Abstract

Computer networks are a system of interconnected computers for the purpose of sharing digital information. The concept of a network began in 1962 when a server at the Massachusetts Institute of Technology was connected to a server in Santa Monica, California. Since that time the proliferation of computers and computer networks has increased significantly. One of the most significant challenges to networks is attacks on their resources caused by inadequate network security. The purpose of this research project was to evaluate open source, free, intrusion detection systems and how easily they can integrate into an existing network. Research was conducted for this study through a review of existing literature pertaining to intrusion detection systems and how they function. The literature also highlighted previous studies conducted on intrusion detection systems, both commercial and open source. In addition to the review of existing literature, the author conducted independent testing on three open source intrusion detection systems. The open source programs, Snort, OSSEC, and Prelude, were selected due to being highly rated in professional publications. The author created a secure simulated computer network, to ensure that each of the programs was tested in a controlled and equitable manner. The findings of this study determined that the three open source intrusion detection systems tested are as capable as commercial programs in securing a computer network.

NETWORK SECURITY THROUGH OPEN SOURCE INTRUSION DETECTION SYSTEMS

By

Russell Anthony Tantillo

A Research Project Submitted to the Faculty of

Utica College

May 2012

In Partial Fulfillment of the Requirements for the Degree

Master of Science

Table of Contents

List of Illustrative Materials

Acknowledgements

This Capstone project would not have been possible if not for my professor and mentor Paul Pantani and content expert Sean Terrill, whose constant encouragement, dedication, and commitment enabled me to successfully complete this amazing accomplishment.

It is a great pleasure to thank everybody who helped by showing me unwavering support throughout the process. I am truly indebted and thankful to my friend Brian Frank for all of his guidance and patience in helping me solidify my ideas and giving me direction. I would also like to thank all of my close friends for their support as well as my parents and brothers for their patience and understanding while I perused my goal. I also owe a very special thank you to Beth for her patience in dealing with the long hours and endless days it took to complete such a significant achievement. Lastly I would like to thank Utica College its faculty and staff for allowing me the opportunity and means to reach the pinnacle of my academic carrier. Finally I would like to thank all of my classmates for whom sharing this experience with was an honor, and truly one of the greatest chapters of my life.

Network Security Through Intrusion Detection Systems

Networks are complicated interconnections of computer systems and peripherals designed to achieve the goal of sharing digital information. The concept of a network started in 1962 at the Massachusetts Institute of Technology (MIT) and culminated in the first wide-area-network (WAN) connection in 1965 when Larry Roberts and Thomas Marill connected the TX-2 at MIT to the Q-32 in Santa Monica via a dedicated telephone line with acoustic couplers (Computer History Museum, 2006). A computer network consists of two or more computers that are linked together in order to share resources such as printers, files, or electronic communications. These computers may be linked through a network cable, router, switch or telephone line (Winkelman, 2011).

Like all interconnected systems, computer networks are vulnerable to attack. Weak points in a network can be, "…exploited, either by designed action of an adversary or by accident, to affect the operational capability or trustworthiness of that network" (Alward, Carley, Madsen, Taylor & Vandenberghe, 2006, p. 3). Once a weakness is recognized it can be exploited either through a series of scripts or a dedicated program to gain entry to the network. Network attacks include viruses, worms, Trojan horses, spyware, adware, scareware, hackers, Denial of Service (DoS) attacks, data interception and zero day attacks (Whitman & Mattord, 2009).

Due to the pervasive and destructive nature of network attacks, implementing and maintaining tools to protect a network should be a primary concern of any business (Phatak, 2011). Security is achieved by preventing and monitoring all unauthorized access to the network and by taking the necessary steps to try and mitigate the various security risks. One component of a network security protocol is an intrusion detection

system (IDS). An IDS can either be network-based or host-based. Network based systems are installed on a network and search for attack signatures which are reported back to a central command program. A host-based system is installed on an individual computer or server and monitors the operating system (OS) and files for signs of intrusion (Whitman & Mattord, 2009).

The purpose of this study was to evaluate and compare three open source, free, IDS programs. The evaluated programs were Snort, OSSEC, and Prelude. The evaluation and testing intended to answer the following questions: How difficult is each program to set up and configure? How difficult is the signature updating process and availability? What were the detection rates for each system under normal conditions using basic vulnerability scanning tools?

Unlike network design, which has a well developed process and significant advantages because it offers modularity, flexibility, and standardization of protocols, "secure network design in not a well developed process. There isn't a methodology to manage the complexity of security requirements" (Daya, 2008). To develop a secure network there are many areas to be considered. End users need to be able to access the information, private information needs to remain private, each user needs to be authenticated on the network through provided credentials, and it also needs to maintain message integrity and non-repudiation, meaning that users cannot refute that they used the network (Daya, 2008).

To try and mitigate these issues there are different types of security tools such as firewalls and IDS systems, as well as encryption and authentication mechanisms. Daya (2008) also states that, "intrusion detection systems are established based on the types of

attacks most commonly used" (para. 13). The most common attacks can be broken down into categories: attacks that are to gain system knowledge; attacks that are to gain personal information, such as eavesdropping and phishing; attacks that can interfere with the OSs functions, such as viruses, worms and Trojan horses; and lastly, attacks that are designed to consume system or network resources uselessly, like DOS attacks, land attacks, smurf attacks and teardrop attacks (Daya, 2008).

Government compliance is also an aspect of network security to be addressed. There is a need for IT directors and security administrators to understand what the compliance requirements are for businesses, and what kinds of resources would be needed for compliance. The Federal Government enacted laws such as; the Federal Information Security Management Act (FISMA), the Gramm-Leach-Bliley Act (GLBA), the Health Insurance Portability and Accountability Act (HIPAA), the Sarbanes-Oxly Act (SOX), and Payment Card Industry Data Security Standard (PCI DSS) in order to make businesses and federal agencies accountable for how information should be stored and made available to an investigation in the event of an incident.

FISMA addresses the need for federal agencies to develop, document, and implement organization wide programs to provide information security and operations support. GLBA ensures confidentiality and integrity of personal financial information stored by financial institutions. HIPAA ensures confidentiality, integrity, and availability of health care information. SOX applies to privacy and integrity of financial data in publicly traded corporations. PCI DSS ensures confidentiality of credit card information stored and used by merchants.

These acts specifically deal with a company's requirements for maintaining and collecting event log files, as well as the duration of storage. An event log is a record of events occurring on the computer systems within an organization's network. These log files have copious amounts of detail and information available within them, some of which are human readable. All of the computer systems and security devices that reside on a network, firewalls, IDSs, routers, antivirus software, and authentication servers all produce log files. Some systems keep more detailed information than others, but all of the information contained in the event log files can be useful in a network security investigation. Each of the government acts references a specific business segment and is subject to the laws and penalties contained in each individual act (Kent & Souppaya, 2006).

There are a few resources the National Institute of Standards and Technology (NIST) indicates that could potentially help mitigate the multitude of issues that face security and network personal when dealing with these log files. Security information and event management (SIEM) software is centralized logging software that can aggregate the log files from the systems files, as well as the security and network devices, into a central display. Host based IDSs and visualization tools are also suggested as viable options for collecting and displaying the log file data (Kent & Souppaya, 2006).

IT directors and system administrators evaluating open source IDS programs are doing so because they are trying to implement an effective security solution while working with reduced budgets. Open source software has several advantages when compared to closed source or commercial products, which are both fee based, "Open source has been proven to provide better value, lower costs, and improved security,

addressing the most important enterprise considerations today" (Portelli, 2010, Para. 6). Open source software can provide business and facilities the ability to provide same as or as close to commercial protection, for little or no cost (Cox & Gerg 2004). According to Cox and Greg (2004), "All of the commercial IDS solutions on the market grew out of open source beginnings. The main difference between the open source and commercial tools is the ease of installation and the fit and finish" (p. 208).

There is a lack of understanding about IDS programs by network administrators, "… describes a product category that's not very new, though many IT managers don't seem to know what it is or its capabilities" (Phatak, 2011, para.1). Their reliability, stability, audit ability, return of investment (ROI), flexibility, and community support are all positive aspects of using open source software, but yet, to date there is a lack of research or data that provides a thorough analysis of open source IDSs. (Whitman & Mattord, 2009). According to Lennon (2009), "Despite the expansion of IDS technology in recent years, the accuracy, performance, and effectiveness of these systems is largely untested, due to the lack of comprehensive and scientifically rigorous testing methodology" (para. 1).

IBM Zurich developed prototype IDS testing platforms by automating attacks and generating background traffic in a way to improve the IDS. The purpose and attempt was to try to develop a completely automatic evaluation platform. MIT - Lincoln Laboratory (MIT/LL) performed the most extensive quantitative tests to date, by developing a corpus of data sets that included background traffic as well as attacks. The testing continued for several weeks in order to gain a better understanding on how IDS programs work and how they can be improved. The Air Force Research Laboratory (AFRL) did testing on

IDS systems in real-time and with a more complicated network setup. The IDS programs were set up and four hours of background traffic, as well as attacks were launched at the host computers. These tests were conducted on commercial as well as government off the shelf IDSs.

MITRE Corporation investigated the characteristics and capabilities of network based IDSs. The testing included seven commercial and government developed systems. These tests examined the capabilities of real time alerting, reporting, and off line analysis. The Neophapsis Laboratories/Network Computer Magazine collaboration evaluated 13 commercial systems and one open source IDS, the tests conducted examined the performance of the systems under realistic traffic loads. Also the researchers examined how high traffic loads can cause dropped packets and undetected attacks. The NSS Group also evaluated 15 commercial IDS systems and one open source IDS. The evaluations consisted of ease of installation and configuration, architecture, as well as types of reporting and analysis provided. Basic attack methods, as well as evasion techniques to circumvent systems, were both used to try to trigger alarm. The tests also included attacks specific to eliciting false alarms. World Fusion Magazine conducted a limited evaluation of five commercial systems on their ease of setup and features, as well as detection accuracy. The tests used common attacks including stealth attacks with no background traffic, but did use artificially generated network loads (Lennon, 2009).

The two evaluations which were conducted by Neophapsis Laboratories and The NSS Group included the only open source IDS system, Snort, in their testing. In the report from The NSS Group, Snort scored very well in all areas tested. It was indicated that it performed very well in the “real world” tests and would perform well on any

typical corporate network (The NSS Group, 2001, p 135). What was not indicated by either The NSS Group's test or the test conducted by The Neophapsis Laboratories is why they specifically included Snort as an open source IDS that was evaluated, but did not include any other open source programs (The NSS Group, 2001). It is the lack of standardized testing, lack of peer software evaluations, and lack of unified methods for testing, which leave a void in evaluating open source IDS programs (Lennon, 2009).

The evaluation of the three IDS programs consisted of an extensive review of existing literature. Moreover, original research through testing of each of the IDS programs was conducted in a dedicated and controlled environment. Based on the research findings, this study will propose suggestions that business owners, information technology (IT) managers, or network security administrators might use in selecting an open source IDS program. In the event the open source variants presented here should fail to perform adequately, I will make recommendations indicating which corporate and enterprise systems would provide adequate solutions.

The following section will consist of the literature review for this project. The author will present information as it pertains to IDSs and how they function. The presented information will consist of a variety of reference sources.

Literature Review

Computer networks are comprised of a series of interconnected computers, and were designed for the purpose of sharing information. Connected networks, especially those connected to the Internet, are vulnerable to unauthorized intrusions. Network intrusions can be in the form of viruses, worms, spyware, and hackers. Securing a network includes monitoring and preventing all unauthorized access to the network. The

purpose of this project was to evaluate open source IDSs and how they can provide security in a network environment.

The research for this project consisted of literature sources including, scholarly papers, news articles, published books, and government publications. The research literature was chosen to provide a clear understanding of all the different aspects of IDSs from their inception to the current state of hybrid systems, and beyond. Additionally, independent testing of open source IDS programs was conducted by the author. The testing conducted was to augment the literature material in evaluating each of the systems on a first hand basis to gain further knowledge of how they operate.

Computer Networks

A Local Area Network (LAN) and Wide Area Network (WAN) are two types of computer networks. A LAN is a network that is confined to a relatively small area, generally limited to a geographic area such as an office building, school or home. A WAN connects smaller LAN networks together, such as a company's headquarters in New York City to a remote office in Florida. A network facilitates electronic communication, file sharing or calibration, such as a video conference (Winkelman, 2011).

Network Vulnerabilities

The average time a new system, which is un-patched and un-defended, can be connected to the Internet before it is attacked is approximately twenty to thirty minutes, yet more than thirty percent of those polled by the National Cyber Security Alliance (NCSA) think they will be struck by a bolt of lightning before they see their computers violated in an Internet attack. In 2004, the computer worm *My Doom* was released.

According to the Internet Security Alliance Data, 1 in 3 small businesses were affected. In comparison to that data 1 in 6 larger enterprise computer systems was affected. Network vulnerabilities exist for all companies, but specifically for smaller business that may not have the technical personnel or resources to put the correct security measures in place, is one facet of the issue.

Small business owners simply dismiss the idea that their systems are vulnerable simply because they believe they are too small for hackers to bother with. The flaw in that notion is that the computer systems that are connected to the Internet are all vulnerable, regardless of location or how they are being used. By the end of 2005 almost half of all small business that utilized the Internet for browsing and email capabilities were successfully attacked and most likely never have knowledge that it happened (Nijnik, 2007).

Because of this interconnectivity between machines, network attacks are important to defend and protect against. When one system on a network is compromised there is a good probability that all the other computers inside that network will also be compromised. Network attacks can be done manually or automatically. In a manual attack, a Trojan horse virus can be introduced that gives the attacker remote control of a single system, then from that system the attacker can systematically compromise other systems. An automatic attack can accomplish the same compromises through the use of a worm virus. Worms do not need to have operator interaction to spread to other computer systems. They have self reproductive and distributive properties built into them. After they infect a system, that system can then automatically infect every other system connected on that same network (Whitman & Mattord, 2009).

Rantala (2008) states, for businesses, "computer viruses accounted for 193,000 hours and other computer security incidents resulted in more than 100,000 hours of system downtime" (Para. 2). External threats only account for a portion of the data loss incurred by a company; the most significant threat comes from within the company. Exfiltrating data is actually a prevalent network security problem. In 2005, The Bureau of Justice Statistics created a report on cybercrime against business and found that seventy-four percent of cyber theft was perpetrated by someone inside the company. There is a need for improvement within the corporate structure of network security.

Kozushko (2003) states that the frequency of attacks is increasing as well as the amount of vulnerabilities networks are exposed to. Because of this, network security personal need to extend the protections created by firewalls by complimenting them with IDSs. Kozushko echoes the statements from other authors, reiterating that not all threats start from outside the firewall, "A vast majority of loss due to security breaches is traced to inside the company" (Page 4). Attacks on the firewalls themselves as well as deliberate evasive actions were also indicated as reasons for the need for additional security measures. These additional measures are not only just for security, but they are also for accountability and documentation. IDSs can provide forensic evidence of not only the attack itself, but also could help identify the location of and provide the needed data to apprehend and possibly prosecute the perpetrator.

Intrusion Detection System (IDS)

Anderson (1980) provides information as to how and why IDSs work. In this report, Anderson explains how important security audit trails are, as a component of computer security, for detecting unauthorized access to files. This is especially necessary

for machines that do not have built in mechanisms for detecting unauthorized activity. The report also explains that not only is there a need for detecting and logging unauthorized activity, but also authorized and excessive activity.

The ideas and procedures documented in this report were the basis for current IDSs. This report was based on an analysis of a company's infrastructure to try and understand if it was possible to determine the different kinds of activities that would be detrimental to that company's stored data. The report presented a plan of action along with a discussion of what should be done to insure that the exposure to the potential threats addressed by Anderson could be effectively added to a company's information security practices.

Lydon (2004), states that in the mid to late 1990's there was a significant increase in interest regarding IDSs, and the value they might bring to network security. There were a significant number of research papers and articles written on the subject. Though there were numerous research studies conducted, there was a distinct differentiation in the research methodologies of each test. Additionally, none of the studies continued where a previous project ended.

In 1998 and 1999, MIT/LL conducted a series of tests for the Defense Advanced Research Projects Agency (DARPA). These tests identified how IDSs function and addressed how they could be improved. After the MIT/LL tests, software developers now had a solid foundation to build upon. Some industry experts still consider the MIT/LL tests to be the most comprehensive set of tests to date.

Whitman and Mattord (2009), state that there are two categories of IDSs, network-based and host-based. Network based systems reside on the network

infrastructure and search for attack signatures. These types of systems require deploying probes or sensors in different locations on the network that report activity back to a central command program. A host-based system is one that gets installed on an individual computer or server and monitors the OS and files for signs of intrusion. To function properly and monitor all activity, host-based IDSs have to be installed on every computer or server that need to be monitored.

Network based systems can monitor an entire network with a few sensors in key positions. These passive systems do not require or use a significant amount of bandwidth or network resources. Conversely, on very large or extremely busy networks, network based systems might miss analyzing some of the packets. Moreover, they cannot analyze encrypted data, and can require heavy user involvement from administrators.

Host based systems can scrutinize and analyze with great detail, the activity of the system they are installed on. Host based systems can monitor encrypted traffic, and can report to a remote centralized console. Host based systems can place large demands on system resources, which can decrease network performance. Additionally, they can be easily disabled by a clever attacker.

Along with the two main types of IDSs, there are also two different methods to how each program conducts event analysis. These methods are signature-based, and anomaly-based. A signature based system uses a system similar to antivirus programs when detecting and blocking infected files or programs. Signature based analysis maintains a database of signatures that can be updated each time a new attack is discovered. Some products can do this automatically, while others require manual updates. This need for consistent database updates can cause performance issues on

systems when partial signature matches occur. Signature based is the most common type of IDS.

Anomaly based is a rule based system. Rules identify what is normal or abnormal, also referred to as heuristics, activity on the system, and then block suspicious activity according to those rules. Anomaly based analysis sets a baseline of what is considered normal behavior and new attacks can be quickly detected. Anomaly based analysis is prone to false positives. Additionally, this method can cause significant system lag when performing large amounts of analysis. A significant drawback to anomaly based analysis is that they require a learning period to create the normal baseline. During this learning period the host system is susceptible to attack.

Denning (1982) pioneered the idea about being able to detect threats from within the network. The whole premise of her study was to call attention to the need for internal controls and monitoring of unauthorized activity through the use of accounting software. There is a bit of nuance in understating the differences in the above statement. Unauthorized activity might suggest that there is a virus or malware on the system. Moreover, unauthorized activity could also represent an unauthorized entity that might have accessed the network. Authorized, excessive activity would be suggestive of a rouge employee that is allowed to access certain files or system resources, but is doing it in a manner that would be considered in potential violation of security policies. Denning also had references to this type of audit trail monitoring for network security. The majority of the author's information is system and program orientated. Denning also states that controls are needed to prevent data leaking; more specifically, not allowing users with lower security clearances to access classified data.

Sans (2001) states that there are studies which refer to IDSs as "burglar alarms," basically they are used to augment the other components of a network security program as a whole (para 5). IDSs are not designed, nor are they marketed as plug them in and forget about them systems. Moreover, that they alone are the perfect answer for a secure network. The research does indicate that a network can be significantly more secure with and IDS in place, especially when augmented by the other components of network security such as antivirus software and firewalls. "The firewall protects an organization from malicious attacks from the Internet and the intrusion detection system detects if someone tries to break in through the firewall" (para.6).

Ho (2001) suggests that the main difference between an IDS and a firewall is that a firewall lacks the capabilities to monitor the network traffic where system attacks would take place. The author also acknowledges that the source of the attacks can be disgruntled employees that intend to utilize their legitimate network privileges to do harm. Because firewalls are only effective at preventing unauthorized access at the point of entry, if an attacker was to circumvent the firewall they would have the ability to move freely throughout the entire network. "To keep a constant eye on network traffic and to know anything out of the ordinary is happening, network security should be supplemented with Intrusion Detection Systems (IDS)" (Page 2).

Eanes (2003) states that, "Some of today's companies are still in denial phase…still believe that just by having a firewall up and running they have more than adequate security protection for the network" (Para 2). This is not the case, especially for network traffic that has permitted to be on the network. If all the other security devices

either failed or unknowingly let malicious data onto the network the only system that could detect it and even stop it is an IDS.

Bace and Mell (2001) present information that IDSs are necessary to protect organizations from the threats that come with increasing reliance on network infrastructures as well as the increasing number of criminal intruders."The question for security professionals should not be whether to use intrusion detection, but which intrusion detection features and capabilities to use" (para 3). Understanding the limitations of what the software can and cannot do is a key aspect of justifying the acquisition of an IDS to an organization and how it will fit into the network security scheme already implemented. Most importantly, understanding how it can help mitigate risk and provided the necessary tools to comply with government regulation as well as provide documentation should the legal need arise.

Meyer (2008) stated that as with real burglar systems being triggered by false alarms, the same is true for IDSs. When deploying an IDS within a network infrastructure, there is a potential for false alarms along with valid alarms, and having someone who is trained to understand the differences is an important consideration. The size of the facility and number of systems that need to be monitored can impact the volume of alerts that would need attention.

In 2008 an IDS was integrated into a large healthcare facility. Shortly after the installation, the IDS was receiving 400 alerts per minute and over 400,000 alerts per day; a significant number of the alerts being false alarms. With that amount of information to address, it quickly overwhelmed the response capabilities of the security group. The security group did not have the necessary personnel to be able follow up on all of the

alerts. The end result was a compromise; the company chose to record all alerts other then ones identified as true false alerts and implemented a filter to elevate the alerts that needed immediate attention. All other alerts were archived for further analysis at a possible later date or for forensic purposes (Meyer, 2008).

Del Carlo (2003) addresses the importance to verify false alerts to ensure that they are in fact just false positives. To add to the importance for verifying false positives, deliberate attacks to a network can appear to be false alerts. Del Carlo explains several methods of how to bypass or confuse an IDS. Network attackers commonly employ these methods to defeat an IDS. "IDSs have weak points and flaws that can be exploited by an attacker to get around the system" (para 1).

Intrusion Detection System Testing

Mell, Hu, Lippmann and Zissman (2003) present information related to the testing of IDS programs by various institutions and organizations. The authors address a report produced by the National Institute of Standards and Technology (NIST), which was sponsored by the Defense Advanced Research Projects Agency (DARPA). The report evaluated commercial programs, government developed programs and one open source program. The report highlighted existing testing of IDSs at that time. The report validated that there is a need for quantitative evaluations of IDSs.

The University of California at Davis (UCD) conducted research, which led to the development of the first automatic attack platform for testing IDSs. The testing included using the early IDS, Network Security Monitor (NSM). NSM used keyword matching to try and detect attacks within the network traffic. These tests were effective

until the IDSs were under high loads. These higher loads caused packet loss and unsuccessful attack detection.

IBM Zurich conducted similar tests, but utilized automated attacks and background traffic to try and improve IDS design. Using scripts and attacks specific for File Transfer Protocol (FTP) servers, the testing focused on detecting attacks specific to vulnerabilities that might be present on those types of servers. Another focus of the study was to automate the tasks required to perform an IDS evaluation.

MIT/LL performed quantitative IDS testing. The testing was sponsored by DARPA which included conducting large scale testing of IDSs. The first test conducted in 1998 produced a corpus of data sets that could be used for further testing of systems. The tests in 1999 differed from the previous tests and were designed primarily to measure the ability of the systems to detect new attacks without prior training for them. The data set that was produced from the testing have led to several tests and extensions of the MIT/LL corpus. Anzen Computing utilized the MIT/LL corpus to test and evaluate five commercial programs to determine if they were susceptible to IDS evasion techniques. The Lincoln Adaptable Real-Time Information Assurance Testbed (LARIAT) was produced as an extension of the MIT/LL corpus, which included two DoS attack to be used for real time IDS evaluations.

AFRL, under funding from DARPA, and who also participated in both the 1998 and 1999 testing conducted at MIT/LL, conducted tests that focused on testing an IDS in real time, but in a more complex network environment. These real time tests were conducted using commercial and government programs. During the course of the evaluations AFRL developed software specific for simulating a large network by

dynamically assigning INTERNET PROTOCOL (IP) addresses to network sessions running on the testbed.

The MITRE Corporation conducted one of the earliest evaluations of commercial and government developed IDSs. The tests conducted evaluations of seven different IDS programs in different phases of attack. The first phase utilized scanning attack tools to allow not only the testers to become more familiar with how the IDSs operate, but also allowed the attackers to practice network intrusions. The second phase was the actual attack phase, which included verifying degrees of difficulty. The tests conducted included real time alerting, reporting, off line analysis, response, and remote management.

The Neohapsis/Network computing tests were evaluations of thirteen commercial IDSs and one open source program Snort. The evaluations were performed under normal traffic loads to gather qualitative and also quantitative results. For the qualitative evaluations they examined the ease of use, stability, cost effectiveness, and depth of the signatures. For the quantitative aspect they examined the number of attacks detected, and thresholds for packet dropping. What the testing did not include in its analysis was false alarm rates.

The NSS Group conducted a test of fifteen commercial IDS programs and one open source program, Snort IDS. These tests examined the ease of installation and configuration, reporting and analysis aspects of the programs. The testing conducted was as minimal and straightforward as possible. They put the IDSs into different phases of load and also used evasive techniques to try to evade the IDS and generate false alarms.

Network World Fusion conducted a more limited review of five commercial IDS programs. They examined the ease of set up, the ease of use as well as evaluating detection accuracy. These tests also used stealth attacks in an attempt to create false positives and deceive the programs. They also tested the systems in different phases of network load to determine if the detection failure rates changed.

Open Source Intrusion Detection Systems

Snort. Snort was introduced in 1998 as a basic open source IDS. In addition, it can also be utilized as a packet sniffer or a packet logger. A packet sniffer, also referred to as promiscuous mode for a network interface card (NIC), allows software to view all of the traffic flowing into and out of the computer it is installed on, as well as all the other computers on the network. A packet logger, like a packet sniffer, can also view information on a network, but unlike a packet sniffer, makes a copy of that data and stores it for analysis at a later time.

Snort was developed to fill a niche in the lack of open source systems that are basic, easy to deploy, and cross platform compatible. Snort can be installed on several different computer OSs. Snort is powerful and flexible enough, allowing the system to be installed into any network security infrastructure as a permanent standalone solution, or an additional solution into existing network security architecture.

IDSs require routine maintenance. Maintenance includes not only the distribution of the signatures (similar to updating definitions in anti-virus software), but also the resources associated with implementing those updates as well as installing network probes or additional host based systems. Snort requires the manual retrieval and

implementation of the signatures from the developers or a third party web site, although there is no actual cost for acquiring the updated signatures.

Snort does not come with the signatures in the installation package. The signature database is how Snort checks the integrity of the file. All known viruses and malicious files have specific file signatures, which are maintained within a database by the developers of Snort. Snort users can download the most current database to ensure that the program remains up to date. Once Snort has been installed on a network, the program checks each file on the network against the signature database to identify dangerous or malicious files (Roesch, 1999).

OSSEC. OSSEC is an open source, host-based IDS which was released for public use in 2003. OSSEC performs log analysis and correlation, rootkit detection, active response, file integrity checking, and time-based alerting. Like Snort, OSSEC is also a cross-platform IDS. As a host based program OSSEC can also monitor Windows Registry files and centralized policy enforcement. It can also be used as a log analysis tool, in which it can process and monitor logs from firewalls, other IDSs, web servers, and other types of authentication logs

For OSSEC, the method for updating the system and the signatures would be to download the newest system files and install them just as the user would have to do at the initial install. Unless the user chooses to install the program on a different drive or use different drive letters, the system installer will detect that OSSEC is already installed and prompt the user that they want to update it. Selecting yes will automatically complete the update process. If the user has any custom or modified rules or configuration files, the update will not replace them (Hay, Cid & Bray, 2008).

Prelude. Prelude is a new type of hybrid IDS. Prelude is designed for Linux based OSs and will not function on Windows based OS at this time. It was released for public use in 1998. Prelude is a network intrusion detection system (NIDS), meaning that essentially it is a network sensor that sits on the network and detects anything that happens. There is a secondary module that does a file analysis to check for changes to the files to indicate if anything has changed. It is not a signature based system so if any changes are needed to be made it would be to update the system to a newer version.

There are several components to Prelude that allow it integrate with other systems to become a complete product. The modules within Prelude work together to draw in data from each of the modules, but they also gather all the logs and data available from the other security devices on the network. Prelude has a specific reporting and display module, which utilizes an Internet-based graphical user interface (GUI) to display all of the information pertinent to what is occurring on the network. This displayed information can include security flaws or data leaks, alarms for potential threats and suspicious events, indicators of a network flow issue. Prelude can also provide proof and evidence in case of a forensic investigation or governmental audit (Zaraska, 2003).

Sources Refuting the Need for Intrusion Detection Systems

Wippich (2007) explores the idea of an alternative way to conduct network security without using an IDS. In this study, the author presents the methods and theories of how to identify what could potentially be security issues for networks along with some of the triggers and issues to look for. Examining how a network functions, the protocols and how network traffic passes through the assigned ports is network security at the very basic level. This study discusses basic network security and education on how networks

operate, rather than relying on systems and programs to do the job without understanding how they perform their tasks.

Sequeira (2002) presents an alternate theory that an intrusion protection system (IPS) is superior to all of the other network security devices. “Earlier systems have served us well, but with the proliferation of sophisticated attacks and the discovery of new vulnerabilities, new methods are needed to protect precious data and network resources” (para 2). IPS systems are basically the same as an IDS except they have the ability to scan and detect, in real time, a potential threat, and then react to it by shutting down systems or other devices. Some companies actually will deploy and IDS and IPS in tandem to not only protect the data within, but also to make network security changes if a threat is perceived. This is a very different approach from how security systems rely on a program to detect what ports are being compromised and then shutting them down automatically.

Sans (2004) addresses the business model of hiring a third party company to deploy and monitor an IDS. The costs to a business to have a security company implement an IDS and manage it could reach into hundreds of thousands of dollars a year. Whereas, using a remote Managed Security Service Provider (MSSP), the cost to a company, for the same kind of coverage, could only be in the tens of thousands of dollars a year.

The purpose of the literature review was to identify the work of previous authors and then add to it, making this study much more substantive and relative. The researched literature created the foundation for what was needed to start conducting tests on the open source IDSs chosen for this study. The understanding of what was done or not done

previously, gave direction to the upcoming methodology section and helped shape not only the lab but also the tests to be performed.

Methodology

The purpose of this study was to evaluate open source IDS programs. In conjunction with a review of existing literature, independent testing by the author was conducted on three open source IDS programs. The three programs evaluated were, Snort, OSSEC, and Prelude. Each IDS system was evaluated on their consistency, predictability, ease of use, and ease of maintenance.

The testing for the project was conducted in three phases. Phase one was the building of the clean and isolated computer systems. Phase two was the testing of the IDS programs. Phase three was the recording of the test data, and conducting comparisons of each IDS program.

Phase One – Computer Configuration

The testing was conducted utilizing two computers. The first computer utilized in this study represented the network server for this project. This computer consisted of a Dell Poweredge 800, with a 2.8 GHz Pentium 4 processor, an 80 gigabyte hard drive, and four gigabytes of RAM. The computer was installed with Windows Server 2008 R2 OS, which was fully patched and updated for latest security updates; this represented a Windows based network server. Also installed on this computer was the Linux OS Ubuntu Server Edition 11.10. This represented a Linux based network server.

The second computer utilized for this study operated as the computer to attack the network. This computer was a Lenovo Ideapad Y510, with a 1.73 GHz Pentium Dual

Core processor, a 160 gigabyte hard drive, and 4 gigabytes of RAM. The OS installed on this computer was Backtrack 5 Linux.

A sanitary image was created of each of the clean installations using the open source program Clonezilla. Clonezilla performs a bit for bit copy of a computer's entire hard drive, creating an identical copy of the data spanned over several smaller files to create the entire image. Once the images were verified, the project moved to the second phase which included installing the IDS programs and beginning the testing.

Phase Two - Testing

A crossover cable was used to connect each of the computers' NICs, to simulate a target and an attack computer. The testing was designed to create as much as of a real world environment as possible, but still able to remain in control of the data streams of each computer without the possibility of outside interference. A crossover cable is a specially made computer cable; the connection pins that are set to receive are on one end, are those that are set to send are on the other end. This allows for direct connection between the two computer systems. Additionally, the computers have to be in the same workgroup network configuration. For the purpose of this test a workgroup named *IDSTEST* was created and applied to both the attack and the target computers. The IP addresses also have to be in the same range as such they were assigned 192.168.1.10 for the target computer and 192.168.1.20 for the attack computer.

The initial testing on each of the systems was for basic competency of each system and how it performed under different network conditions (network load). There were three different load phases of network traffic being generated by an automatic traffic generator in order to simulate normal Internet traffic and real world network

scenarios. The first series of test were conducted using no background traffic at all; this was to gain a baseline for how the program performed after the initial installation. Subsequent tests included a system load of 20 Mbits/sec, 40 Mbits/sec and 60 Mbits/sec.

Test Group 1. The OS Backtrack 5 was installed, which contained the open source risk assessment scanning programs. These software packages were used test the IDS programs installed on the target computer. The tests conducted covered port scans, denial of service (DoS) scans, web scans, file transfer protocol (FTP) scans, and Internet control message protocol (ICMP) scans, in conjunction with mail protocol scans POP3 and SMTP.

Port scans. The testing software simulated an external port scan. This test was conducted nineteen separate times for each IDS. Each IDS was monitored to determine if the scan was detected by the IDS. This number of tests was established based on the existing research regarding IDS program testing. In the existing data, the range of port scan tests varied between five and fifteen. Because this number of tests was arbitrary, chosen by the previous researchers with no justification as to how or why the author for this study chose nineteen to more than adequately test each of the programs.

DoS scans. Testing software simulated a DoS attack. This test was conducted thirteen separate times for each IDS. Each IDS was monitored to determine if the scan was detected by the IDS. The number of tests was an arbitrary number chosen based on the previous research to more than adequately test each of the programs.

Web scans. Testing software simulated a Web scan. This test was conducted ten separate times for each IDS. Each IDS was monitored to determine if the scan was

detected by the IDS. The number of tests was an arbitrary number chosen based on the previous research to more than adequately test each of the programs.

FTP scans. Testing software simulated an FTP vulnerability attack. This test was conducted ten separate times for each IDS. Each IDS was monitored to determine if the scan was detected by the IDS. The number of tests was an arbitrary number chosen based on the previous research to more than adequately test each of the programs.

SMTP scans. Testing software simulated a SMTP vulnerability attack. This test was conducted eight separate times for each IDS. Each IDS was monitored to determine if the scan was detected by the IDS. The number of tests was an arbitrary number chosen based on the previous research to more than adequately test each of the programs.

POP3 scans. Testing software simulated a POP3 vulnerability attack. This test was conducted eight separate times for each IDS. Each IDS was monitored to determine if the scan was detected by the IDS. The number of tests was an arbitrary number chosen based on the previous research to more than adequately test each of the programs.

ICMP scans. Testing software simulated an ICMP attack. This test was conducted eleven separate times for each IDS. Each IDS was monitored to determine if the scan was detected by the IDS. The number of tests was an arbitrary number chosen based on the previous research to more than adequately test each of the programs.

File Deletion. The testing involved selecting an OS system file and deleting. This test was conducted three separate times for each IDS. The IDS was monitored to determine if the IDS would alert to the deletion. The number of tests was an arbitrary number chosen based on the previous research to more than adequately test each of the programs.

File relocation. The testing involved selecting an OS system file and relocating on the hard drive. This test was conducted four separate times for each IDS. The IDS was monitored to determine if the IDS would alert to the deletion. The number of tests was an arbitrary number chosen based on the previous research to more than adequately test each of the programs.

File copy. The testing involved selecting an OS system file and selecting the option to copy the file. This test was conducted two separate times for each IDS. The IDS was monitored to determine if the IDS would alert to the deletion. The number of tests was an arbitrary number chosen based on the previous research to more than adequately test each of the programs.

The file deletion, file relocation, and file copy test were tests that simulated internal vulnerabilities of the network. All of the tests performed in this group were done without any network load or background traffic.

Test Group 2. This phase included increasing simulated background traffic to different variances of network load. Under this simulated load, each of the tests from Test Group 1 were repeated. The tests were conducted simulating a network load of twenty megabits per second (Mbits/sec.).

The automatic traffic generator used for the testing was the Linux based, open source program, Mausezahn. The program was created and designed to allow for traffic generation for stressing networks as well as penetration testing of firewalls and IDS programs. It also has the ability to produce DoS attacks, ping attacks as well as port scanning, but for the purpose of the test only the traffic generator ability was utilized. There were no hidden or stealth attacks included in the background traffic, it was only

used to increase network load and simulate real network traffic in a business environment.

Test Group 3. This phase included increasing simulated background traffic to different variances of network load. Under this simulated load, each of the tests from Test Group 1 were repeated. The tests were conducted simulating a network load of forty megabits per second (Mbits/sec.).

Test Group 4. This phase included increasing simulated background traffic to different variances of network load. Under this simulated load, each of the tests from Test Group 1 were repeated. The tests were conducted simulating a network load of 60 megabits per second (Mbits/sec.).

Phase 3 Test Results and Product Comparisons

Snort.

Test Group 1. Snort detected all nineteen of the Port Scan tests, all thirteen of the DoS scan tests, all ten of the web scan tests, all ten of the FTP scan tests, all eight of the SMTP Scan tests, all eight of the POP3 scan tests, all eleven of the ICMP Scan tests, all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests.

Test Group 2. Snort detected all nineteen of the port scan tests, all thirteen of the DoS scan tests, all ten of the web scan tests, all ten of the FTP scan tests, all eight of the SMTP scan tests, all eight of the POP3 scan tests, all eleven of the ICMP scan tests, all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests.

Test Group 3. Snort detected all nineteen of the port scan tests, all thirteen of the DoS scan tests, all ten of the web scan tests, all ten of the FTP scan tests, all eight of the SMTP scan tests, all eight of the POP3 scan tests, all eleven of the ICMP scan tests, all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests.

Test Group 4. Snort detected all nineteen of the port scan tests, twelve of the thirteen DoS Scan tests, all ten of the web scan tests, all ten of the FTP scan tests, all eight of the SMTP scan tests, all eight of the POP3 scan tests, all eleven of the ICMP scan tests, all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests. Refer to Appendix A for test results.

OSSEC.

Test Group 1. OSSEC detected all nineteen of the port scan tests, all thirteen of the DoS scan tests, all ten of the web scan tests, all ten of the FTP scan tests, all eight of the SMTP scan tests, all eight of the POP3 scan tests, all eleven of the ICMP scan tests, all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests.

Test Group 2. OSSEC detected all nineteen of the port scan tests, all thirteen of the DoS scan tests, all ten of the web scan tests, all ten of the FTP scan tests, all eight of the SMTP scan tests, all eight of the POP3 scan tests, all eleven of the ICMP scan tests, all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests.

Test Group 3. OSSEC detected seventeen of the nineteen port scan tests, nine of the thirteen DoS scan tests, all ten of the web scan tests, nine of the ten FTP scan tests, all

eight of the SMTP scan tests, all eight of the POP3 scan tests, ten of the eleven ICMP scan tests, one of the three file deletion tests, all four of the file relocation tests, and each of the two file copy tests.

Test Group 4. OSSEC detected fourteen of the nineteen port scan tests, seven of the thirteen DoS scan tests, all ten of the web scan tests, seven of the ten FTP scan tests, all eight of the SMTP scan tests, all eight of the POP3 scan tests, ten of the eleven ICMP scan tests, one of the three file deletion tests, one of the four file relocation tests, and each of the two file copy tests. Refer to Appendix B for test results

Prelude.

Test Group 1. Prelude detected all nineteen of the port scan tests, all thirteen of the DoS scan tests, all ten of the web scan tests, all ten of the FTP scan tests, all eight of the SMTP scan tests, all eight of the POP3 scan tests, all eleven of the ICMP scan tests, all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests.

Test Group 2. Prelude detected all nineteen of the port scan tests, all thirteen of the DoS scan tests, all ten of the web scan tests, all ten of the FTP scan tests, all eight of the SMTP scan tests, all eight of the POP3 scan tests, all eleven of the ICMP scan tests, all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests.

Test Group 3. Prelude detected all nineteen of the port scan tests, all thirteen of the DoS scan tests, nine of the ten web scan tests, all ten of the FTP scan tests, all eight of the SMTP scan tests, all eight of the POP3 scan tests, all eleven of the ICMP scan tests,

all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests.

Test Group 4. Prelude detected all nineteen of the port scan tests, all thirteen of the DoS scan tests, all ten of the web scan tests, all ten of the FTP scan tests, seven of the eight SMTP scan tests, all eight of the POP3 scan tests, nine of the eleven ICMP scan tests, all three of the file deletion tests, all four of the file relocation tests, and each of the two file copy tests. Refer to Appendix C for test results.

The direct test of each IDS program was intended to provide additional necessary information for this project. In the next section the author will present the findings of the research conducted. Each program will be discussed and the information garnered from the research conducted will be highlighted

Discussion of the Findings

Network security is a constant struggle to stay current, to stay relevant and to remain secure. The problem is that there are varying opinions of what network security is and how to go about achieving it. At a basic level there are certain components that are required to dictate a certain level of security. Beyond those basic measures is where the debates and theories begin. The purpose of this study was to attempt to determine the effectiveness of open source IDS programs.

This project presented a variety of research literature to provide a complete perspective of open source IDSs. The research material included scholarly journals, research studies, professional projects, and published books. In addition to the literature reviewed, the author conducted controlled testing of 3 open source IDS programs. The following are the findings of this study.

As the literature review indicated there are different reasons for deploying an IDS program into a network; governmental compliance, protecting proprietary company information and protecting customers financial or personal data are just a few. The Internet is constantly evolving and because of that, network security must evolve along with it. Additional measures must be taken in order to safely and securely provide the network and system administrators with the tools necessary to provide that security and compliance.

There are also reasons for not implementing an IDS program. The majority of those reasons revolve around the cost associated with the deploying a system, as well as the maintenance and continuing support that is required. The inverse though is the potential costs associated with a successful attack against a network or the costs related to penalties for non compliance with the requirements of privacy laws. There are laws set forth by the government that require certain facilities to maintain a minimum amount of information security and if those requirements are not met the company can be fined. FISMA as well as HIPAA are examples of regulations dictating how personal data must be handled by a company. IDSs can facilitate complying with those regulations as well hardening networks for information security.

Snort

Snort was tested on both a Microsoft Windows based and a Linux based network server. The installation process for each server environment was easy and offered no difficulties. Snort's performance during the testing was excellent, as it detected all of the vulnerability scans throughout all phases of the testing including all iterations of network load. It was equally effective at detecting the threats from inside the network associated

with the three tests conducted. First test was a file copy, second was moving a file from directory to directory and third was deletion of files.

OSSEC

OSSEC was tested on both a Microsoft Windows based and a Linux based server. OSSEC performed well against the baseline tests. There were reduced numbers when the network load was increased, but still the program was competent with the detection rates. The program also detected the inside threats very well, discovering all three tests without a false positive or missed attempt.

Installation on the Windows system was easy with no hindrances. Installation was a bit more complicated on the Linux system. There are three options for instillation server, agent or local. The system will prompt the user to choose at the start of the install process. The program also provides suggestions as to which choice would be best in most situations. For this testing, the local install was chosen along with the default settings of all of the install prompts. Overall, the install process for OSSEC was quite easy.

Rule and system updates are just as easy as the initial instillation. Updating was not performed during the testing because this system does not provide signature updates like Snort. Updates would involve updating to a newer version of the program, by uninstalling the old version and installing the new.

Prelude

Prelude was only tested on Linux OS server, because it is not compatible with Windows OS at the current time. Prelude was more cumbersome to install than the previous two systems. The installation was complicated by the dependency files that also needed to be installed. The install packages were not available on the developer's web

site similar to Snort and OSSEC. The only way to install Prelude was to use a package manager from one of the supported OSs. Ubuntu is a supported OS; the *apt-get* command was used to install the program. In general, Prelude is slightly more complicated than the previous two systems, not necessarily because the source binaries were not available, but because Prelude is a modular system that has several different aspects to it. There are several other packages that are needed to be installed along with the main system. These packages are for different functions that can be utilized with this system.

Prelude performed well during the testing, producing results similar to Snort and Ossec. Under network loads the vulnerability scans produced results that were a bit less than Snort, but better OSSEC. The inside tests were detected without a problem as well. Regarding updating the system, there are no rules to update. That aspect of this system was not tested but according to the documentation it is just as easy as the other programs to facilitate.

Maintenance

All of the systems will have maintenance requirements after initial installation. For Snort, that would involve the downloading and integration of new signatures for the existing database. This would have to be done every time a new threat is detected and a signature is developed to detect and prevent it. These signatures are maintained and supplied by the developer's website and also via third party web sites. For Prelude and OSSEC maintenance would consist of installing the newest version of the software for enhancements and more proficient threat detection engines.

Technical staff

All of the systems tested require a trained staff to install and provide maintenance on a regular basis. IDSs are not like firewalls that can be simply put into place and forgotten about. All three systems at certain times can be demanding on technical staff in regards to having to configure, update, monitoring of the log files, as well as the programs alerts and false alarms.

Effective Rates

In the previous tests of IDS programs that were reviewed for this study, the researchers were very detailed and precise to duplicate background network traffic as well as trying to reproduce automatic attack signatures. Those tests were automated to try and minimize any changes to the corpus or data sets that were introduced. This has a tendency to lead to very static and almost predictable results.

All of the IDS programs tested in this study performed well against nuisance attacks, similar to those of an amateur hacker. There is no system that would be able to prevent a constant barrage of attacks from a competent and motivated cyber attacker. Even if one of these systems does not stop an attack completely, it will be able to record information about how the perpetrator got into the system, what they did while they were in the network and when the incident occurred. Being able to produce timeline records of incidents is extremely important for a law enforcement investigation and criminal prosecution.

Previous Testing Results

The Neohapsis/Network-Computing evaluation tests focused on ease of use, stability, cost effectiveness, signature quality and ease of customization. The researchers

did try and reproduce background traffic by mirroring traffic from DePaul University in Chicago. The NSS Group evaluated the programs based on their ease of installation and configuration, types of reporting analysis and the systems architecture. The NSS tests varied from the Neophapsis/Network-Computing tests in that NSS did not utilize any background network traffic, but they did use packet manipulation and IDS evasion techniques. Both of these tests produced copious amounts of data and evaluations about the various systems. These tests result were in addition to the first data sets created by MIT studies, funded by DARPA. The purpose of all these tests was to reproduce the same network traffic to determine how the different IDS programs operate as well as making changes and improvements to them.

Cost

As indicated in the previous sections, even though these systems are essentially free of licensing fees, there are still costs associated with deploying them. Time and technical resources can be very expensive if a company does not currently have them available. A company might either need to hire personnel, or utilize a third party company to install and maintain the IDS. The most significant aspect of cost is related to the system on which the programs need to be installed and function on. None of these IDS programs are very demanding in regards to system resource needs. In fact each of these programs can be run on older servers, as opposed to having to buy a new, more powerful or expensive server to just to be able to run one of these programs, which is a significant cost savings. Refer to Table 1 for a description of each program.

IDS	Outside Vulnerability Assessment	Inside Vulnerability Assessment	Installation		Updates	
			Easy	Hard	Easy	Hard
Snort	S	S	√		√	
OSSEC	U	S	√		√	
Prelude	S	S		√	√	

Table 1: Test Results (S = Satisfactory, U = Unsatisfactory)

Limitations of the Study

A limitation of the study was the lack of available research material on open source IDS. Documentation and information pertaining to open source programs is sparse and overshadowed by the commercial counterparts. There were no existing studies on the direct evaluation of strictly open source IDS programs. In the majority of the available literature, if an open source program was identified it was referenced in comparison to the commercial IDS programs.

Another limitation to the study was time constraints. If time permitted the study would have benefited from being able to examine more open source programs. There are several older IDSs that are still considered competent. Including these programs would have been a beneficial in being able to examine how they would perform against the programs tested. In contrast, these older programs, though they are still available, they are no longer being maintained or updated by their developers.

A specific restriction to the study was no human involvement. This restriction negated the ability to incorporate questionnaires or surveys of industry professionals. Direct input from security professionals as well as network administrators about their

professional experience with different systems and configurations could have helped the testing and methodology of testing.

Lastly, the fact that Prelude currently is not cross compatible is a limitation to the study. At this time Prelude is only available on Linux OS, but currently not available for Windows OS. Being able to test all three systems on multiple OSs would have provided a more comparable series of findings.

This concludes the Discussion of Findings section. The findings of the existing research combined with the findings of the research conducted by the author were presented. In the following section the author will present recommendations and conclusions based on these findings.

Recommendations

This study evaluated open source intrusion detection systems and as such this study intended to answer the following questions: How difficult is each program to set up and configure? How difficult is the signature updating process and availability? What were the detection rates for each system under normal conditions using basic vulnerability scanning tools? The recommendations and conclusions are based on the review of existing literature and actual testing of IDS programs by the author.

Comparing open source IDS programs against each other was the best way to evaluate each of the products individually, as well as collectively, in order to determine what advantages or disadvantages one might have over the others. Based on the findings of this study, the best open source IDS program was Prelude. This is the epitome of open source programs and it provides the most forward thinking system built to date. This system is actually multiple systems in one. The ability to have to aggregate system log

files from almost every other device on a network, being able to integrate with other programs, like Snort and OSSEC, and provide its own IDS engines are what sets Prelude apart as the best available open source program tested.

Prelude can be utilized for just one or all aspect of its capabilities because it is a modular system. A user can choose which part of the program to utilize and as new modules are produced, either by the developer or by the user's network security personnel, they can simply be incorporated into the program. This is the same for adding additional network appliances, as such Prelude can be expanded to add another switch, VPN firewall or another network server which can simply be added it to the SIEM.

Recommendations for Future Research

Hybrid IDS programs, like Prelude, are going to become more prevalent. Being able to incorporate all of the positive aspects of multiple IDS programs into one product is where the network security field is transitioning to. Currently, there are a few commercial hybrid systems available, which are very expensive. There have been no direct evaluations conducted on these systems.

Another suggestion for future research involves IPS technology being directly incorporated into firewall applications. IPS programs are different from IDS programs in that they can actively prevent a detected intrusion, where IDS programs can only alert detected intrusions. These types of systems can allow network administrators to take a more offensive approach to protecting a network from unauthorized intrusion.

Conclusions

The purpose of this study was to evaluate and compare three open source, free, IDS programs. The evaluated programs were Snort, OSSEC, and Prelude. The evaluation

and testing intended to answer the following questions: How difficult is each program to set up and configure? How difficult is the signature updating process and availability? What were the detection rates for each system under normal conditions using basic vulnerability scanning tools?

The evaluation of the three IDS programs consisted of an extensive review of existing literature. Moreover, original research through testing of each of the IDS programs was conducted in a dedicated and controlled environment. Based on the research findings, this study will propose suggestions that business owners, information IT managers, or network security administrators might use in selecting an open source IDS program.

In conclusion attacks on computer networks are a constant threat. IDS programs are a necessity for a complete network security system. Today, there is a lack of funding for network security due to companies' reduced budgets, but this should not be the determining factor for not utilizing this technology. All of the programs chosen for this study, Snort, OSSEC, and Prelude, are available for free, therefore eliminating issues regarding lack of funding or resources.

These three programs performed well in the testing conducted for this study. Each of these programs is as capable of providing the protection needed to secure a network as their commercial counterparts. Open source programs might not offer all of the extra features of commercial programs, but for filling the need for basic network security these programs are more than adept. Though all of the programs performed extremely well the one system that performed the best overall was Prelude.

Appendix A

Test Results for the IDS program - Snort

Snort	# of Tests	Test Group 1	Test Group 2	Test Group 3	Test Group 4
Port Scans	**19**	**19**	**19**	**19**	**19**
DoS	**13**	**13**	**13**	**13**	**12**
Web	**10**	**10**	**10**	**10**	**10**
FTP	**10**	**10**	**10**	**10**	**10**
SMTP	**8**	**8**	**8**	**8**	**8**
POP3	**8**	**8**	**8**	**8**	**8**
ICMP	**11**	**11**	**11**	**11**	**10**
File Deletion	**3**	**3**	**3**	**3**	**3**
File Relocation	**4**	**4**	**4**	**4**	**4**
File Copy	**2**	**2**	**2**	**2**	**2**

Appendix B

Test Results for the IDS program - OSSEC

OSSEC	# of Tests	Test Group 1	Test Group 2	Test Group 3	Test Group 4
Port Scans	**19**	**19**	**19**	**17**	**13**
DoS	**13**	**13**	**13**	**9**	**7**
Web	**10**	**10**	**9**	**9**	**9**
FTP	**10**	**10**	**10**	**9**	**7**
SMTP	**8**	**8**	**8**	**8**	**8**
POP3	**8**	**8**	**8**	**8**	**8**
ICMP	**11**	**11**	**11**	**10**	**10**
File Deletion	**3**	**3**	**2**	**1**	**1**
File Relocation	**4**	**4**	**2**	**2**	**1**
File Copy	**2**	**2**	**2**	**2**	**2**

Appendix C

Test Results for the IDS program - Prelude

Prelude	# of Tests	Test Group 1	Test Group 2	Test Group 3	Test Group 4
Port Scans	**19**	**19**	**19**	**19**	**19**
DoS	**13**	**13**	**13**	**13**	**13**
Web	**10**	**10**	**10**	**9**	**10**
FTP	**10**	**10**	**10**	**10**	**10**
SMTP	**8**	**8**	**8**	**8**	**7**
POP3	**8**	**8**	**8**	**8**	**8**
ICMP	**11**	**11**	**11**	**11**	**9**
File Deletion	**3**	**3**	**3**	**3**	**3**
File Relocation	**4**	**4**	**4**	**4**	**4**
File Copy	**2**	**2**	**2**	**2**	**2**

Bibliography

Allen, J., Christie, A., Fithen, W., McHugh, J., Pickel, J., & Stoner, E. Carnegie Mellon University, (2000). *State of the practice of intrusion detection technologies*. Retrieved from website: http://www.sei.cmu.edu/library/abstracts/reports/99tr028.cfm

Alward, R. G., Carley, K. M., Madsen, F., Taylor, V. K., & Vandenberghe, G. Defence Research and Development Canada Ottawa (Ontario), (2006). *Network vulnerability and risk assessment* (ADA477100). Retrieved from http:// www.dtic.mil/dtic/tr/fulltext/u2/a477100.pdf

Anderson, J. P. James P. Anderson Co., (1980). *Computer security threat monitoring and surveillance*. Retrieved from website: http://csrc.nist.gov/publications/history/

Bace, R., & Mell, P. National Institute of Standards and Technology, (2001). *Nist special publication on intrusion detection systems* (SP800-94.pdf). Retrieved from website: csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf

Hay, A., Cid, D., & Bray, R. (2008). *Host-based intrusion detection guide*. Burlington, MA: Syngress Publishing, Inc.

Cox, K. J., & Gerg, C. (2004). *Managing security with snort & ids tools*. Sebastopol, CA: O'Reilly Media, Inc.

Computer History Museum. (2006). *A history of the internet: 1962-1992*. Retrieved from Computer History Museum website: http://www.computerhistory.org/internet_history/

Daya, B. University of Florida Department of Electrical and Computer Engineering, (n.d.). *Network security: History, importance, and future*. Retrieved from website: http://web.mit.edu/~bdaya/www/Network Security.pdf

Del Carlo, C. (2003). *Intrusion detection evasion: How attackers get past the burglar alarm*. Retrieved from SANS Institute website: Http://www.sans.org/reading_room/whitepapers/detection/intrusion-detection-evasion-attackers-burglar-alarm_1284

Denning, D. E. (1982). *Cryptography and data security*. Boston, MA: Addison-Wesley.

Eanes, M. (2003). *Wanted dead or alive: Snort intrusion detection system*. Retrieved from SANS Institute website: http://www.sans.org/reading_room/whitepapers/detection/wanted-dead-alive-snort-intrusion-detection-system_1275

Haines, J. W., Lippman, R. P., Fried, D. J., Zissman, M. A., Tran, E., & Boswell, S. B. Defense Advanced Research Projects Agency, (2001). *1999 darpa intrusion detection evaluation:design and procedures* (Technical Report 1062). Retrieved from Lincoln Laboratory website: http://www.ll.mit.edu/mission/communications/ist/files/TR-1062.pdf

Ho, S. Y. (2001). *Intrusion detection - systems for today and tomorrow*. Retrieved from SANS Institute website: http://www.sans.org/reading_room/whitepapers/detection/intrusion-detection-systems-today-tomorrow_341

Kent, K., & Souppaya, M. (2006). Guide to computer security log management (*NIST Special Publication 800-92*). Retrieved from http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf

Kozushko, H. (2003). *Intrusion detection: Host-based and network -based intrusion detection systems*. Retrieved from New Mexico Tech website: http://infohost.nmt.edu/~sfs/Students/HarleyKozushko/Papers/IntrusionDetectionPaper.pdf

Lennon, E. B. National Institute of Standards and Technology, Information Technology Laboratory. (2003). *Testing intrusion detection systems*. Retrieved from National Institute of Standards and Technology website: http://csrc.nist.gov/publications/nistbul/bulletin07-03.pdf

Lydon, A. (2004). *Complication for intrusion detection systems*. (Master's thesis, Ohio University)Retrieved from http://etd.ohiolink.edu/send-pdf.cgi/Lydon Andrew.pdf?ohiou1088179093

McMillan, J., & Kibirkstis, A. SANS Institute, (2009). *Sans institute updates popular intrusion detection resource for information security professionals*. Retrieved from SANS Institute website: http://www.sans.org/security-resources/idfaq/intrusion-detection-update.php

Mell, P., Hu, V., Lippmann, R., Haines, J., & Zissman, M. National Institute of Standards and Technology, Computer Security Devision. (2003). *An overview of issues in testing intrusion detection systems* (NIST IR 7007). Retrieved from NIST website: http://csrc.nist.gov/publications/PubsNISTIRs.html

Meyer, R. (2008). *Challenges of managing an intrusion detection system (ids) in the enterprise*. Retrieved from SANS Institute website: http://www.sans.org/reading_room/whitepapers/detection/challenges-managing-intrusion-detection-system-ids-enterprise_2128

Microsoft, (2006). *Regulatory compliance demystified: An introduction to compliance for developers*. Retrieved from Microsoft MSDN website: http://msdn.microsoft.com/en-us/library/aa480484.aspx

Nijnik, I. (2007). *Small business network security 101*. Retrieved from IT World Canada website: http://www.itworldcanada.com/WhitePaperLibrary/PdfDownloads/SmallBusinessNetworkSecurity101.pdf

Phatak, P. (2011, January 01). The importance of intrusion prevention systems. *Linux For You*, Retrieved from http://www.linuxforu.com/2011/01/importance-of-intrusion-prevention-systems

Portelli, B. (2010, April 16). Why open source?. *Tech News World*, Retrieved from http://www.technewsworld.com/story/69788.html?wlc=1272461886

Rantala, R. R. U.S. Department of Justice, Bureau of Justice Statistics. (2008). *Cybercrime against businesses, 2005* (NCJ 221943). Retrieved from U.S. Department of Justice website: http://bjs.ojp.usdoj.gov/content/pub/pdf/cb05.pdf

Roesch, M. (1999, November). *Snort-lightweight intrusion detection for networks*. Retrieved from http://static.usenix.org/event/lisa99/full_papers/roesch/roesch.pdf

SANS Institute, (2001). *Intrusion detection systems; definition, need and challenges*. Retrieved from SANS Institute website: http://www.sans.org/reading_room/whitepapers/detection/intrusion-detection-systems-definition-challenges_343

SANS Institute, (2004). *Maintaining a secure network*. Retrieved from SANS Institute website: http://www.sans.org/reading_room/whitepapers/detection/maintaining-secure-network_1445

Sequeira, D. SANS Institute, (2002). *Intrusion prevention systems-security's silver bullet?*. Retrieved from SANS Institute website: http://www.sans.org/reading_room/whitepapers/detection/intrusion-prevention-systems-securitys-silver-bullet_366

Whitman, M. E., & Mattord, H. J. (2009). *Principles of information security*. (Third ed.). Boston, MA: Course Technology Ptr.

Winkelman, D. R. (2011). What is a network?. In *An Educator's Guide to School Networks* Retrieved from http://fcit.usf.edu/network/chap1/chap1.htm

Wippich, B. SANS Institute, (2007). *Detecting and preventing unauthorized outbound traffic*. Retrieved from SANS Institute website: http://www.sans.org/reading_room/whitepapers/detection/detecting-preventing-unauthorized-outbound-traffic_1951

The NSS Group, (2001). *Intrusion detection systems group test (edition 2)*. Retrieved from website: http://www.qnet.it/pdf/ids.pdf

Zaraska, K. Pennsylvania State University, The College of Information Sciences and Technology. (2003). *Prelude ids: Current state and development perspectives.* Retrieved from Pennsylvania State University website: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.106.5542