The Progression of Modern Malware and Necessary Behavioral Changes

Author:
Eric Bailey
eric.bailey@tufts.edu

Mentor:
Ming Chow

Abstract

Many users today think that if they download Antivirus software, they are immune to any malware-based attacks. Further, there are those that use Virtual Machines as a sort of proxy between the virus and the host computer and believe that they are completely safe from potential harm, but even this is not the case. In fact, over 79% of malware distributed in 2014 attempt to bypass the Virtual Machine proxy and 88% use further techniques to combat modern malware detection (Barbosa and Branco). Even large Antivirus companies are still in the eternal struggle with malware producers who are vigilant yet silent. Everything we know, they know—but not vice versa. Because of this, the only guaranteed defense one can have against the wills of the malicious is changing one's actions, not one's software. Currently, there is too much trust on abstraction regarding computer safety. While Antivirus and Virtual Machines can win some battles, they cannot win the war. The purpose of this paper is to educate users of the common pitfalls of malware prevention in 2014, and how to stay vigilant in the future as malware keeps evolving.

Introduction

Malware

Almost everybody who will read this paper knows *what* malicious software is, and have probably been affected by it. However, a large fraction of that group will also have grown complacent in their actions, such as browsing the Internet, installing software, and just generally trusting programs to do what they say they're going to do. This tends to be the case because they may trust their old Antivirus (AV) software, and they have not dealt with malware directly in a number of years. Despite years of experience and relative knowledge of the software environment, many who think they are protected, are in fact not.

Antivirus Software & The Pitfalls of Malware Analysis

The idea of AV software is great! A piece of software that you install and you're completely protected from any harm? What an awesome concept! However, it's not that easy in practice. Unfortunately for AV developers and users, none of the true innovations in malware are open sourced by malware developers, so the only real thing researchers have is observing new/changing malware programs in the wild. Finding large amounts of these programs, determining that they are malicious, and creating AV software to reliably protect against it is certainly a nontrivial problem. Two of the major ways to decide whether or not a program is malicious are dynamic analysis and static analysis.

Static analysis is the process of directly analyzing machine instructions or source code of a program to make inferences about the program. Unfortunately, because legitimate modern malware is very secretive and closed-source to the public, the only option researchers have for modern malware static analysis is actually looking at object dumps or binaries. As one would expect, this is much harder than looking at the source code directly because of the ambiguity caused by low level code, the sheer amount of instruction permutations you need to parse through, and other reasons we will discuss later.

Even though it is possible to do static analysis well (and in fact one of the most reliable ways to analyze malware), malware producers do not make it easy (Branco). Resistance from malware producers has made static analysis less reliable. One method of resistance is packing: Packers change the binary code of a program so it can perform the same instructions (like a .zip file), without being detected by static analysis systems, which look directly for specific lines of assembly code. Packing *malware* is often done pseudorandomly, so even one program with the same source code compiled ten different times might be detected only a handful of times by many AV programs (Cannell). Packing, intentional code obfuscation, and novel ways to hide programs' intentions often lead to a number of false positives by AV systems using static analysis.

Another problem with static analysis is how slow it is. With the sheer amount of instruction permutations that assembly programs create, analyzing all the possible combinations for suspicious sequences of instructions just takes a long time. It certainly does not help that over **400 million new variants of malware** are being produced year - that's over 1 million per day. With such a large sample size of files, gather and analyzing data would demand an impossible amount of time (Shinotsuka). Thus, anti-malware research companies are forced to categorize potential malware programs into different groups, which certainly leaves room for error. A potentially faster solution static analysis is *actually running the malware* and watching what happens.

Dynamic analysis is the process of running a program and taking note of what it does. Some immediate issues come to mind when one talks about running malware on company machines. First, that's exactly what the malware producers want—they want their programs to be run on as many machines as possible. Also, malware programs can easily fool naive dynamic analysis systems; for example, they can choose to go dormant until the system reboots *k* times, then start running malicious lines of code. Therefore, if this program is run once by a dynamic analysis-based system, it wouldn't do anything questionable and would consequently be incorrectly flagged as not malware (Wueest).

Often, AV companies will perform large-scale analyses in a Virtual Machine (VM) environment. That way, if the machine does get infected by malware, the company can theoretically revert to a previous image, and run large batches of malware programs en masse (supposedly) without posing a threat to their actual systems and networks.

Virtual Machines and Host-VM Connections

With cloud computing being the "next big thing" in large-scale software companies, companies are relying more on large virtual networks of virtual machines talking to one another. However, the rapid implementation of VM networks creates many new attack vectors for malicious programmers. Too often, companies take implementation speed and product relevance over security concerns. We are seeing this to be the case in these companies adopting large-scale cloud computing architectures. According to Symantec employee/malware researcher Candid Wueest, "Virtual machine hosting servers are not any less secure than any other type of server - they are just as vulnerable to malware or targeted attacks."

So, what's the problem with VMs? Isn't the advantage of using a VM network the fact that you can reset any part of the system to a previous snapshot if something goes wrong? Again, it's not that easy. First, there still could be a dormant virus lurking in the shadows, even after imaging a supposedly clean VM. Perhaps more threateningly, malware producers have found multiple ways to bypass the VM firewall and only target "real" computers through a method called VM detection. This allows any program to tell whether or not it is being run in a virtual environment or on a legitimate computer/server. Though a seemingly difficult task, there are a number of surprisingly successful methods which require only a few assembly instructions (see: http://vrt-blog.snort.org/2009/10/how-does-malware-know-difference.html or *Supporting Material* at https://github.com/popcorncolonel/comp116-ebailey/finalproject). Some of these methods are outdated by a few years, but malware gets more advanced linearly with AV software. One of the scariest parts about this is that there are most likely VM detection algorithms in use that *researchers have yet to discover*.

VM detection makes it harder for AV companies to isolate malicious code, as a virus can behave differently on VMs and "true" machines. Because it looks suspicious for a program to automatically close in a virtual environment, certain malware programs in VM environments have also been known to feign legitimate instructions for a set period of time, and then quit (Wueest). However, AV researchers are working on ways to intelligently flag programs that do this as well. Again, it's a constant struggle between malware creators and malware researchers.

Not only do many malware programs shroud themselves in the presence of VMs, but also it is actually possible for malware to **move between host and virtual machine**—especially if there is a network bridge, or shared files between the two systems. This is one of the scariest truths for those experimenting with malware; especially large software companies. Even as far back as 2009, malware creators were already making programs and techniques to escape from virtual machine to host via an attack called called Cloudburst (Kortchinsky), which will be discussed more in-depth later.

The only thing to keep a VM environment and entire system/network truly safe is to be as cautious in your computing as you would without any Antivirus protection or a virtual machine crutch. In many cases, virtual machines and host computers are effectively two different computers in the eyes of the overall network. Network worms are still very much able to be transmitted over the network, so users should definitely still treat their virtual environments as they would their actual work environment (Wueest).

Not only is VM detection a big problem for Antivirus companies, there are many small-scale VM users nowadays, who primarily use a VM to browse questionable/potentially unsafe websites or open up files that might harm their "real" machines. Unfortunately many people do this under the false assumption that VMs form this barrier of total protection for your host computer. This is definitely not the case. Immunity, Inc. employee Kostya Kortchinsky says, "[Virtual environments aren't] an additional security layer – [they're] just another layer to find bugs in." In this regard, VM detection and actions by malware create huge problems for both individual users and research companies trying to learn more about the state of malware or risky programs, on a distributed scale.

A discussion of short-term programmatic solutions to the VM detection problem (rather than changing the way you compute) is found in the *Defenses* section.

To The Community

Most of the people reading this paper will understand why evading malware is important. Why I chose this as my topic may not be as clear. It's rather simple: people (in general) are far too complacent in their Internet browsing and rarely take into account the amount of changes in technique that malicious programs often undergo, as exemplified by the recent past.

Data and Statistics

Since most individual malware infections are fairly insignificant and hard to detect en masse, explicit examples of individual infections may not be too interesting. Thus, large-scale data is where interesting trends start to arise regarding the massive amounts of malware being produced, taken in, and analyzed today. For example, the most state-of-the-art antivirus companies and malware researchers are taking in samples of sizes in the millions. This is especially the case when looking at changes in malware over time.

Malware researchers have been working hard to gather as much data as possible. Recently (in 2014), over 16 million programs were analyzed, taking researchers weeks to gather parse, and analyze all the data. Static analysis was primarily used for detection as dynamic analysis on malware has too many complications and too much room for error. Here are some of the most relevant malware statistics to keep in mind when using a machine nowadays:

-79.7% of malware programs employ Anti-VM techniques.

-88.9% had at least one Anti-Reverse Engineering technique (including Anti-Debugging and Code Obfuscation) to fool researchers using static analysis.

-5.4% immediately go dormant (using the system call sleep()), attempting to fool researchers using dynamic analysis, or confuse affected users.

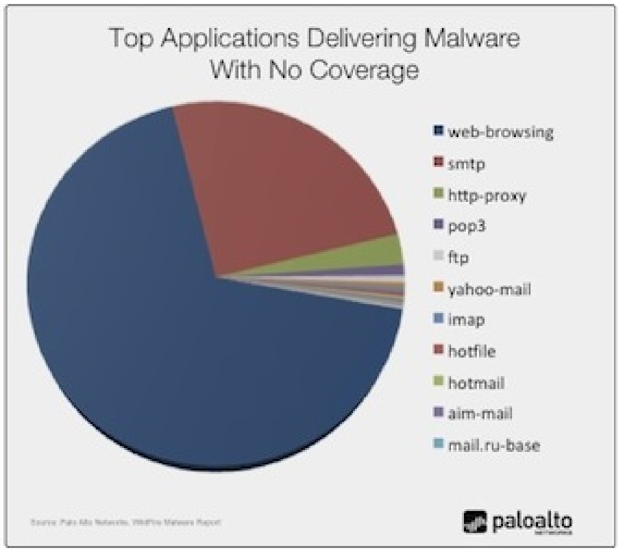
-37.5% are packed, in an attempt to fool AV systems.

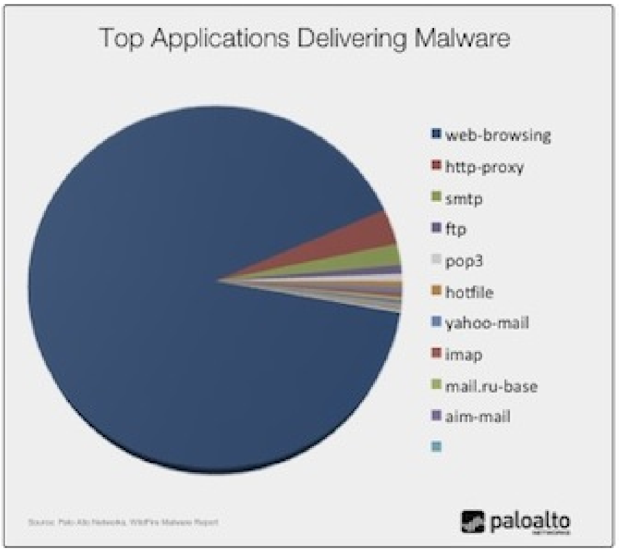
(Barbosa and Branco)

-Over 68% of malware cases originates from browsing the web (under 25% from email).

-About 40% of malware transferred across the web cannot be detected by “industry leading” AV programs.

-Nearly 95% of malware samples remain undetected by antivirus after 31 days.





(Rashid)

Defenses

If Antivirus software doesn't work 100% of the time, and I'm not protected by hiding behind a Virtual machine, what can I do to still be able to browse the Internet safely? Just like biological viruses, computer viruses mutate over time based on what advances the defense systems are making. It's always a game of cat-and-mouse. We may never cure the common cold, and while we haven't, we must be cautious in our actions (don't hang around sick people, don't go to another continent without vaccinations, etc). Running questionable programs in a VM is much like wearing a surgical mask in public—it can't definitively cure illness, but it can work as a filter and help in a large number of scenarios.

What can really we do?

Antivirus programs and virtual machines have some clear flaws (namely, their uncertainty and imperfection). It may sound boring, but the only way to be sure you are safe is to be careful in what you do. Don't go around downloading freeware hastily. For all you know, it could be a novel anti-AV mechanism to breach your defenses. Educate people of this problem; people are far too complacent in their browsing, and they don't even know what the problem could be. Even if you are cautious to an extent, realize that your machine and network cannot be malware free permanently. Take constant maintenance of your system. Take into account that your system could be infected right now. Keep a close eye on the progression of modern malware, and how you need to change to stay safe.

One of the most important things is to, keep your software up-to-date. Hypothetically, suppose your system is immune to malware under the current knowledge of malware. Even though you have the most robust anti-malware system that has ever existed, it almost definitely will not be good enough in a few years. Malware is constantly changing. Keeping software up-to-date is very important, even if it may be a pain to have to do this constantly. Take, for example, the 2009 Cloudburst attack, which allowed the unspeakable: malware programs could escape VMware environments onto the host machine, exploiting certain vulnerable VMware settings and aspects of Windows visual APIs (Kortchinsky). If you were unaware of this or didn't regularly patch your system, **you could still be vulnerable to this attack**, over five years later!

Now, if you do decide to actually use a VMware virtual machine, you should set the following settings in your VMware's configuration just to be safe. Again, any protection you use **cannot** make your system/network completely secure (zero-day vulnerabilities, state-of-the-art advances in anti-VM methods). However, it can act as a filter and drastically cut down on the number of potential anti-VM techniques that will cause weird stuff to happen. So, before you start up a windows VM in VMware, be sure to enable these settings in the virtual machine's .vmx file:

```
isolation.tools.getPtrLocation.disable = “TRUE”
isolation.tools.setPtrLocation.disable = “TRUE”
isolation.tools.setVersion.disable = “TRUE”
isolation.tools.getVersion.disable = “TRUE”
monitor_control.disable_directexec = “TRUE”
monitor_control.disable_chksimd = “TRUE”
monitor_control.disable_ntreloc = “TRUE”
monitor_control.disable_selfmod = “TRUE”
monitor_control.disable_reloc = “TRUE”
monitor_control.disable_btinout = “TRUE”
monitor_control.disable_btmemspace = “TRUE”
monitor_control.disable_btpriv = “TRUE”
monitor_control.disable_btseg = “TRUE”
```

Luckily, many of the basic VM detection methods rely on the default of these settings being false (Cannell), some of which are discussed in the supporting material for this paper.

Conclusion

That the problem of malware is by no means solved. Even with robust protection, there is always the lurking chance of new innovations because of how closed-source the nefarious malware producers tend to be. The only true way to make sure your computer is secure is to make sure your computing habits are secure. No amount of software can protect from *everything*. Even with defensive firewalls/filters (Antivirus systems) and proxy technologies (virtual machines, proxy networks), there are so many breaches to combat these defenses. The only true way to be safe is to be vigilant of mutations in modern malware techniques and anti-malware software, and to be careful with what you do.

References

Barbosa, Gabriel; Branco, Rodrigo; and Neto, Pedro. "... Overview of Malware Anti-Debugging, Anti-Disassembly and Anti-VM Technologies." *Vulnerability & Malware Research Labs*. July 2012.

Barbosa, Gabriel and Branco, Rodrigo (of Intel). "Prevalent Characteristics in Modern Malware." Presentation at *BlackHat USA 2014*. Las Vegas, NV. August 2-7, 2014.

Cannell, Joshua. "A Look at Malware with Virtual Machine Detection." *Malwarebytes*. February 6, 2014. Web.

Cannell, Joshua. "Obfuscation: Malware's Best Friend." *Malwarebytes*. March 8, 2013. Web.

Ferrie, Peter. "Attacks on Virtual Machine Emulators." *Symantec Advanced Threat Research*. 2006.

Kortchinsky, Kostya. "Cloudburst." Presentation at *BlackHat USA 2009*. Las Vegas, NV. June 2, 2009.

Rashid, Fahmida. "Modern Malware Increasingly Using Real-Time Web to Evade Detection." *Security Week*. March 25, 2013. Web.

Shinotsuka, Hiroshi. "Malware Authors Using New Techniques to Evade Automated Threat Analysis Systems." *Symantec*. January 23, 2014. Web.

Wueest, Candid. "Does Malware Still Detect Virtual Machines?" *Symantec Research*. August 12, 2014. Web.

Wueest, Candid. "Threats to Virtual Environments." *Symantec Research*. August 2014.