Summer Research Proposal

Windows Rootkit Detection and Removal Algorithm

Christopher Wood

Faculty Mentor: Professor R. K. Raj

Department of Computer Science, Rochester Institute of Technology

February 6, 2009

Abstract

In the world of computer security, a rootkit is one of the most feared forms of malicious software that threaten computer safety, security, and integrity. Unlike viruses, Trojans, and worms, rootkits do not cause obvious damage or harm to a computer system. Instead, they hide other malicious software and extend access as a system administrator to unauthorized users that allow them to control the system. The main problem with rootkits is that they are extremely difficult to detect. This is due to the fact that they hide files, network connections, processes, and even system memory from programs that are used by the rightful system administrator to detect things that have access to such system resources. In essence, once they work their way into the system, they disguise themselves.

Along with granting the unauthorized user access to the system as an administrator, rootkits may also serve to open up "backdoors" for the user so that they may have access at anytime. In this research project, my goal is to learn more about how rootkits are created and how they operate through readings and research, develop algorithms and techniques to figure out how to find rootkits in a Microsoft Windows environment, determine their access into the system, and develop additional algorithms and techniques to remove rootkits from the system.

Background and Significance:

Rookits exists in many different forms. The most common rootkits were found to be in the hardware/firmware, kernel, library, application, and hypervisor levels (Butler, Hoglund). The hardware/firmware rootkits use the system firmware, which are programs that internally control the system and its hardware, to create a malicious software program. This form of rootkit is difficult to find because system firmware code is not checked to be safe or uncompromised on a regular basis. Rootkits found in the kernel level (the central component of most systems) are especially dangerous. If rootkits can find their way into this level and successful modify the kernel objects of the system, then the entire system falls under its control (Baliga, Iftode, Chen). Library level rootkits modify legitimate programs in the system by patching (modifying the program), hooking (creating a new event to be executed by the program), or replacing system calls with ones that will hide the rootkit from the system. Application level rootkits simply take a legitimate program and modify it so that it doesn't perform as it should. Most of these rootkits employ techniques such as code injection, or modifying a program's existing code to make it act differently, to stay hidden. Rootkits found in the hypervisor level are the most difficult to find of all. They operate by disguising itself as the host operating system when the machine boots up which treats the actual operating system as a guest. Therefore, the rootkit has higher privilege than any program the system administrator runs and is almost undetectable (Radcliff). Since this is the most difficult and hardest to implement rootkit, I do not plan on attempting to find a solution to them in my project.

Project Design:

This project will be divided into four major parts. The first of which will involve researching rootkit history, invasion and evasion techniques, and design in order to get a better understanding of how they operate. The second part involves finding discrepancies in the actual Windows API (Microsoft's core set of application programming interfaces) that differ from a raw collection of data from the system. This information will be used in the third part by scanning file code where discrepancies were found to determine if it is of malicious content. If the files are deemed to be rootkits, then I will determine at what level the file is tied into the system to attempt to repair or remove it, which is the last part of the project. The first actual step will uncover any file that is being hidden from the system administrator, whether it is a rootkit or not. The second step is to determine if the hidden files found in the first step are actually rootkits. Finally, once a rootkit is found and identified, I will attempt to find out how it is being employed and counter-attack it.

Project Goals:

This project will require a certain amount of knowledge of the Windows environment and C programming language. Therefore, I plan on conducting my own personal studies of both before the summer as preparation. I also plan to conduct a literature survey which will include an in-depth study of the book, *Rootkits: Subverting the Windows Kernel,* and the website, http://www.rootkit.com/, as part of my ongoing research. As I stated above, I plan on splitting the project into four different parts. It is my goal to research and implement the search and difference comparator algorithm in the first two parts of the project using the C programming language. I do not plan on using a GUI (Graphical User Interface) for this part as I am only interested in the results, not so much as ease-of-use. A GUI can be implemented

into the program anytime after.

The third part of the project will build upon the algorithm used in the first part by using the files returned by the program and scanning them for malicious content. This task will take longer than the first part since it deals with comparing the source code of the returned programs/files with the code of non-malicious programs/files. This portion of the project will require more professor support than the first three. If all goes well, I want to be able to begin the fourth task. It is not my intention to finish it completely because it is the most challenging, but to at least design a method to determine the rootkit's tie in with the system so that I may implement it at a later time.

Finally, I plan to work with my mentor to find an appropriate regional or national conference to publish my experiences and results of this summer project, either as an experiences paper or a poster paper. This will allow me to build up additional skills in presenting my work to an audience external to R.I.T.

References:

Baliga, Araty, Liviu Iftode, and Xiaoxin Chen. "Automated Containment of Rootkits Attacks". Department of Computer Science, Rutgers University. July 10, 2008.

Butler, James and Greg Hoglund. *Rootkits: Subverting the Windows Kernel*. Addison-Wesley, 2005.

Radcliff, Deb. "How To Root Out Rootkits." *Network World*. Southborough: Aug. 11, 2008. Vol. 25, Iss. 31; pg.28.