Global Information Assurance Certification Paper

Anti-virus Software: The Challenge of Being Prepared for Tomorrow's MalWare Today
GSEC Version 1.2f
Lisa Galarneau

Introduction

Defending ourselves against MalWare – obviously a daunting task since depending on your preferred source of information there are between 30,000 and 63,000 known *"viruses"* out there! As discouraging as this may sound there is also no shortage of good advice on how to establish some lines of defense and how to recover if your defenses fail. One of the common themes in this advice is to select and deploy at least one and possibly more Anti-virus Software packages.

So just what is an Anti-virus package and what will it do for you? Is there a way to measure the efficiency of a package short of buying it, installing it and just crossing your fingers? And finally, in an environment where viruses are increasingly more sophisticated and more easily spread is Anti-virus software up to the task of defending us against tomorrow's threat today?

Setting the Terms of Reference

First, just what is MalWare? Gary Wiggins defines MalWare as:- " – short for malicious software. Software designed specifically to damage or disrupt a system, such as a virus or Trojan horse.[1] For purposes of this paper, the definition of MalWare also extends to Worms.

So then, does Anti-virus Software only provide protection from viruses? No, in actual fact today's software defends against viruses, Trojan horses and Worms – from the software's point of view they are all "viral" code.[1]

What Should an Anti-virus Software Do?

Effective anti-virus software needs to perform several functions. It must detect viral code with a high degree of reliability (minimal false positives or false negatives); it must prevent unwanted behavior associated with viral code and it must remove viral code and repair any associated damage to restore the protected system to its previous state. Further, the software needs to perform these functions using a minimum of system resources and without significantly impacting the performance of other applications. If your anti-virus software took two hours to run a scan each time you turned on your machine how long would it take you to turn off your virus scan?

Most software packages today use a mix of techniques to identify and deal with MalWare. These are signature scanning, integrity checkers, activity blockers and finally heuristics scanners.

---

[1] The author acknowledges that the definition and specific characteristics of viruses, Trojan horses and Worms differ, however, for purposes of this paper they will all be referred to generically as viruses or viral code – it's simply easier than beginning to call Anti-virus software "Anti-MalWare" software.

Signature Scanners

John McAfee is reputed to have written his first Anti-virus software on a bet – it was a simple scanner. Unlike the first scanners where a virus had to be fully identified today's scanners search for a specific set of bytes or 'signature string' believed to be associated with a specific virus; unless the signature string is within their database the scanner cannot identify the virus.

Signature scanners can differ in their degree of sophistication; some scan entire files others scan specific locations within files. They can generally be run on access or on demand and have the advantage of identifying a virus prior to program execution. It should be noted that both the speed and reliability of on access and on demand scans could differ within the same software package.

Signature scanners are highly effective in identifying and disinfecting known viruses; they have a low incidence of false positive reporting. They do, however, require frequent updates and they are less effective against polymorphic viruses.

Integrity Checkers

An interesting application of integrity checking within some Anti-virus software is the use of checksums to supplement signature scanners; once a scanner has identified a virus signature a checksum is run against the file and verified against the checksum for the known virus. This reduces the instances of false positives reported by the scanner.

Integrity checkers themselves do not actually detect viral code they detect changes in files and disks against a baseline measure. These changes can alert users to potential virus activity. User intervention is required to determine if the changes identified are genuinely the results of virus activity or are in fact legitimate changes. There are no associated, automatic, repair or disinfecting programs associated with integrity checkers as there are with signature scanners.

Integrity checkers do not require updates as signature scanners do. There is however significant user effort required in maintaining the database of baseline measurements for integrity checkers as well as to investigate potential virus generated changes. Integrity checkers can generate a significant number of false alarms. They also have some other significant drawbacks as virus detection tools. They alert the user only after a virus has run and modified files; they cannot specifically identify the virus responsible; and they are ineffective against macro viruses and viruses that do not infect program files. They are also ineffective against a specific type of virus called a slow infector virus that was specifically designed to bypass this detection technique and may also miss some stealth viruses unless they are run from a cold boot.

Activity Blockers

Activity (or behavior) blockers are memory resident programs that monitor for suspicious behavior such as the modification of executable files. User intervention is required to determine

whether the suspicious activity should be allowed to complete or not. This means that ultimately the decision to prevent (stop) a suspected unwanted viral behavior rests not with the software but with the user. Like, integrity checkers activity blockers are ineffective against macro viruses – they do however have a higher success rate against Trojan horses and Worms. Tunneling viruses are also able to bypass most current activity blockers.

Heuristic Scanners

Heuristic scanners much like signature scanners can be run on access or on demand. They vary widely in their scope and complexity. Some are able to disinfect and reconstruct files others are not. Heuristic scanners have a higher rate of false positives than do signature scanners but they have the significant advantage of being able to detect unknown viruses.

Heuristic scanners can be classified as static or dynamic. Static heuristic scanners make use of signatures in a similar fashion to signature scanners.[2] Rather than scanning for byte sequences associated with specific viruses static heuristic scanners scan for byte sequences specifically associated with simple virus like behaviors. These scanners are often supplemented with additional programs that seek out some of the more complex virus behaviors associated with polymorphic and encrypted viruses. These scans are associated with positive heuristics; negative heuristics on the other hand specifically seek out byte sequences for non-virus type behaviors. By applying both positive and negative heuristic analysis programs are able to reduce the incidence of false positives reported.

Dynamic heuristic scanners load suspect executable files into 'virtual machines' and emulate their execution. These scanners do not have a 100% detection rate. Some virus can successfully evade dynamic scanners due to specific sets of conditions within their programs (some sparse infectors for example could easily invade detection by the scanners). Dynamic scanning, as a rule, is much more time consuming and resource intensive than static scanning.

The behavioral information collected by the scanners is then analyzed or passed to an expert system[3] to determine with a reasonable degree of accuracy if viral code has been identified. The challenge in programming the analysis tool or expert system is to balance the need to identify unknown virus against the potential inconvenience and cost associated with reporting a false positive result.

Testing Performance

So now that you understand what an anti-virus package should be doing for you – is there a way to measure just how well it does before you pay for and install the software?

---

[2] By definition signature scanners are in fact static heuristic scanners – however they are not generally referred to has heuristic scanners in most industry literature.

[3] In a manner similar to updating signature files for scanners – Symantec also provides periodic updates to its "expert system" that analyzes data collected from heuristic scanning.

There are in fact several organizations that conduct tests on anti-virus software and publish results. Probably the best known is the ICSA (International Computer Security Association.) The ICSA has been testing virus detection software since 1992 and added criteria for virus removal in 1999. Certification requirements include (but are not limited to) 100% detection of WildList[4] viruses from the list published one month prior to the test; 100% detection rate on the ICSA's polymorphic test suite and no false alarms. The ICSA requires products to re-test every 60 days in order to maintain their certification.

Similar criteria are used by Westcoast Labs (Secure Computing) Checkmark program and by Virus Bulletin (VB) which has been testing anti-virus products and publishing results since 1989. The University of Hamburg VTC Malware Tests are also widely accepted.

As well respected as these bodies are there are some issues with their results. ICSA and Westcoast Labs charge a fee for certification – that means not all software packages are tested. Certifications using old WildLists can't keep up with new viruses.

Popular computer magazines also conduct periodic reviews of products periodic reviews of anti-virus software; they generally compare similar products with the aim of declaring a winner! Test conditions are generally less stringent with limited virus samples but the magazines are more likely to provide information on other features like cost, user friendliness and ease of obtaining updates.

In summary, if we recap our expectations of anti-virus functionality there are tests to cover most of the bases.

Requirement	**Tested**
Detect viral code (known)	Yes
Detect viral code (unknown)	No
Minimize False Positives	Yes
Prevent Unwanted Behavior	No
Remove Viral Code and Restore System	Yes – in some cases

It is important to remember when reviewing test results that test conditions will not have mirrored your individual infrastructure or the applications running on it. Further most anti-virus software packages allow some latitude for users to set preferences – this means the software may not perform in exactly the same way in your environment. Signature scanners and some analysis tools associated with Heuristic scanners require periodic updates – failure to update as required significantly reduces the effectiveness and reliability of the anti-virus package. Also, most of these tests do not have specified criteria for such items as speed, use of system resources or 'user friendliness'.

[4] Available at www.wildlist.org this is a list of viruses that are actually spreading worldwide. The list is updated monthly and is cumulative; viruses are added to the list if reported by two separate reporters. Viruses are removed from the list after 12 months without "sightings". The August 2001 list included 208 viruses known to be "in the wild".

Conclusions and Future Directions in Anti-Virus Software

There will never be an Anti-virus software that detects all known and unknown viruses with 100% accuracy all of the time – yet alone a package that will accomplish that and also report zero false positives. Software does exist; today that scans with a high degree of reliability for known viruses and to a lesser degree for unknown ones. If a virus slips past your scanners – activity blockers and integrity checkers can still alert you to suspicious activity or file modifications – potentially before too much damage has been done.

In terms of future directions there are some interesting theories on the table.

Joe Wells – creator of the WildList suggest that should be a decreased emphasis on the detection of zoo viruses. While the reported number of known viruses is numbered in the tens of thousands the most recent WildList (August 2001) listed only 208 viruses known to be in the wild. While the list should not be considered to be complete or up to date due to the time lag between reporting and posting that still means less than 1% of known viruses are actually likely to infect your computer or network. That implies a lot of resources being used to implement detection of zoo viruses in software packages and for users to scan for viruses that are simply not likely to exist outside of a laboratory environment.

A change such as Mr. Wells suggests also implies a corresponding change in the way anti-virus software is tested and rated for performance. Most tests today include a set of tests against 'zoo viruses'; in some cases performance against this set of viruses is the only key difference reported between products.

There is significant concern over the rate at which new viruses are being written and of the impact of increased interconnectivity. The rate at which a given computer can spread a virus and the number contacts a single computer can infect is increasing exponentially. This means the strategy of periodic distribution of virus software updates from centralized sources is becoming less and less viable – it is simply no longer fast enough (or soon won't be) to keep new viruses from reaching epidemic proportions.

Both McAfee and Symantec (Norton) discuss heuristics scanners at length in their marketing and promotional materials – continued advances in dynamic heuristic scanning promises improvements in the detection of previously unknown viruses and a further reduction in the rate of false positive reporting.

More importantly, McAfee's "AutoImmune" and Symantec's "Digital Immune System" are both modeled on biological immune systems. A key focus of both offerings (portions of, which have been deployed,) is to significantly reduce the time required to develop and deploy cleaners or cures for new viruses. McAfee's VirusScan product, on the market today offers some of the features of its AutoImmune technology. Symantec is pushing to automate the process entirely and has made a start with its Norton AntiVirus ™ Corporate Edition 7.6. Theoretical models of an automated "kill signal" are currently being studied. The signal would enable a computer once

it had detected a virus to send signals to its neighbors enabling them to immunize themselves against the virus or disinfecting them and prompting them to send the kill signal along to their neighbors.

These types of changes will also require significant changes in the way anti-virus products are tested. Without creating new viruses – how do you test the detection of unknown viruses? How large a sample of unknown viruses makes the test meaningful? How do you test the efficiency and speed of response of automated processes that collect new virus samples – distribute them for analysis and the distribute the required fixes?

The only certainty going forward is change but it is unlikely that anti-virus software will disappear anytime soon.

Appendix A – Definitions

Trojan Horse – "…a malicious, usually destructive program hidden within what appears to be an interesting or useful program, e.g., a spreadsheet, calendar program or a game. Argumentatively, some people consider a trojan horse a virus. Trojan horses, however, are not self-replicating. Rather they rely upon unsuspecting users to spread them." [2]

Worm – "A program or algorithm that replicates itself over a computer network and usually performs malicious action, such as using up the computers resources and possibly shutting the system down." [3]

Virus – "A program that reproduces its own code by attaching itself to other executable files in such a way that the virus code is executed when the infected executable is executed." [4]

Polymorphic virus – "Polymorphic viruses change themselves with each infection. These viruses are more difficult to detect by scanning because each copy of the virus looks different from the other copies." [5]

Slow infector virus – "A slow infector virus will only infect files when they are created or modified."[6]

Macro virus – "A type of computer virus that is encoded as a macro embedded in a document. Many applications, such as Microsoft Word and Excel, support powerful macro languages. These applications allow you to embed a macro in a document, and have the macro execute each time the document is opened. Unfortunately, according to some estimates, 75% of all viruses today are macro viruses. Once a macro virus gets onto your machine, it can embed itself in all future documents that you created with the application." [7]

Tunneling virus – "One method of virus detection is an interception program which sits in the background looking for specific actions that might signify the presence of a virus. To do this they must intercept interrupts and monitor what's going on. A tunneling virus attempts to backtrack down the interrupt chain in order to get directly to the DOS and BIOS interrupt handlers. The virus then installs itself underneath everything, including the interception program."[8]

Stealth viruses – "A virus must change things in order to infect a system. A stealth virus hides the modifications it makes. It does this by taking over the system functions, which read files or system sectors, and, when some other program requests information from portions of the disk the virus has changed, the virus reports, the correct (unchanged) information instead of what's really there (the virus). Of course, the virus must be resident in memory and active in order to do this."[9]

Sparse Infector Virus:- "This type of virus uses any one of a variety of techniques to minimize detection of its activity."[10]

End Notes

[1] Gary Wiggins, *Living with MalWare* (August 10, 2001) www.sans.org/infosecFAQ/malicious/MalWare.htm [October 16, 2001] (originally cited by Mr. Wiggins from www.webopedia.com)

[2] Kenneth L. Bechtel, II, *Viral Familiarization* (n.d) www.teamanti-virus.org/class97.htm [October 17, 2001], Definitions

[3] Gary Wiggins, *Living with MalWare* (August 10, 2001) www.sans.org/infosecFAQ/malicious/MalWare.htm [October 16, 2001] (originally cited by Mr. Wiggins from www.webopedia.com)

[4] CKNOW.COM Virus Tutorial, www.cknow.com/vtutor/vintro.htm [October 17, 2001], Virus Intro

[5] Gary Wiggins, *Living with MalWare* (August 10, 2001) www.sans.org/infosecFAQ/malicious/MalWare.htm [October 16, 2001] (originally cited by Mr. Wiggins from www.webopedia.com)

[6] CKNOW.COM Virus Tutorial, www.cknow.com/vtutor/vtfastslow.htm [October 17, 2001], Virus Types

[7] Gary Wiggins, *Living with MalWare* (August 10, 2001) www.sans.org/infosecFAQ/malicious/MalWare.htm [October 16, 2001] (originally cited by Mr. Wiggins from www.webopedia.com)

[8] CKNOW.COM Virus Tutorial, www.cknow.com/vtutor/vttunneling.htm [October 17, 2001], Virus Types

[9] Gary Wiggins, *Living with MalWare* (August 10, 2001) www.sans.org/infosecFAQ/malicious/MalWare.htm [October 16, 2001] (originally cited by Mr. Wiggins from www.webopedia.com)

[10] CKNOW.COM Virus Tutorial, www.cknow.com/vtutor/vtsparse.htm [October 17, 2001], Virus Types

Sources / Bibliography

(n.d. indicates no date)

Banes, David. "How to Stay Virus, Worm and Trojan Free – Without Anti-Virus Software." May 16, 2001. www.sans.org/infosecFAQ/malicious/virus_free.htm [October 17, 2001].

Bechtel, Kenneth L. II. "Viral Familiarization". n.d. www.teamanti-virus.org/class97.htm [October 17, 2001].

CKNOW.COM Virus Tutorial, n.d. www.cknow.com/vtutor/index.htm [October 17, 2001].

Drollinger, Dale. "Another Day Another Virus, So What Else Is New?" February 10, 2001. www.sans.org/infosecFAQ/malicious/another.htm [October 17, 2001].

Fiorentino, Jules. "Computer Virology: Protection, Prevention, Identification & Containment." November 22, 2000. www.sans.org/infosecFAQ/malicious/virology.htm [October 17, 2001].

Geffel, Chris. "defending Against the Next deadly Virus." December 20, 2000. www.sans.org/infosecFAQ/malicious/next.htm [October 17, 2001].

Gordon, Sarah. "What is Wild" Paper prepared for the 20th National Information Systems Security Conference. n.d. http://csrc.nist.gov/nissc/1997/proceedings/177.pdf [October 17, 2001].

Gordon, Sarah and Howard, Fraser. "Antivirus Software Testing for the New Millenium." n.d. http://csrc.nist.gov/nissc/2000/proceedings/papers/038.pdg [October 17, 2001].

Graham, John. "Issues with Keeping AntiVirus Software Up to Date." July 25, 2001 www.sans.org/infosecFAQ/malicious/antivirus.htm [October 17, 2001].

Gryaznov, Dmitry O. "Scanners of the Year 2000: Heuristics" July 1995. http://chekware.com/files/av/scan2000.txt [October 17, 2001].

Hruska, Jan. "Computer virus prevention: a primer." August 2000. www.sophos.com/virusinfo/whitepapers/prevention.html [October 17, 2001].

Kabay, Mich. "Today's focus: The low down on antivirus testing, part 1." Network World Newsletter. May 15, 2001. www.badguys.org/michtesting.htm [October 17, 2001].

McNeil, Jed and Milburn, Lisa. "Advanced Virus Detection Technology for the Next Millennium." Databus- Vol 40 No 2. February – March 2000. www.cedpa-k12.org/databus-issues/v40n2/nai.shtml [October 17, 2001].

Muth, Doug. Claws and Paws. n.d. www.claws-and-

paws.com/virus/articles/kinds_of_software.shtml [October 17, 2001.

Runnebaum, Joe. "The Need for Multi-layered Defenses on the Personal PC." November 28, 2000. www.sans.org.infosecFAQ/homeoffice/defenses.htm [October 17, 2001].

Sobers, Larry. "Anti-Virus Architecture: A 4-Layered Approach." October 31, 2000. www.sans.org/infosecFAQ/malicious/anti-virus.htm [October 17, 2001].

Solomon, Alan (Dr.). "A Brief History of PC Viruses. " September 1995. www.claws-and-paws.com/virus/papers/solomon_history.shtml [October 17, 2001].

Sullivan, Vincent. "Basic Virus Protection of the Future." October 24, 2000. www.sans.org.infosecFAQ/malicious/protection.htm [October 17, 2001].

Symantec. The Digital Immune System – Enterprise-Grade Anti-Virus Automation in the 21st century – Technical Brief. July 2001. www.symantec.com/avcenter/reference/dis.tech.brief.pdf [October 17, 2001].

Symantec. Understanding Heuristics: - Symantec's Bloodhound Technology – Symantec White paper Series Volume XXXIV. May 1998. www.symantec.com/avcenter/reference/heuristc.pdf [October 17, 2001].

Welsh, Geoffrey. "Virus Detection The Hard Way". August 6, 2000. http://home.inforamp.net/~crs0794/opinion/virus.html [October 17, 2001].

Wells, Joe. "Field Guide to Antivirus False Alarms." May 1999. www.cyber.com/papers/falseid.html [October 17, 2001].

WildList. www.wildlist.org August 2001. [Ocotber 17, 2001].

Wiggins, Gary. "Living with MalWare." August 10, 2001. www.sans.org/infosecFAQ/malicious/MalWare.htm [October 16, 2001].

Yang, Yi-Hua E. "Advanced Antivirus Techniques in Digital Immune Systems." Fall 2000. http://www-scf.usc.edu/~yeyang/657mid1/657mid1.htm [October 17, 2001].

Last Updated: July 27th, 2017

Upcoming Training

SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
Community SANS Indianapolis SEC401	Indianapolis, IN	Oct 09, 2017 - Oct 14, 2017	Community SANS
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event