A CASE STUDY OF THREE OPEN SOURCE SECURITY MANAGEMENT TOOLS

Hilmi Gunes Kayacik, A. Nur Zincir-Heywood

kayacik@cs.dal.ca, zincir@cs.dal.ca
Dalhousie University, Faculty of Computer Science, Canada

Abstract: Three open source security management tools – *Snort, Pakemon, and Argus* – are benchmarked against DARPA 1999 Intrusion Detection Evaluation Data Set. Performance is characterized using multiple performance metrics. *Snort* is found to have the best performance in terms of detection rate, however it creates more false positives than desired. The results show that different tools perform well under different attack categories; hence they can be run at the same time to increase the detection rate of attack instances.

Key words: Security management, Case Study, Open Source Software, IDS

1. INTRODUCTION

Security management plays an important role in today's network management tasks. Defensive information operations, and intrusion detection systems are primarily designed to protect the availability, confidentiality and integrity of critical network information systems [3]. The automated detection and immediate reporting of these events are required in order to provide a timely response to attacks [2]. A balance therefore exists between the use of resources and the accuracy and timeliness of intrusion detection information. Since most of the commercial intrusion detection systems are at typically thousands of dollars and they tend to represent a significant resource requirement in themselves, for small networks, use of such IDS is not feasible. The objective of this work is therefore to evaluate three open source security management tools in order to understand which one of them will be more useful for network intrusion detection. To achieve this, we have chosen

Snort, *Pakemon* and *Argus*, since they are three of the most popular open source tools [1, 5, 6]. “Pakemon has been developed to share IDS components based on the open source model” [6]. It is an experimental IDS, which aims to detect evasion methods such as fragmentation, disorder, duplication, overlap, insertion, and de-synchronization at the IP or TCP layer. Pakemons's signature structure is simpler than other IDS (such as S*nort*), where this simplicity is both a strength, and a weakness. That is to say, it takes time for IDS organizations to release new signature files. Meanwhile, as the signatures of new attacks are revealed, it is much easier to add them to the lightweight IDS signature databases such as Pakemon [6]. Snort is one of the best-known lightweight IDSs, which focuses on performance, flexibility and simplicity [5]. It is an example of active intrusion detection systems that detects possible intrusions or access violations while they are occurring. Although not as straightforward as the Pakemon system, flexible rule writing is supported in Snort. In contrast to Pakemon and Snort, *Argus* is not an IDS but it is an open source general network management tool [1]. This means that Argus monitors and inspects network traffic and connections both for attempted connections and established connections. In other words, it is a specific IP auditing tool, hence in the case of this work, it is used to analyze the superset of the traffic logged by the other two intrusion detection systems.

2. TEST SET UP AND PROCEDURES

The test setup of this work consists of the following components: DARPA 1999 data set, traffic re-player, and the three open source systems. The data set [4] represents TCP dump data generated over five weeks of simulated network traffic in a hypothetical military local area network (LAN). This data was processed into some 7 million TCP connection records. Our work concentrates on the internal and external traffic collected by the sniffers. In this case for, reasons of expediency, we concentrate on the 2.5 GB of data present in the week 4 data set (week 5 is even larger and beyond the computing resources available). The data used for testing therefore either represented a normal connection or one of the 80 attacks [4]. This work employed one machine as the IDS server and another machine to replay the network traffic using *TCPReplay* [7]. Moreover, all the software used are installed and configured using their default values, and the latest signature files available (February 2002) are used for *Pakemon* and *Snort*. It should be noted that log files of the tools that are evaluated contain different types of entries including different amounts of information about the events that occurred on the network. Therefore, 4 confidence levels are defined for determining the degree of match in order to detect different attacks, table-1.

3. RESULTS

Out of total number of 80 attack instances, *Snort* detected 35 and *Pakemon* detected 27 in total. Indeed, it should be noted that even if we had an intelligent way

to mine *Argus* log file, we could have only detected 70 attacks out of the 80 present in the test data set. To actually determine which tool performs better, two other parameters are analyzed: (1) the number of false alarms and (2) the number of entries that it takes to be parsed by a network administrator to detect those attacks. Figure 1 shows the number of attack related entries over the total number of entries in the corresponding log files. Thus, in both cases it is costly to examine all log files. On the other hand, when the attacks detected by *Snort* and *Pakemon* are examined more closely, a strong commonality exists between the types of attacks detected. As it can be seen in figure 2, Snort on its own is much better than *Pakemon*, however if they work together Their performance increases by approximately 20%. For both of them though, the confidence level of detection is mostly at level-3, figure 3.

	Source IP	**Destination IP**	**Source Port**	**Destination Port**
Level1	X	X	X	X
Level2	X	X		X
Level3	X	X		
Level4	X			

Table 1: Summary of the confidence levels (X indicates the match required)

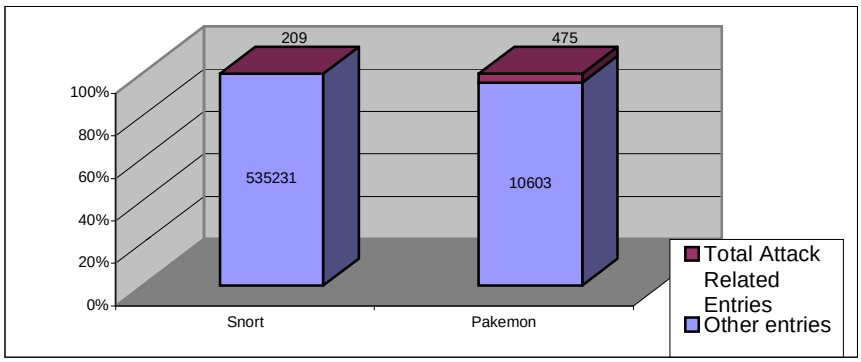

Figure 1. Number of attack related entries in the corresponding log files

4. CONCLUSION

The work presented is a case study, but we believe sufficient to warrant continued development. In particular, we have demonstrated a benchmark evaluation of popular open source security management tools. The results show that none of the tools could capture all the different attack instances: Snort captured ~44% and Pakemon ~34%. Moreover, Snort has ~99% false alarms whereas Pakemon has ~95%. In other words all three generate very large log files, which in return makes it difficult to analyze for network managers. Therefore, it is important to develop filters for these tools to decrease the number of false alarms. Furthermore, we believe that different tools need to be used together to increase the detection rate.

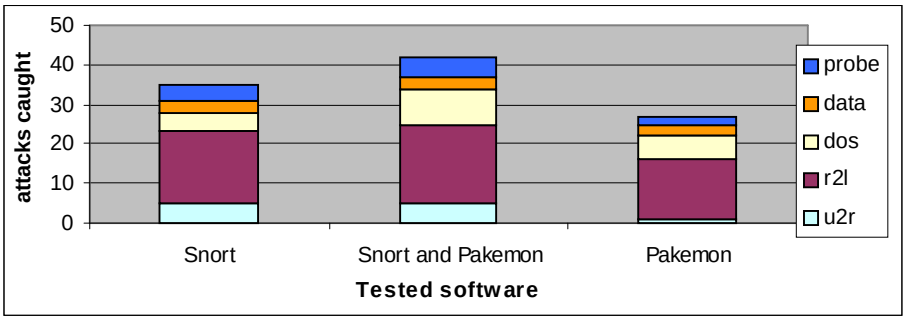

Figure 2: The distribution of attacks that are caught by *Snort* and *Pakemon*

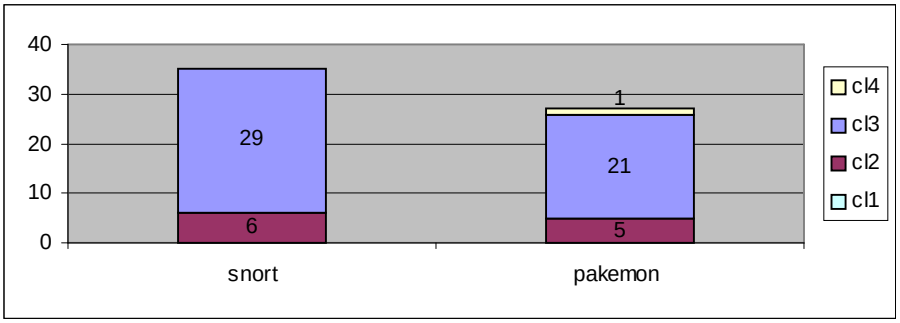

Figure 3: Number of attacks and their corresponding confidence levels for each tool

ACKNOWLEDEGEMENTS

The authors gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[1] Argus, http://www.qosient.com/argus
[2] Bass T., “Intrusion Detection Systems and Multisensor Data Fusion”, Communications of the ACM, Vol. 43, No. 4, pp 99-105, April, 2000.
[3] Kayacik G., Zincir-Heywood A. N., “Evaluation of the Cisco IOS Firewall with Darpa 99 Dataset”, Technical Report, Faculty of Computer Science, Dalhousie University, http://www.cs.dal.ca/~kayacik/download/report.pdf, November 2002
[4] MIT Lincoln Laboratory, http://www.ll.mit.edu/IST/ideval/data/data_index.html
[5] Snort, http://www.snort.org
[6] Takeda K., Takefuji Y., “Pakemon – A Rule Based Network Intrusion Detection System”, International Journal of Knowledge-Based Intelligent Engineering Systems, Vol. 5, No. 4, pp 240-246, October 2001.
[7] TCPReplay traffic replay utility, http://tcpreplay.sourceforge.net/