7 Test Evaluation

This chapter presents an evaluation of the four Snort configurations. The first section discusses the performance of each configuration. Following this, each Snort configuration is conferred a performance ranking based on their performance which was determined by their detection rates presented in the last chapter. Additionally, this chapter addresses the weaknesses of past evaluations.

7.1 IDS Performance

The performance of each configuration is discussed in the following subsections.

7.1.1 Snort 1.7 Custom

The rules used in this configuration were derived from the ruleset that came with the initial package. The icmp-info rules file was omitted, but all the others were left in tact.

This particular configuration had almost no false positives, but also picked up a minority of the attacks. Basically, the only malicious traffic that it detected was Denial of Service (DoS) attacks that were comprised of malformed packets, such as the Ping of Death (PoD) or Teardrop attacks. It did not detect Probe, Remote to Local (R2L) or User to Root (U2R) attacks so performed very poorly in these particular categories.

This configuration detected 17% of the DoS attacks, 0% of the U2R attacks, 0% of the R2L attacks, 0% of the Probe attacks and 0% of the Data attacks. It obtained a performance ranking of 4 with an overall detection rate of 6%.

7.1.2 Snort 1.7 Full

This configuration performed much better than customized ruleset (Snort 1.7 Custom). The rules file for this particular configuration was the full set of rules from the Snort 1.7 bundle which included the tracking of ICMP traffic. As a result it logged a large portion of ICMP traffic.

This particular configuration was very good at detecting some Denial of Service (DoS) attacks as well as Probe attacks. It generated an enormous amount of false positives, however, since it was monitoring ICMP traffic. In order to detect portscans, ipsweeps, notice FTP probes, and pick up vulnerability scanners, ICMP traffic needs to be captured. However, this particular configuration did poorly at detecting Remote to Local (R2L) attacks than the later versions and rulesets.

This configuration detected 26% of the DoS attacks, 0% of the U2R attacks, 3% of the R2L attacks, 55% of the Probe attacks and 0% of the Data attacks. It obtained a performance ranking of 3 with an overall detection rate of 21%.

7.1.3 Snort 1.8.3 Full

The performance of this configuration did the best, recording the highest overall detection rate. It performed admirably at noticing a variety of Remote to Local (R2L) and Denial of Service (DoS) attacks, although had a reasonably high false positive rate (second to the Snort 1.7 Full configuration). The improvement over the 1.7 runs is readily visible.

This configuration detected 52% of the DoS attacks, 5% of the U2R attacks, 32% of the R2L attacks, 82% of the Probe attacks and 0% of the Data attacks. It obtained a performance ranking of 1 with an overall detection rate of 43%.

7.1.4 Snort 1.8.3 Custom

The performance of this configuration was second best. It detected almost the same amount of attacks as the Snort 1.8.3 Full configuration, with a reduction in false positive rates.

This configuration detected 52% of the DoS attacks, 5% of the U2R attacks, 27% of the R2L attacks, 80% of the Probe attacks and 0% of the Data attacks. It obtained a performance ranking of 2 with an overall detection rate of 41%.

7.2 Performance Ranking

Based on the overall results obtained from the testing, as presented in the previous chapter, the performance ranking for the four configurations are as follows:

Ranking No. 1 – Snort 1.8.3 Full (43% detection rate)

Ranking No. 2 – Snort 1.8.3 Custom (41% detection rate)

Ranking No. 3 – Snort 1.7 Full (21% detection rate)

Ranking No. 4 – Snort 1.7 Custom (6% detection rate)

Snort missed attacks because particular protocols or services were not monitored. For example, Snort missed the "arppoison" attack because the ARP protocol was not monitored. It missed the "snmpget" attack because the SNMP service was not analyzed. Also missed was the "ls" attack because the DNS service was not analyzed. The "selfping" command will not be detected by a network-based intrusion detection system, such as Snort, unless telnet sessions are extracted and analyzed when a "ping" command is issued with specific arguments. Finally, some inside attacks launched from the console of victims and did not generate network traffic so Snort did not detect them because Snort requires the attack be embedded in the network traffic.

7.3 *Limitations of Previous Research*

Based on the experience of constructing a test environment and implementing tests on intrusion detection systems, the limitations of past evaluations are uncovered and highlighted here:

7.3.1 Limitations of the NSS Group 2001 Evaluation

The weakness of these tests is that the background traffic is inconsistent with real life traffic. The traffic was generated using Adtech AX/4000 broadband test system and a Smartbits SMB6000 traffic generator which are designed to test network equipment, not intrusion detection systems. Two problems with this technique is firstly the likelihood of false positives being generated is reduced, if not eliminated and secondly that the actual attacks would differ significantly to the background traffic.

However the relevance of this type of test on a IDS is debatable. In a production environment it is unlikely that a network would be operating close to network saturation for any length of time. If this was the case the network would be redesigned or upgraded. The greatest criticism of this testing process is the lack of testing for false positive alerts.

7.3.2 Limitations of the 1998 DARPA-LL Evaluation

There were a number of problems with the 1998 evaluation. Lincoln Labs cited that the 1998 evaluation was only to provide the exploits against Unix hosts and was only supposed to initially be used for IDS that had been developed using DARPA grants [Lippmann 1999]. Leaving out Windows NT from this evaluation really harms the 1998 evaluation credibility.

One of the major drawbacks in running the evaluation is that a listing of the attacks in the actual test data is not available. Instead, two of the normal "learning data" weeks had to be used. Two biggest problems were the methodology used to come up with the exploits and the way they were executed. There were 38 different kinds of malicious traffic used, but they were executed in no real logical order. Some sort of attacker intelligence should have been placed into the attack routines, even in the preliminary data. Merely adding malicious traffic is not sufficient. Attacks and exploits are sent for a purpose and usually come in a set order. Similar to real-life crime, there is always some amount of reconnaissance before an attack takes place. Even relatively unsophisticated attacks like DoS are usually performed for a purpose and are somewhat calculated. It would have made more sense if the attacks were executed in some kind of logical order.

There were also problems with the IDSs under evaluation. Some of them can only run on super computer clusters while others were extremely complex and not well documented.

7.3.3 Limitations of the 1999 DARPA-LL Evaluation

The major flaws in the system itself deal with the test bed, and how the researchers decided to evaluate the systems. The test bed generates a series of generic packets based on statistical properties and uses a minimum of hosts. Even though the tool used to generate this traffic is not publicly available and the statistical properties used to develop it are unknown, it cannot be as accurate as capturing real-time traffic in a corporate or government network. This helps avoid security implications of publicizing government network traffic, but comes with an inherent cost. It is extremely difficult to

replicate people's actions within a computer program. From an accuracy point of view, it would have been much better to use real traffic.

System type (network-based or host-based) determines how well certain attacks are detected. For example, local-only attacks, where a user misuses software on his/her own computer, could not be detected by network-based IDSs. On the other hand, host-based systems will miss attacks against routers and other computers in most cases. Since the two systems vary heavily in operation and purpose, it would be better to evaluate them separately.

7.4 Summary

This chapter provides evaluations on the four Snort configurations based on their performances. In the first part, the performances of the different Snort configurations were discussed. In the second part, these configurations were conferred their respective performance ranking. A discussion of why certain types of attacks were undetectable was also included. Finally, the weaknesses of previous intrusion detection systems evaluations were addressed.