Remote Intrusion Detection

Mario Sutton
Fayetteville State University
1200 Murchison Road
Fayetteville, NC 28303
msutton2@broncos.uncfsu.edu

Stacy Prowell, Ph.D.
Oak Ridge National Laboratory
1 Bethel Valley Road
Oak Ridge, TN 37831
prowellsj@ornl.gov

Abstract

Maintaining a safe cyber infrastructure is one of the many objectives the U.S. is trying to accomplish for its people, but with hackers developing more ways to intrude into places they are unwanted, this task of maintaining a safe cyber infrastructure becomes more difficult. As critical infrastructure is increasingly targeted, better ways are needed to either keep the bad guys out of the network or to detect them before they can do any major damage. While trying to keep hackers out of the cyber infrastructure altogether is challenging, the idea of detecting them before they damage or steal important information can be explored.

The Beholder project is a DOE-funded research project with collaboration from GE Research on remote intrusion detection for SCADA and critical infrastructure systems. Information such as average time and energy use of software on safe and secured computers is collected and used as a baseline. Next, ways of how that information can help us detect intrusions are investigated. Operations used by hackers require more time and energy from the computer, thus increasing its workload. SCADA will be addressed first because it typically has a more stable configuration, a consistent workload, and can also relay information from many locations to a single location for inspection. Through comparisons in baseline and compromised workloads in an infrastructure, observations can be made to best determine when an intrusion may be occurring and actions can be taken to stop hackers in their tracks.

Introduction

Protection of critical cyber infrastructure is an important task to uphold. This task becomes even more important each day as the number of hackers increases and improve their methods of obtaining access into undesired territory. As their methods improve, so must the

DRAFT

methods used to protect the critical cyber infrastructure that is relied on by millions of people in the United States each and every day. This is where Remote Intrusion Detection can be an advantage. Remote Intrusion Detection arises from the idea that things such as malware, spyware, and rootkits all create differences from the normal when an infected machine is in operation versus a clean machine. If these differences are studied, then detection of these problems can be identified before any type of damage is done to the machine or data is stolen and/or destroyed permanently. Early detection leads to less problems and a possibility of a safer and more secure cyber infrastructure. With cooperation from GE Research, the idea of Remote Intrusion Detection becomes even more of a reality with the use of their D400 Substation Gateway along with Supervisory Control and Data Acquisition (SCADA) systems as well.

Background

Intrusion Detection is defined as the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. The main focus of intrusion detection systems (IDS) are identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators. There are four main types of intrusion detection technologies, network- based, host-based, wireless, and network behavioral analysis. Network-based IDS keeps tabs on network traffic and analyses traffic for any suspicious activities. Host-based IDS keeps tabs on the user's daily activities and monitors for suspicious activities. Wireless IDS focuses on wireless traffic and monitors for suspicious activities. Network behavioral analysis examines network traffic to identify threats that generate unusual traffic flows [1].

Most attacks made to cyber infrastructures are through the use of malicious software, or malware. Malware is defined as a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or otherwise annoying and disrupting the victim [1]. Malware is widely available on the internet and includes viruses, worms, spy-ware and ad-ware [4]. Most attackers want to infiltrate the infrastructure with little changes, and the most

common type of malware used for this task is a rootkit.

Rootkits are “kits” of software used to gain and maintain root / administrator privileges on an unauthorized machine. Rootkits are known greatly for their stealth, and attackers use them for that very purpose. Many rootkits exist, but they can be categorized into five types, kernel-mode, user-mode, boot-loader type, hardware-type, and virtual-based or hypervisor-based. Kernel-mode rootkits dive deep into the kernel of the operating system. In this position, it cannot be detected by most anti-virus and anti-spy ware software and can operate at root level, allowing it to make the necessary changes to stay hidden, such as log modifications and system call redirections, thus removing trust from the operating system, and providing backdoor access for future use by the attacker. User-mode rootkits exist on the outer most layer of the operating system. There they can alter security and hide processes, files, system drivers, ports, system services and modify application files and their behavior. Boot-loader type rootkits replaces or modifies the legitimate boot-loader to where the rootkit loads before the actual legitimate loader. Hardware-type rootkits can hide in hardware, like network cards, during shutdown and re-install itself when the machine is restarted. Hypervisor-based rootkits exploit hardware features such as Intel VT or AMD-V virtualization technologies. Hypervisor level rootkits hosts the target operating system as a virtual machine and therefore they can intercept all hardware calls made by the target operating system [2] [6].

IDS are a keepsake in the fight to secure the cyber infrastructure although there are some problems with modern day IDSs. First is that use of just one of the technologies limits the amount of intrusion detection that can be done. A solution to this problem is just using more than one IDS technology, but that opens another issue. The second issue is that implementation of two or more IDS technologies simultaneously can be rather difficult. Third, and most important issue of all, is that all of these IDS technologies only detect intrusions after the infrastructure has been compromised. This is not good due to the fact that many intrusions only need a short period of time to create something such as a backdoor that permits unlimited access into the infrastructure. Also, if there is an intrusion, although notification to the administrator can be made, if they are not closely around at the very moment the intrusion happens, then the infrastructure is still in great danger of having important information stolen or destroyed. This is what makes the idea of remote intrusion

DRAFT

detection special.

If information is gathered on how different types of malware affect computer performance, detection of intrusions before system compromise becomes very possible. Remote Intrusion Detection is the idea of combining the known methods of intrusion detection and making it more automated and making earlier detection possible as well. This combination improves how intrusions can be detected and the fact that if it is automated in some ways, that intrusions can possibly be detected more accurately and more often. This also increases the range at which intrusion detection can be implemented, establishing it where there are very few people and reducing the amount of people needed to operate and maintain the intrusion detection systems in these remote locations, thus increasing the strength of the cyber infrastructure and also increasing its efficiency as well.

Supervisory Control and Data Acquisition (SCADA) systems are used all over the United States as a means of monitoring and maintaining control over important procedures and resources such as “electrical power grids, water distribution, wastewater collection and management systems, oil and gas pipelines and railway transportation systems [5].” The main function of a SCADA system is to collect information and relay that information to a Master Station where it is analyzed and decisions are made by the Master Station itself or a human on whether any modifications or adjustments need to be made. Another important aspect of SCADA systems is that they can be placed in remote locations where people are scarce, making it easier to monitor and manage whatever may be in those locations.

The D400 Substation Gateway is a device from GE that is a “secure, substation-hardened gateway that collects metering, status, event, and fault report data from serial or LAN based intelligent substation devices. The D400 summarizes data from the substation devices and makes it available locally/remotely through a standard secure web browser like hypertext transfer protocol through secure shell (HTTPS).” The many benefits that it has advanced security features, full SCADA protocols, industry standard communication support and secure pass through/terminal services among others. It also has important applications such as advanced gateway and automation, fault recording and data logging and secure remote access [3].

DRAFT

Resources

The D400 Fedora Core 11 Virtual Machine is a software element that runs the Fedora operating system by Linux. It also runs the D400 build, which is a virtual method of exploring and learning how the D400 works and how it can modified to meet the user's specific needs. Also needed is the Virtual Box Virtual Machine form Oracle, which is software that runs a virtual environment or guest operating system within a host operating system, such as a local machine. This virtual environment is what is used to run the D400 build and all of its operations. An Intel Pentium 4 HT processor, located inside the local machine, is also used as the host operating system.

Process and Implementation

A literature review was first needed because this was all new material. Time was spent reading many pdf's and webpages on intrusion detection, rootkits, and the D400 to grasp a good understanding of what needed to be accomplished during this research. After the literature review, first thing done was to getting Virtual box up and running with the Fedora Core 11 Virtual Machine on it. This involved downloading Virtual box onto the host machine and setting up the Fedora guest operating system on the host machine's operating system. Once that happened, the Fedora workstation was setup as needed. Next, several rootkits were analyzed for important data such as how they infiltrate infrastructures, which files that they implement to gain and maintain access on unauthorized machines, and what they do to stay hidden on unauthorized machines. There were many rootkits studied such as sucKIT, Knark, Adore-NG, Phalanx2, which are kernel-level rootkits, LRK5, and T0rn, which are user-level rootkits [7]. Also, they system calls that these and other rootkits modified were also studied to find which system calls are targeted most by them and use that knowledge towards this research. This data was used to learn which files needed to be focused on for the implementation of an intrusion detection system for the D400. Also, coding is done, creating a block of code that gathers the execution time of any program that it is required. This code will be used to establish a baseline for the execution time for programs on a clean machine. This data will be compared to the execution time of programs on a compromised machine and conclusions will be made based on the comparisons of clean and compromised machine execution time programs.

DRAFT

Results

The collection code needed for this research was created and it was observed collecting time execution data. Also, Linux rootkits relevant to the D400 device were also discovered along with the system calls that are targeted most by rootkits as well.

Conclusion

This research was aimed at finding a beginning to an idea of Remote Intrusion Detection. A lot of information was sifted through to find the important stuff related to this research. From this research, we were able to successfully conduct initial timing tests for a clean machine. From the timing tests, baseline execution time for selected system calls can be established for comparisons between clean and compromised machines in future work on this research.

Future Work

Future work includes actual real time implementation. Most of this research was conducted virtually using the Fedora workstation and the Virtual box virtual machine on the local host machine. Virtual and real time implementation with the actual D400 device will allow more accurate research of how it actually works and also allows the creation of solutions for any other issues that may arise with implementation. Also, compromised machine tests need to be conducted as well for future comparisons.

Acknowledgements

The Research Alliance in Math and Science program is sponsored by the Office of Advanced Scientific Computing Research, U.S. Department of Energy. The work was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725. This work has been authored by a contractor of the U.S. Government, accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. The author would like to thank Dr. Stacy Prowell for the opportunity to work on this project. The author would also like to thank Aaron Ferber for the tremendous amount of help he provided during this research. Additional acknowledgements go to Tim Schulz, Evan Frenn, and Jarilyn Hernandez for their help as well. Finally, special thanks are

DRAFT

given to Debbie McCoy, who made provisions for this research experience and made this opportunity possible.

References

[1] Scarfone, K. , Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems.* National Institute of Standards and Technology.

[2] Rootkits. *OmniSecu.com.* Retrieved June 2012 from http://www.omnisecu.com/security/rootkits.htm

[3] General Energy. n.d. Multilin D400. *GEDigitalEnergy.* Retrieved June 2012 from http://www.gedigitalenergy.com/multilin/energy/catalog/D400.htm#new

[4] Wikipedia. n.d. Malware. *Wikipedia The Free Encyclopedia.* Retrieved July 2012 from http://en.wikipedia.org/wiki/Malware

[5] Stouffer, K.. , Falco, J. , Kent, K. (2006). *Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security.* National Institute of Standards and Technology.

[6] Kassner, M. Sep. 17, 2008. 10+ things you should know about rootkits. *TechRepublic.* Retrieved June 2012 from www.techrepublic.com/blog/10things/10-things-you-should-know-about-rootkits/416

[7] Baliga, A., Iftode, L., Chen, X. n.d. *Automated Containment of Rootkits Attacks.* Retrieved July 2012 from www.cs.rutgers.edu/~iftode/paladin08.pdf

DRAFT