Global Information Assurance Certification Paper

Noe Gonzales - 2002
GSEC, Practical Assignment v1.4, option 1

Virtual Sandbox Reality

Summary

As with all technologies, the computer has the ability to accomplish great feats of good as well as great feats of evil. Since the time of commonplace computer use in the mid-1980s, computer viruses have been in existence wreaking havoc amongst the computer community. Viruses are almost non-biased and still continue to be a nuisance to the newbies and savvy computer users alike.

In this paper, I will briefly look at the current status of antivirus software, the challenges that new viruses bring and the evolving techniques being developed to combat these new viruses. Ultimately, I will focus upon one such technology known as a 'Sandbox' and how sandboxing can be used in the future to help in real-time virus detection and eradication.

1. Introduction

I would almost bet that you know someone, perhaps even yourself, that has been afflicted. Perhaps you are familiar with some of the symptoms: a computer throwing up data as files get corrupted, data passing through the e-mail digestive system, the mind-numbing confusion of messages that seem to come out of nowhere, the feeling of needing a fresh start but perhaps never to boot again! As part of the evolving computer community user that you are, you may already know that you've caught a 'new-age' computer virus. Very much like a natural physical virus, these man-made computer programs replicate causing varying symptoms for the computers in which they use to breed.

Inevitably you may ask: Why me? There may be any number of reasons. First off, did you update your antivirus software? Are you like most home computer users who don't even use antivirus software much less keep it updated? Did you open that e-mail that could have contained one of quite a number of suspect file extensions? Maybe it was an e-mail from a friend of yours so you knew it was safe. Bad move if so. Looking at a better-case scenario, let's assume that your corporate laptop is updated weekly with the latest virus definitions. How did you still manage to catch a virus? Perhaps you caught one of the newest virus variations or maybe even a new strain.

Similar to physical viruses, computer viruses adapt to their changing environment. This is the purpose of this practical assignment, to look at the challenge of the modern viruses, and how we can catch and prevent the 'new-age' virus.

2. What are the challenges?

Considering the fact viruses are very damaging to business worldwide, not to mention home users, if it were easy to get rid of these menaces it would be done already. According to Fortune Magazine, “The Melissa virus outbreak cost about four days of productivity and did about $400 million in damage and losses, whereas the following virus outbreak, Love Letter, cost about five hours of productivity and did an estimated $8-$15 billion in damage and losses.“ [1] Please note that only two viruses were mentioned in this article but billions of dollars were spent in clean up. This may give you a small indication of how monumental this problem really is. Couple this information with the fact that although most of them do not make the headlines of your favorite virus news source, new viruses emerge daily.

2.1. How does antivirus software work now?

Have you wondered why antivirus software catches some viruses and not others? Current antivirus techniques stem from the antivirus researchers getting a copy of the virus. The virus may be obtained from ‘the wild’ or sometimes the virus programmer may forward the virus to the antivirus companies. The fact remains that the antivirus researchers have to get hold of the virus before they can began working on a cure. In this day and age there are other methods but as of yet this is still the de facto standard for the industry.

2.1.1. “The new signatures are here! The new signatures are here!”

Once a virus has been analyzed and decoded, a file signature (a.k.a. virus definition) is created. These are the files that you are so desperately trying to download once you hear of a threatening virus. To read Jeffrey O. Kephart and William C. Arnold describe file signatures, they write “Rather than requiring an exact match, the typical practice is to use just a small piece of the virus code as a means for identification. These short templates, called signatures, are much easier to handle, and reveal nothing useful to virus authors. There is an additional, very important advantage to using short signatures: they still work even when other parts of the virus change.” [2] File signatures do work but they are not the cure-all for the viruses that can corrupt hundreds of computers within minutes of infection. Antivirus companies have made improvements in the overall process but file signatures still cannot be created and delivered in minutes. The virus has to be captured, analyzed, have the file signature created and only then can it be pushed out to your computer. Oh, and did I mention the problems with that? No? I guess this is a good time to do so then.

2.1.2. Unmanaged vs. Managed

As most security conscious computer users know, humans are in large part the weakest link when creating a business computer network. As my experience has shown me, while all business computer users may have antivirus software, many users do not update the virus definitions (even if it's pre-scheduled to do so). There are some users who will even go out of their way to disable the antivirus software because it slows down their productivity. In doing so knowingly leaving themselves open to any number of attacks. Instinctively, this is something I always look for on client computers because I know that if I don't invest those few moments that it could possibly mean unwanted frustration and overtime for my salaried self. This is an example of unmanaged software. Basically, the user has the tools but the discretion on whether or not to use the tools has been given to him/her. This is not always the preferred scenario.

This option gives a network administrator control of pushing out file signatures to client computers thereby not depending on the end users' discretion. Most antivirus companies have a method to manage large numbers of business client computers. But again, the file signature has to have been downloaded from the antivirus company. Of these two options, I would definitely choose the managed software solution.

2.2. What makes viruses so hard to catch anyway?

File signatures work great for the common computer virus but not all viruses are common. "The more we advance the easier it is to advance" someone once said. Often new viruses are updated versions of old viruses so that a new virus definition has to be created. As it stands today, there are a plethora of known viruses that can infect your computer. As if it weren't bad enough, viruses are getting more complex in their attack methods as they constantly try to outwit the antivirus software.

2.2.1. Classifications

Viruses can be broken down into a number of major classifications [3]:

Boot Sector viruses: These virus types load into the Master Boot Record (MBR) of a floppy/hard disk allowing the virus to gain control of a computer.

File Viruses: These virus types reside within memory and mainly infect *.com, *.exe, *.drv, *.dll, *.bin, *.ovl and *.sys files.

Multi-partite viruses: These virus types are a combination of boot sector and file viruses.

Macro viruses: These virus types are application specific (not operating system specific unlike other virus types) and account for the fastest spreading and largest number of virus types.

Script viruses: These virus types are written in script languages, such as Microsoft Visual Basic (*.vbs) script and Java (*.js) script. They are often scripts embedded into HTML files becoming active when a web page is being viewed. Script viruses should not be confused with ActiveX or Java applets that require already downloaded components.

Worms: These viruses replicate with or without user intervention. Instead of spreading from file to file, they spread from computer to computer since they are network aware.

Hybrids: Although not considered to be viruses, viruses can contain other programming code such as pranks or trojans. Trojans do not replicate, but can open up back-doors to your computer. Perhaps allowing it to be remotely 'zombied' at a later time by a 'cracker'.

2.2.2. Characteristics

Now that we have covered the basic types of viruses, let's discuss the traits that give a virus its 'personality' per se. Major virus characteristics are as follows [4]:

Memory-resident: The virus enters the computer's memory where it can easily replicate itself into programs or boot sectors. This is the most common type of virus in the wild.

Non-resident: Also known as Direct Action Infector. This type of virus does not stay in memory after the infected program is closed. This virus can only infect other programs, files or boot sectors while the infected program is open. This type is not common in the wild.

Stealth: The ability to hide from antivirus detection. This type manifests in two ways- Full Stealth) The virus can redirect disk reads to avoid detection and Size Stealth) This type will alter disk directory data.

Encrypting: Technique of hiding by transformation of program code. Virus code converts itself into cryptic symbols. To execute and spread the virus, however, the virus code will have to be decrypted. At that point the antivirus program will detect it.

Polymorphic: Ability to mutate by changing its code segment to look different from one infection to another. This type of virus is a challenge for antivirus detection methods.

Triggered Event: An action built into the virus that is set off by specific criteria (a date, time, key press or key sequence, DOS function).

Antivirus disabling: A technique used by viruses to disable antivirus software to avoid detection.

IP Spoofing: A technique used to hide the origin of the 'attacking' computer.

Tunneling: An ability of the virus to load itself under the antivirus software layer and closer to the hardware layer thereby evading detection [5].

3. What does the future hold? Did somebody say 'Sandbox'?

Having covered the different virus types and characteristics that have to be dealt with, we have a better idea of what we need to cope with these viruses.

In an effort to keep up with the ever-evolving virus, the antivirus companies have also been working on updating their scanning techniques. One way to do is through the use of Heuristics. As defined by Webopedia, "Heuristic programming is characterized by programs that are self-learning; they get better with experience. Heuristic programs do not always reach the very best result but usually produce a good result." [6] Another method that we may see become more commonplace is to use virus scanning technology at points-of-entry such as firewalls, gateways and routers in addition to the current standard of residing on the mail server and computer clients.

One up-and-coming method of Heuristics is known as a 'Sandbox'. This is how the Norman antivirus company describes this technique: "'Sandbox' describes the technical solution; the program establishes a simulated computer in an enclosed area, allowing the virus to replicate on the simulated machine under careful monitoring, when the virus has been activated, the sandbox is examined and the vaccine is produced immediately. This 'sandbox technology' should not be confused with traditional heuristics." [7] Traditional heuristics looks for certain known code patterns within programs to determine if the code is actually a virus. This traditional heuristic method often results in false alarms since sometimes legitimate code can be flagged as a virus.

3.1.1. Similar concepts, Different technologies

The Sandbox theory is not new to computers but is new in the fact this it is aimed towards antivirus technologies specifically. This technology is already being used by the Java machine and Honeypots.

The Java virtual machine is a technology that restricts functionality from untrusted sources. It does this by running downloaded Java code from within the Java machine's sandbox. In doing so the Java machine keeps code from

maliciously tampering with a computer.

Honeypots are another form of sandbox technology. They are currently employed 'in-the-wild' to discover new methods of network attacks. With honeypots, computer networks are purposely staged hoping to be compromised by a cracker. Once compromised, researchers can then study all of the cracker's activities to determine if new techniques are being used. If that is the case, the knowledge is then passed onto the whitehat community (and unintentionally to the blackhat community as well).

The time has now come for sandbox technology to be used in eradicating viruses.

3.1.2. Sandbox Requirements

In order to make sandbox technology a reality, hardware virtualizations must be met to create an environment that is used to fool the virus [8]:

Storage Device: Fixed disks of any size or layout. Will be initialized with a file system (logical drive C), including system and goat files (binary executables) prior to the bootstrap process.

IO Manager: Will handle all I/O requests, and will also have access to RAM.

ROM: Read-Only-Memory. Used to store BIOS code.

RAM: Initialized to a known value before bootstrap. Memory below 1 MB is paged linear, and memory above 1 MB is paged using linked lists.
User: We need a virtual user to be there to control mouse and keyboard activity. Sample values are pre-inserted into the keyboard.

CPU: An emulator, that can emulate any instruction including Multi Media eXtensions (MMX), floating point unit (FPU), etc. Flag handling, interrupt/exception/fault handling is also done by the emulator.

In addition to hardware requirements, the software environment must be tailored as much as possible to suit the particular virus being studied. Since viruses run on a number of OS platforms including but not limited to DOS, Windows, OS/2, Apple and Linux, the sandbox should be able to emulate these environments as well.

The more sensors that can be placed within the virtual environment while still maintaining the look and feel of a non-virtual environment the better. This is more crucial with honeypots since humans have the ability to question and then,

hence, investigate. As of this writing, viruses are not that smart yet.

3.1.3. Advantages

As you can see the, sandbox methods give us quite a utility in fighting viruses and have quite a number of advantages over other technologies. I'm sure it will prove to be a needed utility for quite a time to come.

3.1.3.1. More True Alarms

As was mentioned earlier, sandboxing is a newer form of heuristics. The problem with traditional heuristics is that sometimes in looking for bad code, the heuristic software sometimes flags good code as being tainted which is an undesired result. Also, traditional heuristics use a pre-defined set of rules, similar to virus file signatures in a sense, that are used to identify virus code. However, virus code is not confined to these sets of rules and the more complex viruses tend to think out-of-the-box in their exploits thereby evading detection.

However, with sandboxing, a virus is not analyzed by its code. It's analyzed in part by its characteristics, which results in fewer false alarms. This is another advantage of sandboxing since honeypots have been taken down before by advanced crackers who discovered they were being watched.

3.1.3.2. Looking into the crystal silicon ball

One of the best features of sandbox technology is that protection happens in real-time. When the virus is captured, it is analyzed right there and then. Looking towards the future, the Norman antivirus company writes

> "Imagine a mail coming through a gateway system. It holds an attachment. The attachment is 'virtually executed' in the sandbox, and is stamped as a virus. The scanner engine will immediately start working out detection & disinfection. If it succeeds, the attachment will be cleaned, and the mail continues through. If it can't, it's deleted. The next time the same attachment comes through, the scanner has a signature to detect it, and already knows whether it can be disinfected or not. Everything necessary to do so is already located on the machine – nothing needs to be transmitted to any lab for analysis." [8]

Even though this scenario applies to an e-mail attachment, it could eventually be applied to every day common occurrences of software downloading. Using a sandbox may also eliminate the need to filter out common questionable file types that also have legitimate business use in the corporate world.

3.1.4. Disadvantages

While sandboxing seems to be the answer to virus eradication, there are some shortcomings that have to be dealt with. As with most software, sandbox technology is in the gap between functionality and usability.

3.1.4.1. Oops, I'm human

One of the areas for concern in sandbox technology is that it is a program written and used by humans. As most programmers know, it's easy to get a program to work but takes much more time and effort to make it unbreakable. Even with dedicated systems being used solely for sandboxing, the chance for errors in programming still exists even if minute. Even if errors don't exist, will the computer still have resources left after running and analyzing every file that comes attached? Or will needed processing power be another reason to get the latest CPU chip?

3.1.1.2. Holy are thou

Even though sandbox technology exists for the computer, this is a two-fold problem. First you need to get the virus 'injected' into the computer that has the sandbox otherwise non-protected computers risk infection and destruction. Secondly, if a virus runs on the client of a non-dedicated system, there is always the possibility that it can run outside of the sandbox. In the lab, researchers 'reset' the victimized sandbox computer after analyzing a virus making sure to cleanse all configuration files, memory values, binaries and so on. This makes sure they get a disinfected computer in which the starting condition is a known state and not a variable state. In the real world, root and administrator type accounts require the ability to install software. Otherwise what good is a computer? And history shows us that these accounts can be compromised as well.

4. Conclusion

Few things are certain but I'm sure the virus vs. antivirus fight will continue for quite awhile. Sandboxing is moving into a new era of overall virus technology with proven effective concepts. While trying to be more proactive than reactive, time must be of the essence. As the viruses learn how to live and adapt so must we learn how to study them and strive for solutions.

Still there are questions which lead to more questions: When the sandbox resides on the actual client computer where today's users have some freedom to install programs, does it then have the fate of the Java virtual machine? Does OS development need to change for sandboxing to work without fault? Do we need self-repairing computers? [9] Does the average business computer user acknowledge the fact that his/her business computer belongs to the company? What about the home computer users? Is the sandbox client in the gray area of taking control and taking too much control? Or is it paving the way with its sister

technologies of a new computing future? If virtual systems make it to the client what else could be captured? These are questions that may be answered sooner than I think.

Sandbox technology alongside other developing technologies such as Behavior Blocking and the Auto-Immune system [10], should prove to be quite effective. Perhaps the lines between them will blur in the future to help us come up with the ultimate fighting machine against the 'living' virus. One veteran researcher Jeffrey O. Kephart stated it as follows, "To address these problems, we have designed an immune system for computers and computer networks that takes much of its inspiration from nature. Like the vertebrate immune system, our system develops antibodies to previously unencountered computer viruses or worms and remembers them so as to recognize and respond to them more quickly in the future. We are careful to minimize the risk of an auto-immune response, in which the immune system mistakenly identifies legitimate software as being undesirable. We also employ nature's technique of fighting self-replication with self-replication, which our theoretical studies have shown to be highly effective." [10]

Just keep in mind that we are only scanning for viruses within these virtual systems at this point. As it stands today, to be secured in the networking world, there still needs to be multiple defense perimeters established against the various attach techniques. Sandboxing is just one technology to be familiarized with in the security professional's toolchest.

References

[1] Gabon, Timothy. "Security ROI - Payback for Comprehensive Protection." URL: http://www.fortune.com/sitelets/sections/fortune/tech/2002_07security.html (July 11, 2002).

[2] Jeffrey O. Kephart and William C. Arnold. "Automatic Extraction of Computer Virus Signatures." URL: http://www.research.ibm.com/antivirus/SciPapers/Kephart/VB94/vb94-node2.html (July 11, 2002).

[3] Author Unknown. 'Virus Info Center – Virus Primer." URL: http://www.trendmicro.com/pc-cillin/vinfo/vprimer_virus_type.htm (July 12, 2002).

[4] Author unknown. "Norton Antivirus Information Brief." May 2000. URL: http://www5.pc.ibm.com/ww/me.nsf/WW-webdocs/A6A4836D68B269A585256A9D006E9802/$FILE/xbnortf.PDF (July 14, 2002).

[5] Author unknown. “Tunneling Viruses.”
URL: http://www.cknow.com/vtutor/vttunneling.htm (July 15, 2002)

[6] Author unknown. “Heuristic Programming.”
URL: http://www.webopedia.com/TERM/h/heuristic_programming.html
(July 15, 2002).

[7] Author unknown. “Patenting Deep Scan and Sandbox technology; Norman introduces a new technique for eliminating new computer viruses.” October 25, 2001. URL: http://www.norman.no/press_release/2001_oct_25.shtml (July 12, 2002).

[8] Natvig, Kurt. “SANDBOX TECHNOLOGY INSIDE AV SCANNERS.” Virus Bulletin Conference. September 2001.
URL: http://www.norman.no/documents/nvc5_sandbox_technology.pdf
(July 12, 2002)

[9] Stang, David J. “Fighting Computer Virus Infection through Auto-Immune Responses Applying Principles of Life to Anti-Virus Technology.”
URL: http://vx.netlux.org/lib/ads01.html (July 14, 2002)

[10] Kephart, Jeffrey O. “A Biologically Inspired Immune System for Computers.” Artificial Life IV, Proceedings of the Fourth International Workshop on Synthesis and Simulation of Living Systems. 1994. URL: http://www.research.ibm.com/antivirus/SciPapers/Kephart/ALIFE4/alife4.distrib.html (July 14, 2002).