft Windows 98
Macro viruses:
    Intel Celeron processor @333MHz , 128Mb of SDRAM
    Microsoft Windows95 OSR2 4.00.950 B

DOS program viruses
There were a total of 227,760 virus samples from the 214 selected viruses. It means that the average number of samples for a virus is 1064.3. About 50% of these viruses are in the ItW, in the Standard, or in the Polymorphic Test-sets of Virus Bulletin.

Macro viruses
There were 17,391 word macro virus samples from the 1120 different viruses selected for the test.

The following anti-virus software were tested:
Table 2: Tested anti-virus software

AntiVirus	Version
Virobot Prof.	3.0
Norton AntiVirus	4.0
F-Secure AntiVirus	5.30.0
RAV Tray v.8	8.2.1.12
Sophos 3.49 beta	2.00 (Build 0117)
Panda AntiVirus Platinum	6.0

All of the mentioned anti-virus was downloaded about one week before the test so it means that the most recent and up to date version was used.

The purpose of the preliminary test was only to demonstrate that the testing procedures are workable and that the test can provide useful information for end users and developers as well. According to this purpose, it is does not matter which products were selected for the test.

Disinfection results of DOS viruses

The following table shows the number of bugs found according to disinfection procedures.

Table 3: Disinfection results on DOS viruses

	Panda	Norton	Virobot	F-Secure	RAV	Sophos
Number of changed files	88293	109275	29879	90002	48228	0
Number of changed files that totally equal to the original goat	71342	79248	25162	66468	33159	0
Number of files with ’Z’ flag	341	5013	57	4900	3134	0
Number of files with ’S’ flag	2728	874	663	1366	663	0
Number of files with ’E’ flag	881	0	0	1	555	0
Number of files with ’Y’ flag	884	796	15	1	1050	0

Number of files with ’o’ flag	5503	6955	2421	7005	5961	0

NOTE: Sophos AV does not deal with disinfection, so it cannot be used for disinfecting DOS viruses.

Disinfection results of macro viruses

The following table provides the results when testing each of the anti-virus products for disinfection of macro viruses.

Table 4: Disinfection results of macro viruses

	Panda	Norton	Virobot	F-Secure	RAV	Sophos
Number of files that include macro(s) after disinfecting	1234	1531	1535	2756	1546	1557

Summary

The low rate of virus detection may have occurred because in the cases the AV demo version was used. The regrettable occurrences of these cases highlight that anti-virus developers should deal more with quality engineering and testing. According to practical experiences adding detection and disinfecting algorithm of a new virus to an existing database can affect the behaviour of the whole program.

In the future we would like to extend this test as follows:

- Execute tests on other virus types.
- Increase the number of different viruses as well as the number of virus samples per a virus.
- Compare the test results of two or more AV programs developed by the same company.
- Make the whole procedure as automatic as possible.

References

Leitold, F. (1995). Automatic Virus Analyser System. Proceedings of the 5th International Virus Bulletin Conference, Boston USA, 1995, pp. 99-107.

Bontchev, V.(2001). Anatomy of a virus epidemic. Proceedings of the 11th International Virus Bulletin Conference, Prague, Czeh Republic, 2001, pp. 389-406.