Journal of The Korean Speleological Society, 2006, 73, 15~19

Network Intrusion Detection System Using SNORT

Wooyoung Soh* · Jason Sigfred**

1. Introduction

There has been a great interest in testing and benchmarking on Intrusion Detection System (IDS) during the past years. Majority of these works, however, were conducted as comparisons, rather than evaluation, on different IDSs. This paper designs and implements a high availability system configuration for network intrusion detection system and benchmarks SNORT version 2.0 as the subject, its performance in such environment and how it can be optimized. This benchmarking is used to gather empirical results in the form of performance data and evidence of both the new system's effectiveness and usefulness.

The objectives in this paper are to design a low-cost high availability system configuration for network intrusion detection system(NIDS) using SNORT to evaluate the system performance to better understand how much system resources is used by SNORT in such a system environment and how SNORT can be affected by "stressed"conditions in the computing environment. One of the main objectives of this work is to design a fault-tolerant NIDS in a two-node environment. This framework comprises functionally redundant systems that provide reliable service despite NIDS failure.

In 1999, the U.S. DARPA initiated the latest evaluation on IDS. It is considered as the most comprehensive scientific study for comparing the performance of different IDS. The network data generated from its private controlled network are used to evaluate IDS. The data is analyzed off-line to determine which sessions are normal and which constitute intrusions. Although this is beyond the scope of the study, more of these attacks can be found on these [2,3]

* Computer Engineering Dept., Graduate School, Hannam University, Korea

conducted.

The most cost-effective approach to increasing the system reliability is to implement a fail-over configuration. Fail-over setup involves pooling together multiple computers, each of which is candidate server for a file systems, databases or applications. Each of these systems monitors the health of other systems in the cluster. In the event of failure in one of the cluster members, the others take over the services of the failed node. The takeover is typically performed in such a way as to make it transparent to the client systems that are accessing the data.

A typical fail-over cluster implementation consists of multiple systems attached to a set of shared storage units, such as disks, connected to a shared SCSI. Each of the cluster members usually monitors the health of others via network (e.g.Ethernet) and/or point-to-point serial connections. And recently, viable Linux-based cluster offerings that run on commodity hardware, such as the Beowulf Cluster, have become available.

To have a failsafe intrusion detection system, Linux High Availability (HA) machine were set-up, using several open source packages available. Virtually every UNIX supplier has their own HA software solution to provide customers " This work was supported by a grant No. R12-2003-004-01002-0 from Korea Science & Engineering Foundation."with fail-over server systems at moderate prices or even free of cost. To come up with such fail-over environment, several HA solutions were used. Such environment is discussed below.

2. Experiments

The experiment has been conductedusing a simulated network and SNORT 2.0. The IDS is evaluated using the captured sample tcpdump traffic from MIT's Lincoln Lab IDS evaluation performed in 1998 [4]. This sample data contains only simple attacks, which are included to illustrate how intrusion detection system will be evaluated. Attacks include instances where a remote user illegally obtains local user-level privileges or local root-level privileges on a target machine and instances where a remote user surveys a potential target for weaknesses or searches for potential targets. Attacks in the sample data include: Guess, ping-sweep, port-scan,

phf, Rlogin, Rsh, Rcp. With the sample test data, there are 272 non-intrusive sessions and 39 intrusive sessions. Another test involve is injecting packet onto a test network on which the subject SNORT was running. On this test, it engages the TCP protocol. In some cases, the tests have some interacting between the injected packets and the third host, representing a hypothetical "target" of attack. In each test, this target host was the exploit addressee of the entire packet injected. In testing or evaluating SNORT, sets of performance objectives were identified first(which are similar to the design goals cited in [5]) including Broad Detection Range, Economy in Resource Usage, Resilience to Stress.

Fail-over Test : Experiments have been performed to ensure the high availability of SNORT. Several of these tests were performed like taking down the primary host, kill its power, and the eth0 heartbeat cable.

Takeover Test : During testing, few scenarios have been deployed to check whether high-availability is attained. One situation was to run the SNORT, get its process, kill it and verify whether the back-up node take over as the sensor. Another condition tested was when the SNORT host was shutdown. The results were all as expected. A crossover cable (eth0) was used for the monitoring of ones "heart beats". In the case of the inter-node/heartbeat network failing, the nodes simply carried on normal operation and do some alerts.

Sensor Node's Performance : One of the benchmarks measured the overall SNORT host performance including CPU average load, memory statistics and I/O transfer rate during the subject IDS SNORT is running with and without the backup node. Average performance of the SNORT host was experimented without and with the backup node. The results show that the CPU utilization of the SNORT sensor node varies when backup node constantly monitors the "heat beat" of the said sensor. And it depicts that the overhead of the host is not that high.

Basic Detection Test : For this test, the output of SNORT that are being logged and saved to the database using the tool ACID from the "baseline" test conducted, were being analyzed. As mentioned earlier that with this test, it is to ensure that SNORT is properly configured with its plug-ins, particularly its preprocessors and rules, and to check if SNORT reacted to the test by

either reporting or not reporting the sidestep attacks. By considering the SNORT's output and the specific attack packets used for the test, it was able to deduce significant characteristics of the IDS.

Resource Usage Test : The SNORT host's resource usage was tested. During the actual test using the 1998 DARPA's sample data, the total disk space used by SNORT's output through ACID was measured by using the UNIX sar (system activity reporter) command. The measurement have been made during the running of sample test data from 1998 DARPA IDS evaluation. With this report, total number of transfers per second that were issued to the physical disk was collected. Other experiments conducted include the total number of read requests per second issued to the physical disk, as well as the write requests per second issued to the physical disk. Comparing the result in the first environment where the backup node was disabled

Stress Test : CPU run queue of SNORT host was measured in such stressful conditions. It is hypothesized that such stress in the form of high load on the SNORT host might affect its ability tomonitor network connections. Aside from this, memory activities and swap space utilization statistics were also measured. Values of such statistics were taken from sar tool. First the amount of free memory available, the amount of used memory, percentage of used memory, the amount of free swap space and several other values were tested during the experiments.

3. Conclusions

This paper presented a network environment of a fail-over network intrusion detection system in a high availability machine. The results from the study show the viability of such approach for redundant IDS. This paper also presented an inexpensive high-availability solution design for the mission-critical needs without requiring the use of expensive additional hardware or software.

On the whole, this two-node system configuration environment provides with much better system availability and reliability, especially on security systems. It is a great improvement over the single node, as it is affordable to do server maintenance.

Futuredirections of this research would include the design and implementation of a

multi resource high-availability environment such as redundant database for the IDS alert logs, file system takeover and others are also planned to be undertaken as future research. Another future study would be the design of a multi nodes environment, a kind of a parallel computing environment where each node performs different tasks. Because intrusion detection systems spend most of CPU cycles reading incoming packets, distinguishing what they are and matching them to some rules, parallel computing would be a suitable environment by assigning each node with these jobs.

The empirical results gathered from this study and the aforementioned intentions to make such processing environment, enhancements on the effectiveness and usefulness of security systems in general and intrusion detection systems in particular will surface. A more reliable and stable setting, despite security system's failure, hopefully will be developed.

References

[1] SNORT's Documentation, URL: http://www.SNORT.org

[2] K. Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems", M.Eng. Paper, MIT Department of Electrical Engineering and Computer Science, June 1999.

[3] K. J. Das, "Attack Development for Intrusion Detection Evaluation", M.Eng. Paper, MIT Department of Electrical Engineering and Computer Science, June 2000.

[4] Richard Lippman, et. al., "The 1999 DARPA Off-Line Intrusion Detection Evaluation", submitted to Proceedings of 3rd International Workshop on Recent Advances in Intrusion Detection (RAID 2000).

[5] N. Puketza, et. al., "A Methodology for Testing Intrusion Detection System", Proc. 17th National Computer Security Conference, October 1994.