A Comparison of Four Intrusion Detection Systems for Secure E-Business

C. A. P. Boyce, A. N. Zincir-Heywood
Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada
{boyce, zincir} @ cs.dal.ca

Abstract

This paper evaluates three rule-based open sourced network intrusion detection systems – Snort, Firestorm, Prelude – and one rule-based commercial system – Dragon. The 1999 DARPA Dataset, which is the only public data set used for IDS benchmarking to the best of authors' knowledge, is used to perform the evaluations. Results discuss how each system performed and the possible benefits of any one system over the other.

1. Introduction

Intrusion Detection Systems (IDS) play an important role in an organization's security framework. Security tools such as anti-virus software, firewalls, packet sniffers and access control lists aid in preventing attackers from gaining easy access to an organization's systems but they are in no way foolproof. Today, many organizations are embracing e-business. For companies whose main revenue is dependent upon e-business, the downtime associated with an attack can result in the loss of hundreds of thousands of dollars. In addition, loss of consumer confidence may put a company out of business. These factors make the need for a proper security framework even more paramount. A tool is therefore needed to alert system administrators to the possibility of rogue activities occurring on their networks. Intrusion Detection Systems can play such a role. Therefore a need exists to understand how open source tools of this type compare against commercial ones.

Many companies engaging in online business activities unfortunately do not see security as an important issue. From a business standpoint this may be because the return of investment in security is not immediately noticed. Additionally, implementing security tools such as IDS within an organization may be very expensive. These costs are definitely prohibitive to many small sized organizations. Thus, a study is required to be able to make an effective decision in selecting an intrusion detection system. In this work, three open source intrusion detection systems – Snort, Firestorm, Prelude – and a commercial intrusion detection system, Dragon, are evaluated using DARPA 1999 data set in order to identify the factors that will effect such a decision.

The remainder of the paper is organized as follows. Section 2 introduces intrusion detection systems under evaluation. Section 3 presents the test environment and procedures set up for this work. Results are given in section 4 and conclusions are drawn in section 5.

2. Intrusion Detection Systems

Intrusion Detection Systems fall into two categories, Network based intrusion detection systems (NIDS) and Host based intrusion detection systems (HIDS). Network Intrusion detection systems operate by analyzing network traffic whereas Host based systems analyze operating system audit trails. Within these two, their method of detection are categorized based upon two criteria, anomaly or pattern detection. Systems based upon anomaly detection build a profile of what can be considered normal usage patterns over a period of time and trigger alarms should anything deviate from this behaviour. Within this type of detection lies a subsection which is based on protocol standards. Pattern detection identifies intrusions based upon known intrusion techniques and trigger alarms should these be detected.

The objective of the authors is to compare three rule-based open source network intrusion detection systems with one rule and anomaly based commercial system. This was carried out using the 1999 DARPA Dataset, which is the only IDS benchmarking to the best of the authors' knowledge. The following describes the tools under evaluation.

2.1 Snort

Snort is an open source network intrusion detection system capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis and content searching/matching in order to detect a variety of attacks and probes such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts and more. It uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plugin architecture [1].

2.2 Firestorm

Firestorm is a high performance network intrusion detection system. It is fully pluggable and hence extremely flexible. It is capable of detecting a wide variety of attacks. In addition it has decode plugins available for many protocols, preprocessors to allow supplementary modes of detection, full IP defragmentation and intelligent TCP stream reassembly among other features. Entries are made to a log file in text format and it has the ability to log to a remote management console [2].

2.3 Prelude

Prelude is a general purpose hybrid intrusion detection system. It is divided into several parts a network intrusion detection system and a reporter server. The network intrusion detection system is responsible for packet capture and analysis. Its signature engine is designed to read Snort rulesets but, it also has the capability to load rulesets from any most Network Intrusion Detection Systems. The report server is reported to by the NIDS contacts and logs all intrusions. This architecture allows for several sensors to be deployed throughout a network all report to one central management console [3].

2.4 Dragon

Dragon is a rule and anomaly based commercial intrusion detection system with an extensive library. This allows it to be capable of detecting a wide range of attacks from network attacks and probes to successful system compromises and backdoors. The system used in this evaluation is however only a trial download and comes with about a third of the signature database [4].

Intrusion Detection Systems fit into three categories. Some work by detecting attacking attacks as they occur in real time. These can be used to monitor and possibly stop an attack, as it is occurring. Others are used to provide forensic information about attacks after they occur. This information can be used to help repair damage, understand the attack mechanism and reduce the possibility of future attacks on the same type. The final category of systems can detect never seen before new attacks. The open source IDS fit into the category of those providing forensic information. The commercial system fits into the category of detecting new attacks as well as providing forensic information.

3 Test Set Up and Procedures

To carry out testing of the Intrusion Detection Systems, use was made of the Darpa data set, Tcpreplay, two Pentium three 850 MHz computers and cross-coupled network cable.

The Darpa 1999 data set as stated earlier the only known IDS benchmarking dataset to the authors' knowledge consists of network traffic and audit logs collected over the course of five weeks from a simulated network, figure1.

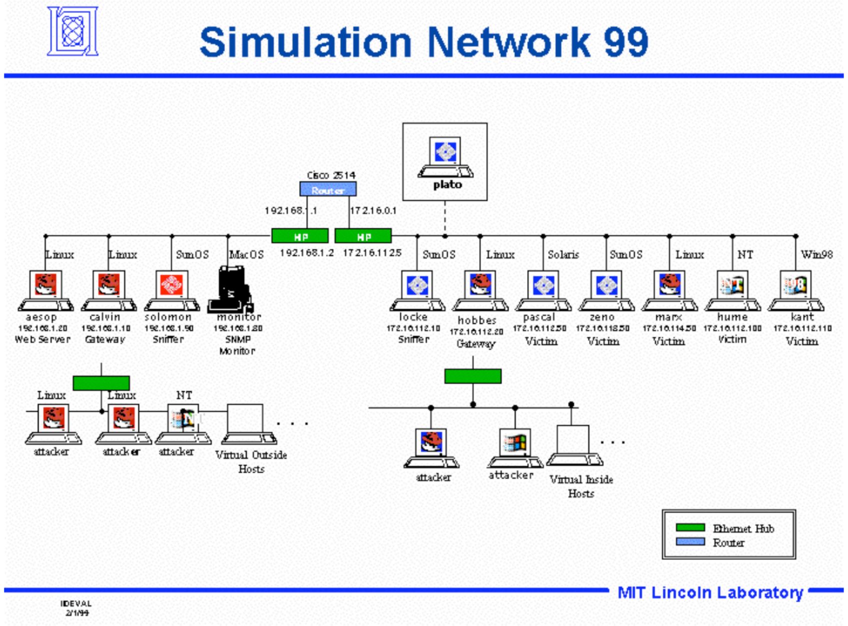

Figure 1: The simulated network used for testing [5]

The test bed consisted of four victim machines which are the most frequent targets of attacks in the evaluation (Linux 2.0.27, SunOS 4.1.4, Sun Solaris 2.5.1, Windows NT 4.0), a sniffer to capture network traffic, and a gateway to hundreds of other emulated PCs and workstations. The outside simulated Internet contained a sniffer, a gateway to emulated PCs on many subnets and a second gateway to thousands of emulated web servers. Data collected for evaluation included network sniffing data from both inside and outside sniffers, Solaris Basic Security Module (BSM) audit data collected from the Solaris host, Windows NT audit event logs collected from the Windows NT hose, nightly listings of all files on the four victim machines and nightly dumps of security-related files on all victim machines [6]. This data was collected over the course of 5 weeks.

The test bed consisted of several million connections for TCP services. This was dominated by web traffic, however several other forms of services were used including mailing services, ftp to send and receive files, telnet and ssh for remote log into computers.

Five attacks types were inserted into the traffic and were of the following:

Denial of Service Attacks:

This is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine [7]. Several types of DOS attacks were used within the test bed. These ranged from attacks which abused perfectly legitimate features. Some which created malformed packets that confused the TCP/IP stack of the machine trying to reconstruct the packet and lastly, those which took advantage of bugs in a particular network daemon.

User to Root Attacks:

This class of attack begins with the attacker gaining access to a normal user account on the target machine by one of a number of methods be it password sniffing, social engineering or a dictionary attack. The attacker then attempts to gain root access on the system by exploiting a known or unknown vulnerability. The most common form of this attack is the buffer overflow attack where a program copies to a static buffer more information than it can hold. The result of this is the attacker can cause arbitrary commands to be executed on the operating system.

Remote to Local Attacks:

This type of attack occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine [8]. This can be carried out by exploiting buffer overflows in network server software. Another method of carrying out this attack is to exploit weak or misconfigured system security policies.

Probes:

These do not fit into the category of attacks but are actually programs which automatically scan networks of computers to gather information or find known vulnerabilities [9]. Several types of probes were used in the test bed. They were those that determine the numbers of machines on the network. Those that determine which services are running on a particular system. Finally, there were those that determined the names or other information about users with accounts on a given system.

Data:

These attacks involve either a user or administrator performing some action that they may be able to do on a given computer system, but that they are not allowed to do according to site policy. Often, these attacks will involve transferring "secret" data files to or from sources where they don't belong [10].

3.1 Experiment

Traffic collected from week 4 consisting of logs from both the inside and outside sniffers was used for the evaluation purposes. Reasons for this being, the first 3 weeks contained training data, which allowed the intrusion detection systems to become familiar with normal traffic patterns, while week 5's data was considerably larger and would have taken longer to run.

To replay the captured traffic the Tcpreplay utility created by SourceForge was employed. Tcpreplay replays packets captured from a live network, additional functionality allows for the traffic to be replayed at various speeds including that at which it was captured. This is done in hope that Network Intrusion Detection Systems can be more thoroughly tested [11].

The four systems were downloaded, installed and configured using their default settings to one of the testing machines and the necessary supporting software downloaded. On the second machine tcpreplay and the relevant Darpa files were downloaded. The two machines were then connected using the cross-coupled network cable and each machine given an IP address differing from those contained within the dataset. Each system was started individually and a day's traffic replayed. For the majority of tests traffic was initially replayed at 1 MB/s and the time it took to replay each day varied from 25 minutes to one hour. However, to demonstrate performance under varying network throughput traffic was replayed at speeds of 2, 3, 4, 5 and 10 MB/S.

3.2 Evaluation Procedure

Each of the systems being tested produced different entries to their own log files. In order to gain useful knowledge from this data, scripts were written to extract the information. Results were compiled into a database and then further analyzed. The information extracted from each file where possible was the

- Source IP
- Destination IP
- Source Port
- Destination Port
- Description of Attack
- Rating of Attack

This information were possible was compared against the Identification Scoring truth list provided along with the data set. Each IDS log file entry was given a rating based upon one of four confidence levels:

Level 1 (C1): The intrusion detection system detects with a confidence level of 1 if the following conditions are met:

- Source IP and port in the log file matches that in the Identification Scoring truth list.
- Destination IP and port in the log file matches that in the Identification Scoring truth list.

Level 2 (C2): The intrusion detection system detects with a confidence level of 2 if the following conditions are met:

- Source IP in the log file matches that in the Identification Scoring truth list.
- Destination IP and port in the log file matches that in the Identification Scoring truth list.

Level 3 (C3): The intrusion detection system detects with a confidence level of 3 if the following conditions are met:

- Source IP and port in the log file matches that in the Identification Scoring truth list.
- Destination IP in the log file matches that in the Identification Scoring truth list.

Level 4 (C4): The intrusion detection system detects with a confidence level of 4 if the following conditions are met:

- Source IP in the log file matches that in the Identification Scoring truth list.
- Destination IP in the log file matches that in the Identification Scoring truth list.

4. Results

The results from each Intrusion Detection System were broken down into insider and outsider traffic and the results for each aggregated under the different categories. Within each of these sub categories results were then broken down into the various attack types to further see which system scored better. Additionally the assumption was made that if an Intrusion Detection System caught an attack one then chances are it would catch it again. Thus in the aggregation of these totals if an attack occurred more than once it was only listed once provided it was caught by the Intrusion Detection System. It should also be noted that these results state the number of different individual alerts for each intrusion detection system not the total number of alerts generated per attack type.

With the Insider traffic there were a total of 38 different attack types. Of the four systems under evaluation Dragon caught the most attacks with a total of 23, Firestorm caught 16, Snort 15 and Prelude 12, figure 2.

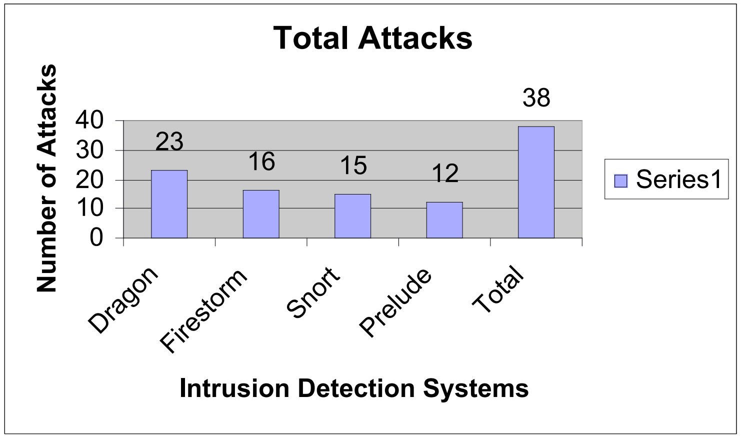

Figure 2: Total number of individual attacks caught by each system for insider traffic.

From these results we have to ask what is an acceptable lower bound for an Intrusion Detection System, the three open sourced systems all catch below 50% of the attacks types and Dragon catches a mere 60% of the attacks. For any organization this is clearly not good enough as any of the attacks, which slipped through any of the IDS, could have resulted in the crashing of the network. This is however only a top-level analysis of the results and therefore deeper, further analysis is required. Analysis at a more refined level would allow us to see if the attack types missed by the Intrusion Detection Systems were spread equally among the 5 attack types or was skewed to one particular type of attack.

As stated earlier there were five categories of attacks within the inside traffic, Denial of Service (DOS) attacks, User to Root (U2R) attacks, Remote to Local (R2L) attacks, Probe attacks and Data attacks. The R2L attacks constituted the highest number of individual attacks with a total of 20, DOS had 8 attacks, Probe 5 attacks, U2R 5 attacks of which one was a console attack so not likely to be detected by the Intrusion Detection Systems and there was just one data attack.

With reference to figure 3, within the DOS category Dragon and Snort led the way with a total of 4 attacks caught by each; Prelude caught 2 of these attacks and Firestorm 1. As stated earlier there were 5 U2R attacks but the Intrusion Detection Systems could possibly catch only 4. Of these 4 attacks Dragon caught 2, Firestorm caught 2 Prelude 1 and Snort did not catch any. Out of the 20 R2L attacks Dragon caught 14 of these; Firestorm caught 10, Snort 8 and Prelude 7.For the Probe attacks Snort caught 3 of these attacks, Prelude, Firestorm and Dragon all caught 2. The Data category consisted of only one attack; Dragon and Firestorm both caught this attack whereas Prelude and Snort did not.

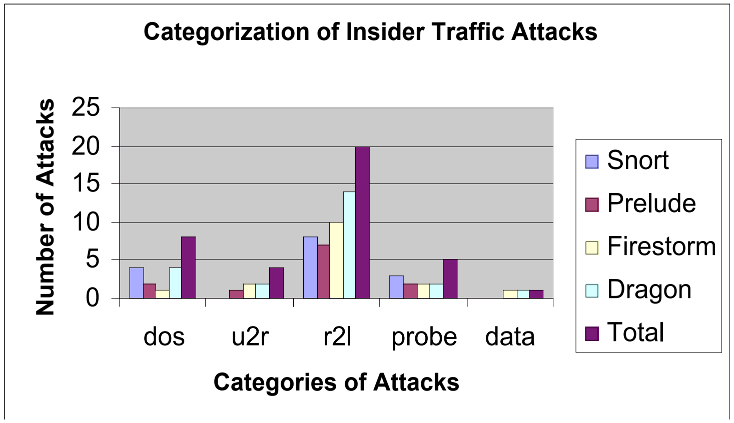

Figure 3: Attacks in the different categories in the data set for insider traffic

If we set an acceptable lower bound of 50% for each attack type to be caught by each Intrusion Detection System then we can see that for the Denial of Service attacks that Dragon and Snort barely made this threshold, Prelude scored 25% and Firestorm 12.5%. The description of Denial of Service attacks stated earlier in the paper may give a reason as to why these systems scored so lowly. Those attacks which abuse legitimate features are an example. Network Intrusion Detection systems work at lower levels of the TCP/IP stack by examining information contained within the packet headers. If the exploit abuses a feature on the target system while not breaking any of the specified rules regarding packet creation and use then it is possible for them to miss these attacks. Further investigation reveals that there were 3 attacks that were not caught by any of the Intrusion Detection Systems. A deeper investigation into the makings of these attacks being missed by the Intrusion Detection Systems reveals that they abused perfectly legitimate features. One such attack for example abused the ping command and how packets are sent on the broadcast address (xxx.xxx.xxx.255). Abusing such thus makes attack of this type more difficult to track. For these attacks to have been caught the systems would have required keeping a record of where requests were coming from. This then introduces the problem of more processing being required at the trade off of overall efficiency. Conversely, there was no individual attack that was caught by all of the systems.

In the User to Root category Dragon's performance improved as it caught 60% of the attacks. Firestorm and Prelude both met the minimum threshold of catching 50%, and Snort scored 25%. The attacks within this category employed methods such as buffer overflows, which were easily detectable to the intrusion detection systems. Additionally by using keyword spotting these attacks can be easily detectable. This is provided the attack is not made stealthier by methods such as keyword hiding. Within this category no attack was missed by all of the systems. Conversely, no attack was caught by all of the systems. One point of mention is that there were 5 attacks within this category. Of these one was carried out at the console. A console-based attack will not generate any network traffic and thus not be detected using network intrusion detection systems. So in actuality, the performance of these systems was better than originally stated.

In the Remote to Local attacks, Dragon performance improved again catching 70% of the attacks. Firestorm met the minimum threshold of 50%, Snort caught 40% and Prelude caught 35%. Five of the attacks were not caught by any of the Intrusion Detection Systems. The reasons why these attacks were missed by the IDS were not clear as no descriptions of them were found. It can however be assumed that these attacks may have been variations on another known attacks. This highlights a problem found with rule based IDS, if an attack is a slight variation from an already known one then a new signature has to be written for to catch the attack. Thus highlighting the inflexibility of using signatures. Conversely 5 attacks were caught by all of the Intrusion Detection Systems. Four of these attacks shared the common trait that they exploited an error(s) in the security policy configuration of the machines attacked. These actions then allowed the attacker to operate at higher level of privilege than intended. Exploiting a bug in a trusted program carried out the fifth attack. The actions required to carry out these attacks all leave evidence. The performance of the systems in catching these attacks can be attributed to the signatures being employed. Further, many attacks have a unique signature, which makes them difficult to alter [6]. Signature writing for this category of attack is thus easier than for other categories

In the Probe category Snort this time scored above the minimum threshold by scoring 60%. The three remaining systems scored 40%. All of the systems missed 2 of the attacks within this category. Of these, one was a probe of the machines of the network. Further investigation revealed that although this attack was not made stealthy by slow and random scanning it was probably still missed because the thresholds in the rules were set to issue alerts for only more rapid probes. Furthermore abusing a legitimate feature was how this attack was carried out. The feature abused once again is the Ping command. One attack was caught by all of the intrusion detection systems. More investigation revealed that this attack was carried out by abuse of a legitimate feature. This attack differed from those missed in that the probe was for services running. Vulnerabilities on many systems are usually well known and the attack normally follows a specified pattern. This therefore makes it possible for signatures to be written which can track when such an attack is being carried out. On the other hand, the Data category contained only one attack, Dragon and Firestorm caught this while Snort and Prelude did not.

With the Outsider traffic there were a total of 36 different attack types. Of the four systems under evaluation Dragon caught the most attacks with a total of 24, Firestorm caught 18, Snort 13 and Prelude 12, figure 4.

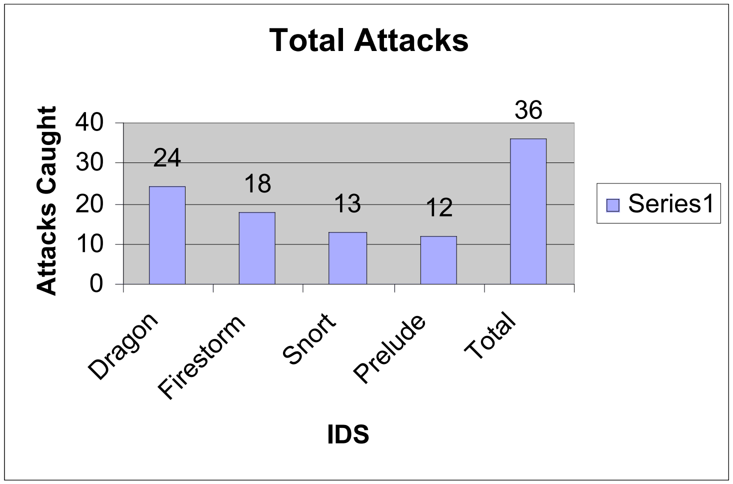

Figure 4: Total number of individual attacks caught by each system for outsider traffic.

Out of the 21 R2L attacks Dragon caught 14 of these; Firestorm caught 11, Snort 7 and Prelude 6. Within the DOS category Dragon caught 4 attacks Snort caught 3, Prelude caught 2 of these attacks and Firestorm 2. For the Probe attacks Snort caught 3 of these attacks, Prelude, Firestorm and Dragon all caught 2. There were less U2R attacks contained within the outside traffic than the inside traffic. Of these 3 attacks Dragon caught 2, Firestorm caught 2 Prelude 1 and Snort did not catch any. The Data category was made up of only one attack, Dragon and Firestorm both caught this attack whereas Prelude and Snort did not, figure 5.

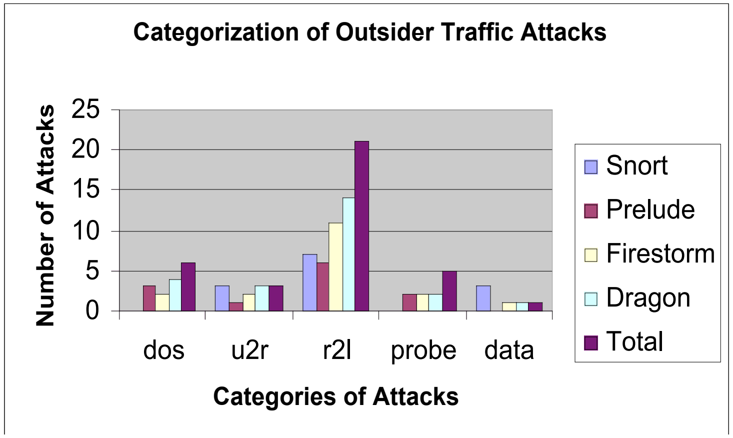

Figure 5: Number of attacks in the different categories in the data set for outsider traffic

The DOS category contained a fewer number of attacks for outside traffic than the insider traffic. There was also the introduction of one attack that was not within the insider traffic. Dragon remained the highest scorer catching approximately 67% of these attacks. Prelude scored 50%. Snort also scored 50%. Firestorm scored 16%. Two of the three attacks missed from the insider traffic were present in the outsider traffic and once again all four of the systems missed these attacks. Interesting to note is an attack missed by Firestorm in the insider logs was caught by the system in the outsider logs. Deeper analysis revealed that this attack was present in a day for which there was no insider traffic log. A possible reason for this occurrence was that fewer attacks were present in Tuesday's traffic than were in Monday's traffic. Thus the likelihood of an attack being missed is less.

The U2R category also contained fewer attacks. Also no new attacks were present in this category. Dragon scored 100%. Firestorm scored 66%. The attack not present in the outsider logs was one caught by both Prelude and Snort thus these systems scored lower. Prelude scored 33% and Snort scored 0%. In this category Dragon also missed an attack in Monday's inside traffic that it caught in Tuesday's traffic. This was attributed to fewer attacks being in Tuesday's traffic.

In contrast to the other categories there were more attacks present in the R2L category. Three of the four attacks missed by each system for the insider traffic were also missed when the outside traffic was replayed. However, there was the case that an attack caught by all of the systems when the insider traffic was replayed was missed by all of the systems in the outsider traffic. Conversely an attack missed by all of the systems from the inside traffic was caught by 3 of the systems when the outside logs were replayed.

The probe and the data category contained the same number and type of attacks for the outsider traffic. Each of the Intrusion Detection Systems scored the same in these categories.

It is not only important for IDS to be able to catch attacks but it must also show that it can maintain performance under a variety of conditions. Under increasing network throughput it is assumed that performance of Intrusion Detection Systems will decrease. The reason for this assumption being that a system would have to carry out packet analysis at a faster rate. This would then result in more intrusions being missed. Thus to test this assumption traffic a day's traffic from both inside and outside was chosen and replayed at varying speeds ranging from 2MB/S to 10 MB/s as stated earlier. The upper limit was chosen as this represented the throughput on anyone floor within the computer science department at the school. The choice of which traffic was to be replayed was based on which day had the most varied attacks. The data resulting from these results were assumed to model the behaviour for the other days.

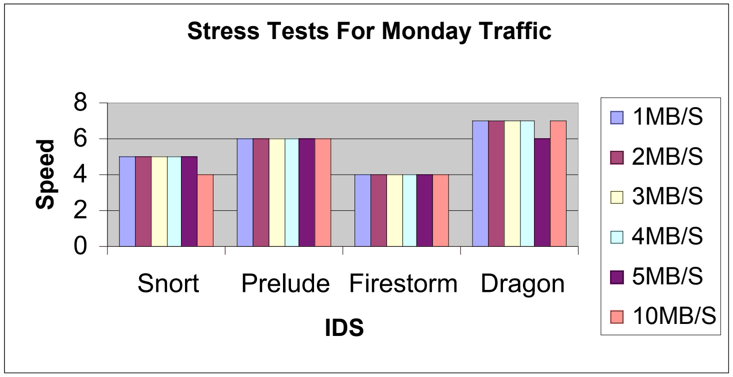

Figure 6: Number of attacks caught for inside traffic replayed at various speeds

For inside traffic, the log file for Monday was chosen. This file contained 17 individual attacks and as stated has a varied number of attacks within each attack category. Each Intrusion Detection System also performed fairly well on this traffic. Figure 6 shows that Snort's performance was consistent until it reached the upper bound then it missed an attack. Prelude was consistent for all speeds catching the same number of attacks. Firestorm was also consistent catching the same number of attacks throughout. Dragon's performance was interesting in that its performance at the upper bound was consistent with those at the lower bounds and at a rate of 5 MB/S did its performance drop off.

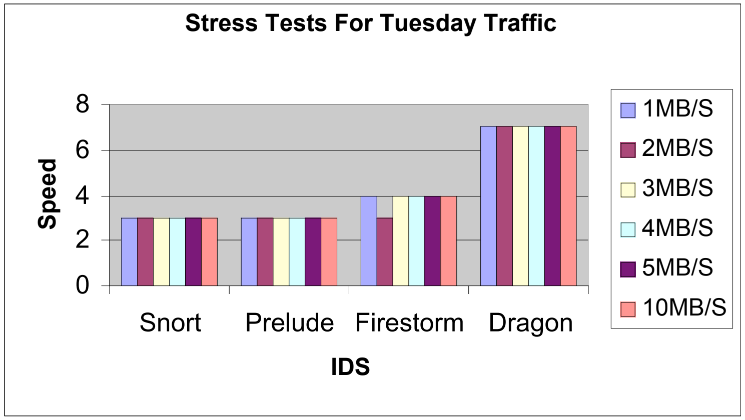

Figure 7: Number of attacks caught for outside traffic replayed at various speeds

For outside traffic, the log file for Tuesday was chosen. This file contained 12 individual attacks and as stated have a varied number of attacks. Despite not containing any attacks within the probe or data categories, the file did contain new attacks which were not found in any of the other days' traffic. Figure 7 shows that Snort's performance was consistent for all speeds. Prelude was also consistent for all speeds catching the same number of attacks. Firestorm missed an attack when the traffic was replayed at 2 MB/S but remained consistent for all other speeds. Dragon's performance unlike when the inside traffic was replayed remained consistent. This was encouraging as it showed that it could still catch a high number of attacks at high speeds.

These intrusion detection systems all require third party software in order to capture packets off the wire. Therefore, it is possible that any drop in performance was not the result of one of the systems themselves but more likely can be attributed to the supporting software.

An important metric in the measuring of intrusion detection systems is the false positive rate. Therefore an investigation was undertaken into the number of log files entries for each Intrusion Detection System. This was broken down into total inside traffic entries and total outside traffic entries. From each of these the numbers of actual attack entries were extracted to see what percentage of the whole they represented.

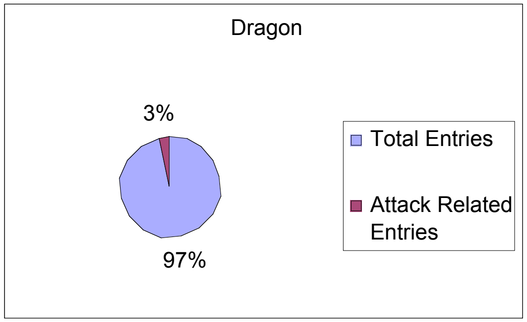

Figure 8: Size of insider traffic log file of Dragon

From figure 8, we can see that the amount of attack related entries numbered 3% of the total entries. In figures it equated to a total of 72,451 entries where only 2425 entries represented attack-related information

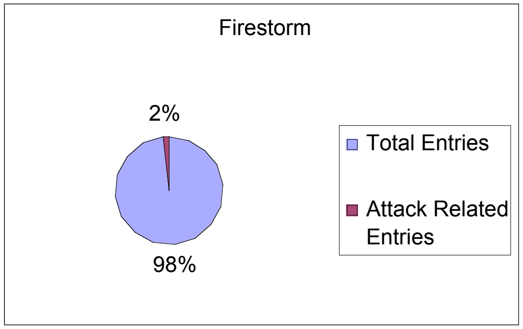

Figure 9: Size of insider traffic log file of Firestorm

Firestorm had the second highest number of entries with 3748 being the total number of entries where only 64 represented actual attack related information. Figure 9 shows that the attack related entries accounted for only 2% of the total information.

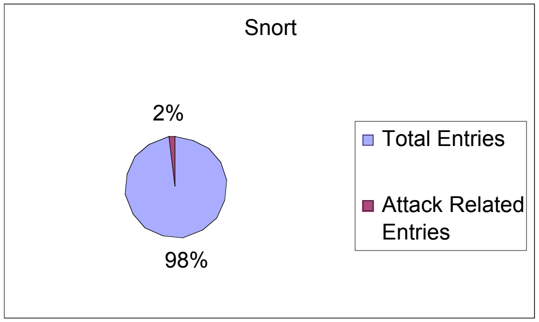

Figure 10: Size of insider traffic log file of Size of Snort

On the other hand, Snort's log files contained a total of 3005 entries. Of these only 66 was attack-related information. We can see that this only amounted to 2% of the total, figure 10.

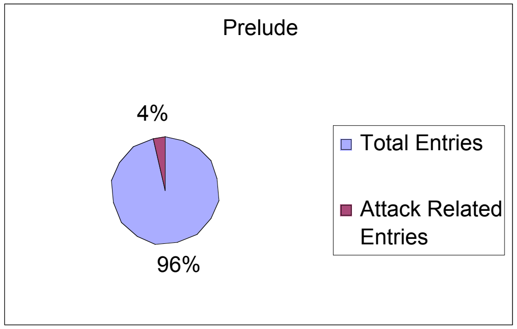

Figure 11: Size of insider traffic log file of Prelude

Prelude had the smallest number of log file entries; in total there were 1671 entries. Of these 61 was attack-related information. This amounted to 4% of the total entries, figure 11.

The results shown above that the problem of false positives reported by Intrusion Detection Systems is still there. The best percentage of four by Prelude shows that a lot of analysis time is still required to go through log files to determine what it is actually an attack. Deeper analysis showed that a direct correlation between the number of signatures employed and the number of entries in the log files of the open source systems. Firestorm employed 1568 signatures, which was its entire rule set of 44. Snort which the second highest number of log file entries used 1390 rules which equated to 36 of the 48 possible rules it stored within its signature database. Prelude with the fewest entries used 890 signatures or 26 of the 33 rules in its signature database. Dragon the commercial system under evaluation used approximately 370 signatures, however its high number of log file entries can most likely be attributed to its combination of being rule and anomaly based system. For each system, signatures written were grouped under different rules, this makes it easier to then turn off certain rules, which may not be applicable to the network.

The outsider traffic logs contained fewer attacks and thus the total number of entries in the log files be smaller. For Dragon and Prelude this occurred and they both represented the systems with the highest and lowest number of entries respectively. Snort and Firestorm were opposite to this however with a higher number of entries in their respective log files.

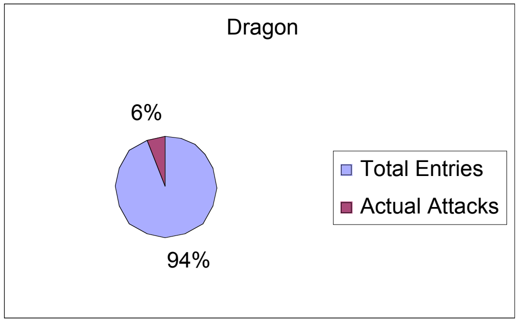

Figure 12: Size of outsider traffic log file of Dragon

Dragon had 49714 entries with 3078 entries being attack-related information. We can see that this attack related entries accounted for 6%, figure 12.

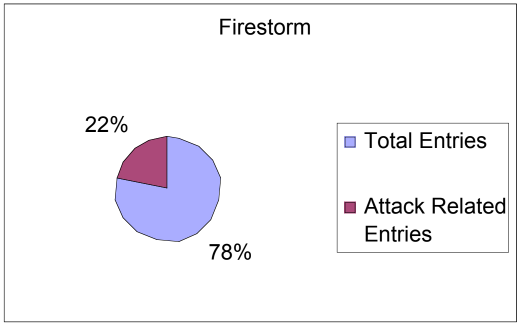

Figure 13: Size of outsider traffic log file of Firestorm

Firestorm once again had the second highest number of entries with 6230 entries with 1771 entries accounting for attack related entries. Figure 13 shows that this attack related information represented 22% of the log file.

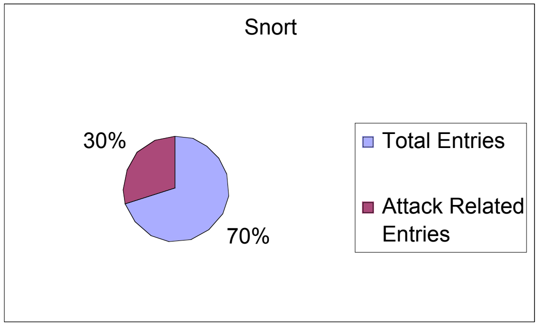

Figure 14: Size of outsider traffic log file of Snort

Moreover, Snort followed with 4087 entries with 1729 entries accounting for attack related entries. This attack related information represented 30%, figure 14.

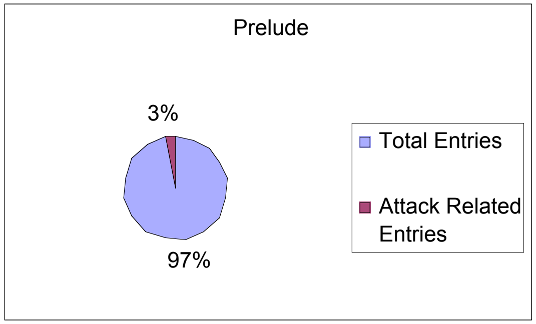

Figure 15: Size of outsider traffic log file of Prelude

Prelude as stated earlier once again had the smallest number of entries. For this system there were 1119 entries with 34 entries accounting for attack related entries, which is only 3% of the log file, figure 15.

Dragon, Firestorm and Snort all showed a higher attack to total entries ratio for the outsider traffic. Deeper investigation revealed that the high number of attack related entries could be matched to a single attack rather than more attacks being caught.

The Confidence with which each Intrusion Detection System scored was also an important metric for analysis. Ideally with a confidence level of 1 an analysis can gain a better understanding of which possible service an attacker is using to attack a machine on the network. Also by having the correct port entry the analysis can also make an assumption as to which service is being attacked. In many situations however a service from a connecting computer may not use an established port to carry out its attack. Therefore, a confidence level of 2 can be used.

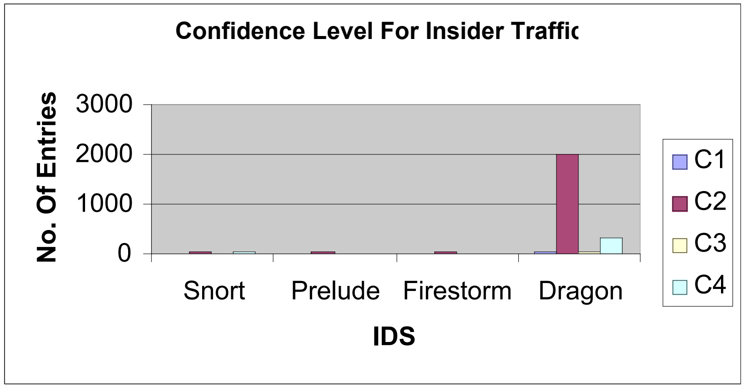

Figure 16: Confidence levels for each IDS for insider traffic

For the insider traffic, figure 16, Dragon's C2 entry makes it difficult because of its large number of entries to see how the other IDS perform. However the data provided by each system did show that each scored the highest with in the C2 category. Therefore an analysis from going through the log files can indeed quickly narrow down the services which are under attack. Additionally for this traffic, Dragon was the only IDS which had C1 entries.

For the outsider traffic, figure 17, we can see the results were slightly different, three of the systems Snort, Firestorm and Dragon all had a high number of C2 entries in relation to the other levels. Although not seen from the figure, data provided also showed that in relation to the other levels Prelude also scored highest in the C2 level.

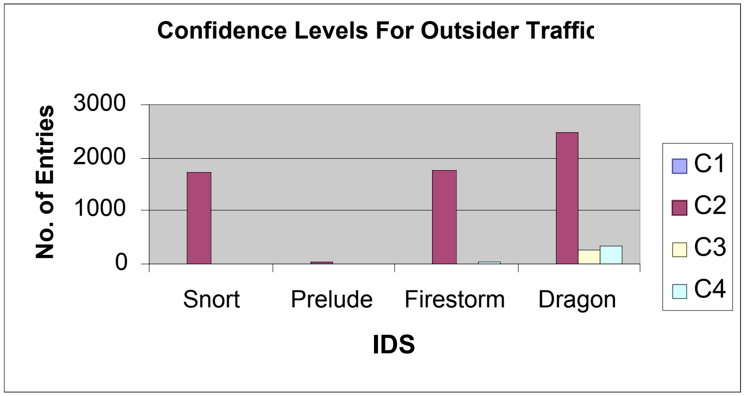

Figure 17: Showing confidence levels for each IDS for outsider traffic

5. Conclusion

From this work it can be seen that Dragon performed better than or scored as high as the three open source systems in four of the five categories for the insider traffic. With the exception of one category it caught at least 50% or more of the attacks in each. In addition many of these attacks were caught with a confidence level of two. Approximately 83% of the attack related entries in the log files were at level 2. Therefore, many of these entries correctly state where the attack is coming and possibly what service is under attack. On the downside Dragon had many false entries as stated earlier therefore it requires a lot of time for analysis to figure out what may be an attack and what is not.

Firestorm was the best performer for the open source systems. In three of the five categories it caught at least 50% of the attacks in each. It however scored very poorly in the Denial of Service Category catching only one of the eight attacks present. 73% of the attack related information in the log entries was at level 2. Thus it was possible to know where the attack was coming from and what service was under attack.

Snort, performed admirably in the DOS category catching more attacks than any of the other systems. Additionally it performed well in the Probe category. In both it caught 50% or more of the attacks present. However, in the other three categories it did not perform very well. Within its attack related log file entry information 54% was at confidence level 2 while 44% was at confidence level 4. Thus, in analyzing the log entries of Snort one can only can only be sure of the attacking IP information with some level of confidence.

Prelude scored the lowest of all the systems under evaluation. In one category it scored 50% and in another two categories it scored 40%. In the denial of service category it scored very lowly catching only two of the eight attacks present. Prelude did however within its attack related information, 75% scored at confidence level 2. Therefore in the analysis of the data it is possible to know where the attack is coming from and the service under attack.

For the outsider traffic Dragon again out performed the open source systems catching 50% or more of the attacks in four of the five categories. It once again did not score well within the probe category. Within its attack related entries 80% was at a confidence level of 2.

Firestorm's performance mirrored that of how it did for the insider traffic scoring 50% or more in three of the five categories. 98% of the information within its attack related information was at confidence level 2. Thus this system performed better on this metric with outsider traffic than it did for insider traffic.

Snort once again performed well in the DOS and Probe categories catching over 50% of the attacks in each but fell down in the other 3 categories. 99% of the information within its attack related information was at confidence level 2. Therefore, this system also performed better on this metric for outsider traffic than for insider traffic.

Prelude for outside traffic once again scored the lowest. Its best ranking came within the DOS category where it caught 50% of the attacks. In the other 4 categories it scored rather poorly ranging from 0% to 40%. 91% of the information within its attack related information was at confidence level 2. Thus this system like the others performed better on this metric for outsider traffic than insider traffic.

Although detecting intrusions at the confidence level of 2 seems very promising, it should be noted that this rating could only be achieved because of the supporting information provided along with the dataset, such as that found in the attack identification list. In a real situation all an analysis will see is the large number of log files from which he or she must try to figure out what constitutes an attack and what does not. All 4 systems as shown earlier have a high incidence of false alarms in their log file entries. For outsider traffic it is slightly better than insider traffic but that is only because an attack was caught which generated a large number of entries. At another level while this attack was caught and the false alarm rate fell it came at the expense of an attack being missed by each system. As stated earlier there was a direct correlation between the number of rules in place and the size of the log files produced. Therefore theoretically by reducing the number of rules in use the size of the log files should be reduced. Each system was used with its default rules in place, it is possible therefore that a system administrator or security expert with a better understanding of the particular network for which the IDS is to be employed can turn on only those rules which are important for that network. This should then help to reduce the size of the log files and in return the number of false alarms. However, it should be noted here that reducing false alarms can also cause a decrease in the detection rate as well.

The most readily seen effects suffered by an e-commerce site by any of these attacks will come from the DOS category. As shown several of these attacks were missed by all of the systems. DOS attacks as explained earlier can make a service unavailable. Although the attacks only resulted in a temporary disruption of service, any time down for an e-commerce can result not only in the lost of potential revenue but dedicated customers as well.

For those organizations not willing to or cannot afford a commercial based system such as Dragon must then make a choice within the range of open source tools available. Overall Firestorm performed better than the other 2 systems under evaluation. However, Snort performed better in the two categories where Firestorm did not. Therefore, it is the decision of the authors that a combination of these two systems would provide the widest coverage for an organization looking to implement an Intrusion Detection System. The coverage of these two systems would match or exceed Dragon's performance. Additionally, when both systems log files are combined their false alarm rate is far less than that of Dragon. Therefore reducing the time an analysis would need to sort through the entries

There are many challenges, which still exists for Intrusion Detection Systems. This work has shown that even when some old attacks are used none of the systems detected them. It also showed that Intrusion Detection Systems could not be at fault when attacks are based on the abuse of perfectly legitimate features. Faults still exist in the way Operating Systems are designed and built.

Many attacks are based upon exploiting these faults. To further compound the problem many applications are built today with security not part of their design format. While it may be possible to write signatures for these vulnerabilities as they are found, many unknown vulnerabilities also exist which are being found daily and exploited by persons with malicious intentions. High False alarm rates continue to be a problem and until these can in some way be reduced Intrusion Detection Systems will continue to receive bad ratings.

This work focused on attacks based at compromising hosts to achieve some desired goal. Further research will investigate the performance of these tools under attacks, which are specifically designed to bring down the intrusion detection systems themselves.

Acknowledgements

The research has been supported by grants from the Natural Sciences and Engineering Research Council of Canada and Dalhousie University Electronic Commerce Executive Committee.

References

[1] Snort. http://www.snort.org. [Last visited: August 29, 2003]
[2] Firestorm. http://www.scaramanga.co.uk/firestorm/. [Last visited: August 29, 2003]
[3] Prelude. http://www.prelude-ids.org/index.php3. [Last visited: August 29, 2003]
[4] Network Intrusion Detection Systems. http://www.networkintrusion.co.uk/N_ids.htm#Dragon. [Last visited: August 29, 2003]
[5] MIT Lincoln Laboratory. http://www.ll.mit.edu/IST/ideval/docs/1999/Network_Topology.gif. [Last visited: August 29, 2003]
[6]Lippmann Richard, Joshua W. Haines, David J. Fried, Jonathan Korba, Kumar Das "The 1999 DARPA Off-Line Intrusion Detection Evaluation", *Draft of paper submitted to Computer Networks*, In Press, 2000.
[7] MIT Lincoln Laboratory. http://www.ll.mit.edu/IST/ideval/docs/1999/attackDB.html#dos. [Last visited: August 29, 2003]
[8] MIT Lincoln Laboratory. http://www.ll.mit.edu/IST/ideval/docs/1999/attackDB.html#r2l. [Last visited: August 29, 2003]

[9] MIT Lincoln Laboratory. http://www.ll.mit.edu/IST/ideval/docs/1999/attackDB.html#probes. [Last visited: August 29, 2003]
[10] MIT Lincoln Laboratory. http://www.ll.mit.edu/IST/ideval/docs/1999/attackDB.html#data. [Last visited: August 29, 2003]
[11] Tcpreplay. http://tcpreplay.sourceforge.net/. [Last visited: August 29, 2003]

Biographies

Christopher Boyce is a student in the Masters of Electronic Commerce at Dalhousie University. He obtained his B.Sc. in Computer Science and Mathematics (Hons.) from The University of The West Indies Cave Hill Campus. His current research deals with Network Intrusion Detection Systems as part of a security in depth for companies engaging in online business.

A. Nur Zincir-Heywood is an Assistant Professor in the Faculty of computer Science at Dalhousie University, NS, Canada. She obtained her B.Sc., M.Sc. and Ph.D. in Computer Engineering from Ege University, Turkey in 1991, 1993 and 1998 respectively. Her research interests include network services and management, network information retrieval, and the effects of the Internet and information technologies on socio –economic development.