Running head: REVERSE ENGINEERING MALWARE

Reverse Engineering Malware

Shaun DeRosa

University of Advancing Technology

NTS-222

Abstract

In today's world of constant technological advancement, society must be aware of the simultaneous advancement in malicious programs known as malware. Behind the scenes this malware, also known as viruses, worms and trojans continues to evolve along with the rest of technology. Currently one of our best defenses comes in the form of commercial software known as anti-virus programs. Anti-virus programs search files for "signatures" of known malware, the program breaks the file down into its assembly code and searches for these signatures. In an attempt to circumnavigate the efforts of anti-virus software, malware designers have created programs called "polymorphic viruses" that change each time they are run. Anti-virus developers and others believe reverse engineering may be a useful tool in combating malware and its future incarnations. Reverse Engineering "…can be defined as analyzing and disassembling a software system in order to understand its design, components and inner-workings." (Rozinov, 2004, p. 3)

Reverse Engineering Malware

The purpose of this paper is not to reverse engineer any specific malware, but to examine the methods used to do so. Reverse engineering in its most simple of definitions is to disassemble and examine a program in an attempt to understand not only its effect on its host machine, but the ideas and processes used to create the program in the first place. Where as current anti-virus methods search a program's code looking for a specific signature associated with known malware; reverse engineering takes a much more detailed and thorough approach.

With new advancements in information security come chances for attackers to find and exploit new vulnerabilities. Attackers create new ways to adapt and overcome new security measures and anti-virus scanning techniques. Security professionals have begun using reverse engineering as a way to try and be proactive as opposed to reactive in their attempts to stop attacks. If developers can use reverse engineering to understand the current techniques being used to exploit their software, they can try and predict the evolution of the attackers' methods and patch a possible vulnerability before it is ever exploited.

Method

The first and most important step in reverse engineering any type of malware is to create a safe and controlled environment where the malware can be observed without the possibility of infecting other hosts. The most efficient and reliable method is to use a virtual workstation; examples include VMware and Microsoft Virtual PC. Other tools used in the process include debuggers, disassemblers, hex editors, network sniffers, and compilers. There are many other utilities that may be used depending on the specific program being studied.

Once the virtual environment is created and the tools gathered, the malware in question is loaded into the environment. Locating a copy of a specific virus, worm or trojan is simply a matter using a search engine on the internet; on the more well known viruses the search can take

as few as five minutes. It is surprisingly easy to locate malware on the internet, however there are also websites dedicated to the study of malware where you can search for your target virus, worm or trojan. One such site, http://vx.netlux.org, contains an incredible amount of information on the subject, from source code to polymorphic engines and beyond.

There are two different modes of analyzing the software, static and dynamic. Static analysis consists of studying the program without executing the program. Information such as how the malware acts inside of the operating system before it is executed as well as if virus scanners can detect the dormant virus, can be determined using static analysis techniques. Dynamic analysis is the monitoring of the program once it has launched and has begun performing processes. This is where great deals of information about the design of the program can be found. Once the program has been executed tools can be used to determine factors such as what ports it may be using to communicate, how it replicates and spreads itself, what local resources it uses and what local files it may use or damage to name a few. After observing and noting the actions taken by the program when run using the monitoring tools, the next step becomes breaking the program down into its source code to try and discover the inner workings of the executable.

To analyze the source code of the malware, researchers use a debugger and disassembler. Tools I often found in my research that are used for these purposes include DataRescue IDA Pro for disassembly and OllyDBG for debugging. Using these techniques and tools the malware can be analyzed at its root, known as assembly instructions. Running the program through the disassembler, each line of the assembly language can be seen as the program executes it. The debugger's job is to help sort through all of the assembly language by providing breakpoints, which can be inserted at certain points in the code, effectively halting the program at that point and allowing researchers the chance to isolate and study specific lines of code. Careful analysis

of this code and its construction can lend insight into the tendencies and habits of the attackers creating these programs.

In order to try and observe the actions of the malware from an external viewpoint other utilities are employed. Running "sniffers" such as Snort or TCPdump can help collect data concerning what information the program is sending over which ports and to whom it is sending it. These sniffers intercept packets of data as they are being sent by the malware over the network, allowing researchers to understand what information the virus may be targeting. Other utilities that monitor the effects of the virus can be run from clean systems on the network to try and locate vulnerabilities that may have been created by the malware. A tool found at www.insecure.org named NMap Security Scanner scans target computers looking for open ports that could be used by an attacker to send and receive data. Similarly www.nessus.org is the home of the Nessus Vulnerability Scanner, "…started by Renaud Deraison in 1998 to provide to the internet community a free, powerful, up-to-date and easy to use remote security scanner." (www.nessus.org), the program will scan a computer for backdoors and listening ports that may have been created by the infecting malware.

Results

The techniques, theories and approaches to reverse engineering malware are as varied as the malware itself. Tools and utilities that may be crucial to examining a specific trojan may not be as efficient or helpful when examining a specific worm. The essence of reverse engineering malware is the same regardless of the methods used; to try and better understand the threats to information security and develop defenses against them.

Everyone who has ever driven a car knows that if they depress the gas pedal the car will move forward, much the same way the majority of computer users know the click of a mouse

button can transport them through the internet. If however, a driver stepped on the gas only to have the horn and lights activate, the car would be brought to a mechanic for repair. One step further in the process is knowing how and why the problem occurred in the first place and how to stop it from happening again, this process of breaking down all of the components in the car and analyzing how they interact with each other to find the root of the problem is the basic idea behind reverse engineering. In essence it is the process of taking a problem and working backward to break it from its whole into all of its components to better understand its actions, its methods and its weaknesses.

Discussion

Reverse engineering has a lot to offer in the constant battle against malicious software. It is an important part of the process used by attackers to write the malware, it needs to be better utilized by the security field to combat future attacks. While reverse engineering a specific virus may only give insight into the methods of the initial programmer, we need to remember that the code and programming for these viruses are shared amongst a community of attackers. Taking the time to better understand the way one virus operates can provide valuable information in the fight against future attacks by different authors. The constant back and forth between attackers and defenders is in essence a technological war, as such, strategies used in war can be applied in some extent to the information battle that continues today and shows no sign of ending. Reverse engineering malware can be summarized effectively by a single statement in "The Art of War" by SunTzu; "If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle." (Tzu)

References

Dedicated to providing information about computer viruses to anyone who is interested in this topic. (n.d.). Retrieved August 16, 2007, from VX Heavens Web site: http://vx.netlux.org

OllyDbg debugger. (n.d.). Retrieved August 15, 2007, from OllyDbg v1.10 Web site: http://www.ollydbg.de

The IDA Pro Disassembler. (n.d.). Retrieved August 15, 2007, from DataRescue Web site: http://www.datarescue.com/idabase

Lyon, G. (n.d.). *Nmap Security Scanner*. Retrieved August 15, 2007, from Insecure.org LLC Web site: http://www.insecure.org

Mohandas, R. *Hacking the Malware - A Reverse Engineer's Analysis*. Retrieved August 15, 2007, from http://rahulmohandas.blogspot.com

Nessus Vulnerability Scanner. (n.d.). Retrieved August 16, 2007, from Tenable Network Security Web site: http://www.nessus.org

Rozinov, K. (2004). *Reverse Code Engineering: An In-Depth Analysis of the Bagle Virus* (1.0st ed., p. 3). Government Communication Labratory - Internet Research. Retrieved from http://rozinov.sfs.poly.edu/papers/bagle_analysis_v.1.0.pdf

Tzu, S. III. Attack by Stratagem (Lionel Giles, Trans.). In *Sun Tzu on The Art of War*. Retrieved from http://www.chinapage.com/sunzi-e.html

Zeltser, L. *Reverse-Engineering Malware*. Retrieved August 16, 2007, from http://www.zeltser.com/reverse-malware-paper