Proceedings of the 2011 10th IEEE International Conference
On Cybernetic Intelligent Systems, September 1-2, London, UK

Time Series Analyses for forecasting Network Intrusions

Joshua Ojo Nehinbe
University of Essex, Colchester, UK
jnehin@essex.ac.uk

Abstract-Intrusion Detection Systems are fast-growing techniques for monitoring and garnering electronic evidences about suspicious activities that signify threats to computer systems. Generally, these mechanisms overwhelmingly describe and record patterns of suspicious packets as alerts in the form of intrusion logs. Thereafter, analysts must subsequently validate the content of each intrusion log to ascertain the validity of each alert. Secondly, high level of expertise is required to discern each alert. However, more time and resources are unduly spent at the expense of countermeasures that ought to be proactively initiated to thwart attacks in progress. Accordingly, TSA-Log analyzer that uses a computationally fast technique and a uniform baseline to determine patterns of intrusions is proposed in this paper. Validations that are carried out on five publicly available datasets demonstrate that propagation strategies of intrusions, efficient countermeasures and the extent of similarity of intrusions can be forecasted giving the knowledge of the patterns of alerts in intrusion logs.

Key word- ***Alert; timestamp; time series analysis; intrusion; patterns of intrusions; intrusion detection system***

I. INTRODUCTION

From the time Denning (1987) published his paper on Intrusion Detection Expert System (IDES), there are increasing cases of unlawful accesses into computer systems, otherwise known as intrusions, in several quarters of the globe (Lazarevic et al, 2005). For these reasons, Intrusion Detection Systems (IDSs) are fast-growing mechanisms for monitoring, gathering and reporting electronic evidence of activities that depict threats or intrusions to computers and computer resources (Debar and Wespi, 2001; Scarfone and Mell, 2007).

Fundamentally, in Rehman (2003), each Intrusion Detection System (IDS) can extract a wide range of attributes from a suspicious packet and the host machine where the IDS is installed, to comprehensively describe the packet. Usually, typical IDS will trigger alerts whenever it detects any suspicious activity. The alerts can be logged to form intrusion logs for analysts to conduct subsequent review about the events so that the validity of the reports and the nature of the alerts can be ascertained. It is after these have been done accurately that efficient countermeasures that are capable of thwarting attacks in progress can be carried out. However, how to validate alerts of IDS and how to establish the nature of each alert are serious challenges over the years for a number of reasons (Scarfone and Mell, 2007). Firstly, IDS generally generates overwhelming warnings that contain embedded records of patterns of suspicious packets in order to maximize intrusion detection. However, high level of expertise is needed to understand each alert and to select useful baselines for extracting patterns of attacks in each intrusion log.

The validity of alerts in intrusions cannot be unveiled by manual analyses of most intrusion logs. Consequently, series of methods have been proposed over the years. However, automated methods for reviewing intrusion logs that use clustering and alerts aggregations are often criticized because they have the capabilities to suppress patterns of attacks across different intrusion logs (Alder et al, 2007). These have necessitated the developments of some third party applications such as Barnyard, Analysis Console for Intrusion Database (ACID), Sawtch, Open Source Security Information Management (OSSIM), Simple Event Correlator (SEC) and the THC's Netdude packet translation (Alder et al, 2007). However, most of the existing automated toolkits are limited to certain IP protocols. Taking the THC's Netdude trace management and ACID for instance, the toolkit will also modify the packets from within the interface (Alder et al, 2007). Besides writing sufficient correlation rules for categorizing all alerts in very large intrusion logs is a major argument against automated tools such as the SEC that uses correlation rules to reformat alerts of Snort (Alder et al, 2007). Importantly, most of these tools are not generally adaptable for processing alerts of other IDSs. In essence, analysts must reformat and normalized the input logs with other programs before most of the existing tools can process the intrusion logs to a certain degree. Accordingly, more time and resources are unduly spent at the expense of countermeasures that ought to be proactively initiated.

Accordingly, we present TSA-Log analyzer to reduce the aforementioned problems. TSA-Log analyzer uses the concept of time series analysis to extract comparable patterns of intrusions in intrusion logs. In so doing, our method uses a computationally fast technique to extract patterns of intrusions from intrusion logs using a predefined baseline. TSA-Log analyzer is implemented with C++ and the model is validated with series of intrusion logs triggered by Snort in intrusion detection mode.

One of the substantial contributions of this paper is that we have been able to demonstrate that analysts can forecast the extent of similarity of attacks giving the knowledge of the patterns of alerts in a set of intrusion logs. This paper has also shown the usefulness of timestamp for understanding propagation strategies of attacks. Capability of the TSA-Log analyzer for designing efficient countermeasures that are capable of complementing measures for concurrently thwarting multiple attacks is another contribution of this paper. The remainders of this paper are organized as follows. Section 2 will discuss related works. Section 3 will discuss the concept of time series analysis of intrusion logs. Section 4 will discuss the evaluative datasets and the method proposed in this paper. Section 5 will discuss the results of various experiments we conduct with the proposed model. Section 6 will conclude the paper and offers suggestions on future research work that can be explored to improve the limitations of this paper.

II. RELATED WORKS

There are quite a few researches that actually use time series analysis to extract patterns of attacks in intrusion logs in a recent time. For instance, Viinikka et al (2006) used time series to isolate patterns of alerts of packets that deviated from normal behaviour using a set of background noises. The scholars built a database of normal behaviours in the form of filtering rules and each alert was validated with each of these rules to detect deviations. Nonetheless, since the concept of normal behaviours may continuously change with time, security policy of an organisation and different networks, in the field IDS, the efficacy of this model depends on the ability of a human operator to select suitable normal behaviours.

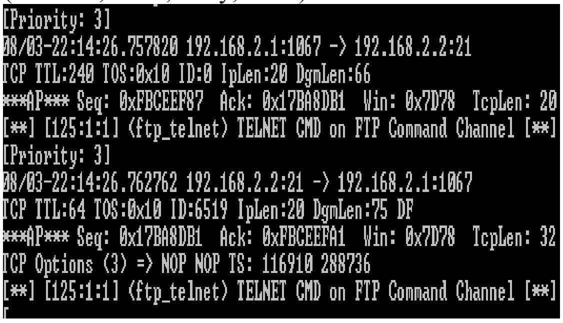
Qin and Le (2003) used time series analysis and clustering techniques to correlate events that have similar attack scenarios. Although, this model may visually demonstrate casual relationships among sources of attacks and their respective destination addresses, however, the model was mainly on the reduction of overwhelming alerts triggered by IDSs. In addition, in Yusof (2008**),** the work of Qin and Le (2003) was criticized for its inability to completely correlate many intrusion logs. Hence, visual analysis of alerts that the model can establish is limited to small datasets.

In Ren et al (2005), sequence of records that combined addresses with time to form a time series model was proposed. According to the authors, specific streams of alerts can be eliminated giving the knowledge of the unsuccessful connections attacks made within a given the time window. The method visually displayed attacks that have similar addresses using attacking keys. However, the validity of the results with attacks from a broad range of evaluative datasets was not established.

To the best of knowledge, most of the existing models lack the capabilities to be adapted for comparing patterns of attacks in different segments of computer networks.

III. TIME SERIES ANALYSIS ALERTS

Trends are usually embedded within the manner at which packets migrate from one computer to another especially if the communication is continuous over time and this is the basis of time series or temporal analysis of intrusion logs (Brebner, 1997; Shay, 2004).



```
[Priority: 3]
08/03-22:14:26.757820 192.168.2.1:1067 -> 192.168.2.2:21
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:66
***AP*** Seq: 0xFBCEEF87  Ack: 0x17BA8DB1  Win: 0x7D78  TcpLen: 20
[**] [125:1:1] (ftp_telnet) TELNET CMD on FTP Command Channel [**]
[Priority: 3]
08/03-22:14:26.762762 192.168.2.2:21 -> 192.168.2.1:1067
TCP TTL:64 TOS:0x10 ID:6519 IpLen:20 DgmLen:75 DF
***AP*** Seq: 0x17BA8DB1  Ack: 0xFBCEEFA1  Win: 0x7D78  TcpLen: 32
TCP Options (3) => NOP NOP TS: 116910 288736
[**] [125:1:1] (ftp_telnet) TELNET CMD on FTP Command Channel [**]
[
```

Figure 1: Snort's alert of a public dataset

In other words, a time series of an intrusion log is a measure of alerts as they are triggered by IDS within uniform time intervals.

Experiences have shown that the IDSs such as Bro and Snort concatenate information about the year, month, date, hour, minute and second that a suspicious packet is detected to form the timestamp of each alert. In Rehman (2003), the timestamp of an alert signifies the date and time when and IDS triggers an alert. Two examples of alerts triggered by Snort in IDS mode are shown in Figure 1 to prove that every alert has a timestamp.

The extent of the usability of the entire string that forms a timestamp of each alert for reducing alerts workload especially if the attacks are not bunched together has been discussed in Nehinbe (2011). In this paper, the premise is that instead of completely discarding the timestamp, it is plausible to explore some hidden patterns and regularities in intrusion logs using some fields extracted from the timestamp of each alert. Thus, this paper uses a set of predefined time interval that range from one second through sixty seconds to determine patterns of alerts in a set of intrusion logs. The premise is that by so doing, analysts will be able to make objective comparison of the patterns of attacks across different segments of computer networks.

A. USES OF PATTERNS OF INTRUSIONS

Han and Kamber (2006) and McCabe (2003) assert that temporal analyses can be used to examine the availability of resources, resource delay and resource utilization in computer networks.

Time series analysis of intrusion logs gives a pictorial perspective of the nature or patterns of attacks, how alerts or attacks change in relation to other alerts or attacks within the same intervals and the extent of redundant alerts (regularities) and irregularities (uniqueness of alerts) in a set of intrusion logs.

Furthermore, analysts can use temporal analyses of intrusion logs to design efficient countermeasures that are capable of thwarting multiple attacks in progress. For instance, patterns of attacks that show repeated events are indications of how to avoid redundant countermeasures.

Additionally, forecasting of countermeasures, and forecasting of alerts workload are other uses of time series analyses of intrusion logs.

B. CHALLENGES IN FINDING PATTERNS OF INTRUSIONS

There are numerous challenges that are associated with the usage of time series analysis to gain insightful knowledge about hidden patterns of network intrusions.

In Glass et al (2008), intruders can perform varieties of operations within computer networks because the motives of intruders may suddenly change as the targets are compromised.

In Karen and Scarfone (2007) and in Nehinbe (2011), intruders can carry out varieties of nefarious activities against target machines. For examples, intruders can launch the DDoS attacks, network scanning attacks, buffer overflow attacks, phishing attacks, computer viruses, worms, Trojans, spyware or any other forms of attacks against the target systems. Some intruders may illegally delete, replicate, corrupt and personalize the data already stored in the compromised computer systems by restricting legitimate accesses to such resources if the attacks are left to succeed. Similarly, some intruders may compromise just some segments of the target systems. Furthermore, some stealthy intruders may launch probes that may maliciously modify the configurations of the compromised machines. Essentially, giving the variability of operations that attackers can perform in compromised computer systems, thus, a fundamental research issue here is how to preempt patterns of these probes. This is because different probes may correspond to different nefarious activities. In other words, each of these operations may constitute different patterns of attacks.

Attacking sessions can vary from one intrusion log to another. For instance, the time zones, date and location where the datasets are extracted are often different from each other. In most of the publicly available datasets that are designed for investigating IDS related issues, for instance, one of the DAPRA 2000 datasets was collected on the 7th of March, 2000 while the second scenario of the same dataset was collected on the 16th April, 2000 (DAPRA, 2011). Similarly, the DEFCON-10 dataset was collected on the 3rd of August, 2002 (CTFC, 2011).

From the foregoing analyses, it is difficult to use entire strings that form the timestamp of alerts to extract patterns of attacks or alerts across different intrusion logs. Consequently, selection of a suitable temporal attribute of alerts is a central problem in detecting patterns of attacks in the field of IDSs.

IV. THE TSA-LOG ANALYZER

We use Snort to generate input alerts that are used for verifying the ideas proposed in this paper. Snort is a packet sniffer and it is commonly used for conducting most of the IDS researches (Alder et al, 2007; Rehman, 2003),

Some local rules are added to the default rules of Snort to improve the detection capability of the detector. Thereafter, Snort is operated in an intrusion detection mode to sniff each of the evaluative datasets in an offline mode.

A. DISCUSSION OF DATASETS

Five publicly available datasets that are used for evaluating TSA-Log analyzer proposed in this paper extracted in packet capture (PCP) formats. The datasets are subsequently preprocessed into Snort's readable formats.

The LLDDoS 1.0 and LLDDoS 2.1.0 are two of the datasets we use. They are extracted from the repositories maintained by the Lincoln Laboratory at the Massachusetts Institute of Technology, United States (DAPRA, 2011). Both datasets are examples of Distributed Denial of Service (DDoS) attacks. The LLDDoS 1.0 dataset was DDoS attacks launched by novice attacker while the LLDDoS 2.1.0 dataset was DDoS attacks launched by experienced attacker (DAPRA, 2011).

Moreover, we obtain three datasets from the repositories maintained by the Shmoo Group. We respectively use the DEFCON-8, DEFCON-10 and

DEFCON-11 datasets respectively (DEFCON, 2011). Some of the attacks in the DEFCON-8 dataset subsume the fragmented packet attacks, ports scan attacks and buffer overflow attacks.The DEFCON-10 dataset contains attacks on administrative privilege, FTP attacks, ports scan and port sweeps attacks, fragmented attacks that intended to cause buffer-overflow, directory traversal attacks and IP spoofing attacks. The DEFCON-11 dataset contain attacks such as attacks that targeted the administrative privileges of legitimate users, hijacked session logins, attacks that probed the networks, fragmented and crafted packets.

The LLDDoS 1.0 dataset generates 834 alerts while the LLDDoS 2.1.0 generates 816 alerts. Also, the DEFCON-8 dataset generates 909,648 alerts; the DEFCON-10 dataset generates 5,372 alerts while the DEFCON-11 dataset generates 8,510 alerts. The alerts of each dataset subsequently formed input data for the TSA-Log analyzer.

B. IMPLEMENTATION OF THE TSA-LOG ANALYZER

We implement the concept of the TSA-Log analyzer with C++ programming language. The program runs on a windows XP desktop machine with a configuration of Intel® Core(TM) 2 CPU, 2.40GHZ and 3.50 RAM as shown in Figure 2.

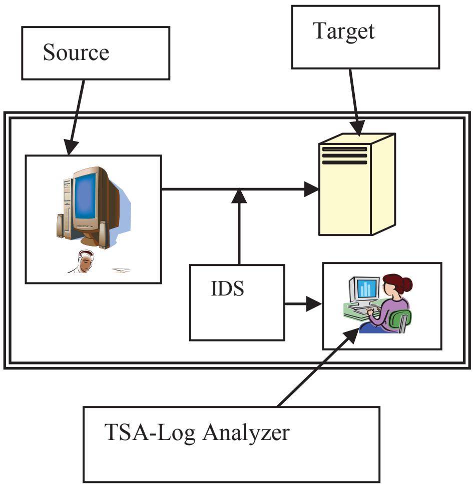
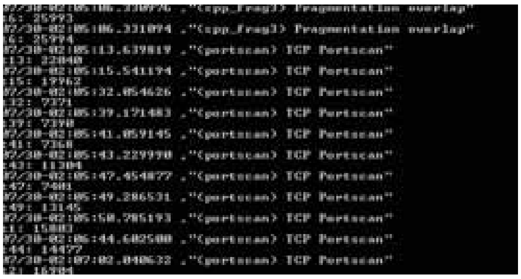


Figure 2: TSA-Log analyzer

The method uses by the TSA-Log analyzer to extract patterns of attacks in each intrusion log is divided into three phases. We refer to them as the *processing phase,* the *pattern mining phase* and the *validation phase*. Importantly, at least one of the attributes of the input alerts must contain timestamp. Thus, in the processing phase, the program automatically receives raw alerts from the input dataset passes all the alerts to the subsequent phase.

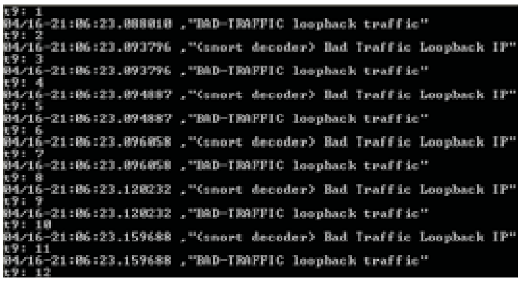
In the pattern mining phase, the program searches for the field that held the value for ***seconds*** in each alert. This field is computationally matched against a uniform baseline in single passes to determine patterns of intrusions in the input dataset.



```
07/30-02:05:06.330976 ,"(spp_frag3) Fragmentation overlap"
6: 25993
07/30-02:05:06.331094 ,"(spp_frag3) Fragmentation overlap"
6: 25994
07/30-02:05:13.639819 ,"(portscan) TCP Portscan"
13: 22040
07/30-02:05:15.541194 ,"(portscan) TCP Portscan"
15: 19962
07/30-02:05:32.054626 ,"(portscan) TCP Portscan"
32: 7371
07/30-02:05:39.171483 ,"(portscan) TCP Portscan"
39: 7370
07/30-02:05:41.059145 ,"(portscan) TCP Portscan"
41: 7368
07/30-02:05:43.229990 ,"(portscan) TCP Portscan"
43: 11304
07/30-02:05:47.454077 ,"(portscan) TCP Portscan"
47: 7401
07/30-02:05:49.286531 ,"(portscan) TCP Portscan"
49: 13145
07/30-02:05:50.785193 ,"(portscan) TCP Portscan"
1: 15003
07/30-02:06:44.682500 ,"(portscan) TCP Portscan"
44: 14477
07/30-02:07:02.040632 ,"(portscan) TCP Portscan"
2: 16704
```

Figure 3: DEFCON-8 datasets

Along the line, this process categorizes all alerts into intervals within the baseline. The baseline has sixty intervals starting from one to sixty to indicate the occurrences of attacks monitored within a range of one second to sixty seconds respectively.



```
t9: 1
04/16-21:06:23.000010 ,"BAD-TRAFFIC loopback traffic"
t9: 2
04/16-21:06:23.093796 ,"(snort decoder) Bad Traffic Loopback IP"
t9: 3
04/16-21:06:23.093796 ,"BAD-TRAFFIC loopback traffic"
t9: 4
04/16-21:06:23.094887 ,"(snort decoder) Bad Traffic Loopback IP"
t9: 5
04/16-21:06:23.094887 ,"BAD-TRAFFIC loopback traffic"
t9: 6
04/16-21:06:23.096058 ,"(snort decoder) Bad Traffic Loopback IP"
t9: 7
04/16-21:06:23.096058 ,"BAD-TRAFFIC loopback traffic"
t9: 8
04/16-21:06:23.120232 ,"(snort decoder) Bad Traffic Loopback IP"
t9: 9
04/16-21:06:23.120232 ,"BAD-TRAFFIC loopback traffic"
t9: 10
04/16-21:06:23.159688 ,"(snort decoder) Bad Traffic Loopback IP"
t9: 11
04/16-21:06:23.159688 ,"BAD-TRAFFIC loopback traffic"
t9: 12
```

Figure 4: LLDDoS 2.0.2 dataset

In addition, in the validation phase, the program uses a set of validation routines to ensure that an alert does not belong to more than one interval. The validation routines also ensure that all alerts are processed.

Some of the stages in the processing of the datasets are shown in Figures 3 and 4. The results obtained on each dataset are tabulated and are discussed in a subsequent section

V. RESULTS AND DISCUSSIONS

We perform five experiments with the datasets. The results obtained show that there are different patterns of attacks in each of the evaluative datasets.

In Figure 5 for instance, the patterns of attacks in the DEFCON-10 dataset indicate continuous attacks. The attacks were steady at the beginning of the attacks. Instances whereby the attacks suddenly increased and subsequently decreased were also observed in the dataset.

Similarly, like DEFCON-10, the patterns of attacks in the DEFCON-11 dataset as shown in Figure 6 indicate continuous attacks. However, the attacks are likely to be different from the attacks that constitute the DEFCON-10 dataset because there are feasible differences in the patterns of the attacks that constitute both datasets.

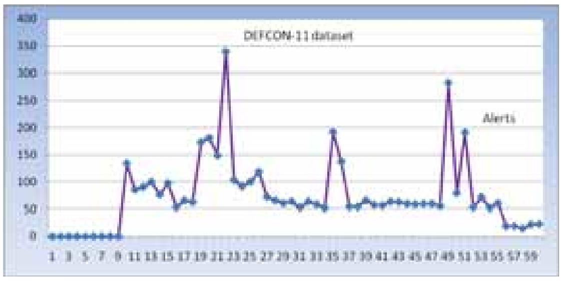


Figure 5 Patterns of attacks in the DEFCON-10 dataset

The pattern of attacks in the LLDDoS 1.0 is shown in Figure 7. The results suggest repeated attacking pattern before the attacks later overflow the target networks.

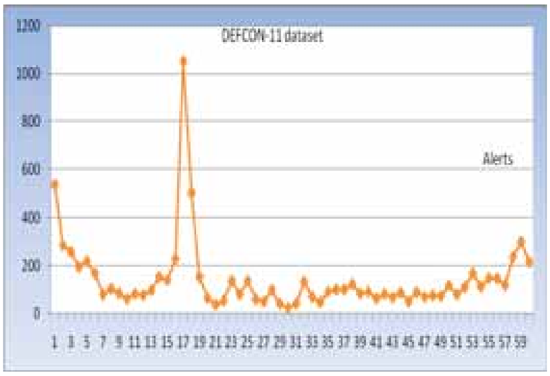


Figure 6 Patterns of attacks in the DEFCON-11 dataset

Like LLDDoS 1.0 dataset, the patterns of attacks in the LLDDoS 2.1.0 dataset as shown in Figure 8, suggest another mode of flooding attacks.

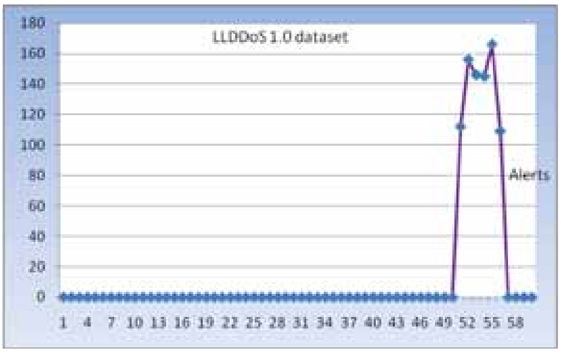


Figure 7 Patterns of attacks in the LLDDoS 1.0 dataset

The results suggest two different periods whereby the attackers repeatedly flooded the victims and an interval where the quantities of attacks suddenly increased. The pattern of attacks in the DEFCON-8 dataset as shown in Figure 9 is different from the patterns of attacks in the previous datasets. The results show instances where the attackers repeatedly launch similar attacks against the host machines.

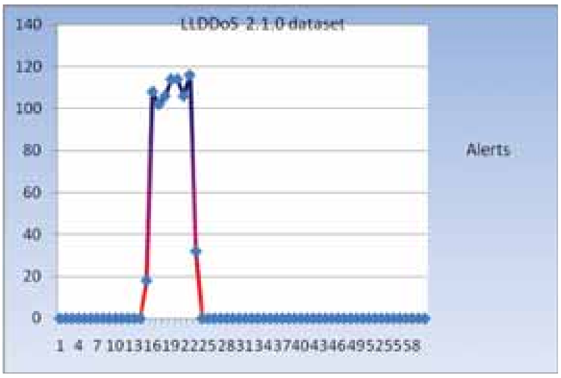


Figure 8 Patterns of attacks in the LLDDoS 2.1.0 dataset

Essentially, the results shown by the TSA-Log analyzer do not feasibly demonstrate that the attacks in all the datasets occurred together. Rather, in Figure 7 and 8, the results show that the attacks in both datasets appear to have occurred in a reverse order.

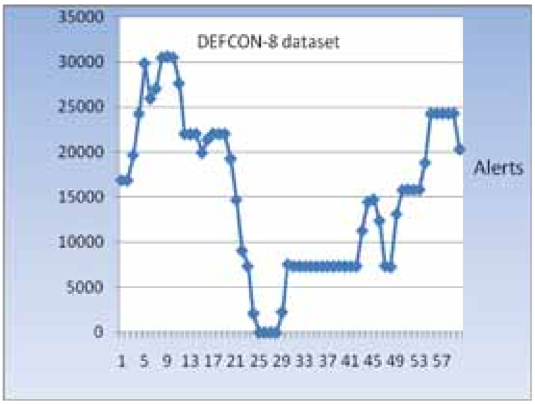


Figure 9 Patterns of attacks in the DEFCON-8 dataset

Furthermore, the above analyses substantiate the existence of patterns of attacks and possible discrepancies in the patterns of attacks from one computer network to another. The results shown by the DEFCON-8, LLDDoS 2.1.0 and LLDDoS 1.0 datasets further demonstrate that it is possible to implement efficient countermeasures to thwart most of the attacks in the datasets whenever they exhibit repeated attacking patterns or strategies.

VI. CONCLUSIONS

This paper demonstrates that the present day IDSs still need further enhancements in terms of intrusion formatting and logging of suspicious packets. We also show that series of methods for augmenting intrusion logs so that analysts can achieve prompt countermeasures are inadequate. Consequently, we propose TSA-Log analyzer for detecting patterns of attacks in a set of intrusion logs without the need for a human operator to normalize or format the input alerts with any external program.

The results show that hidden trends are embedded in the migrations of suspicious packets from their sources to their respective target destinations. So, by mere looking at the patterns obtained, analysts can make strong decisions about the similarity of attacks that IDSs have recorded in a collection of intrusion logs. In other words, analysts would be able to preempt countermeasures that are necessary to thwart attacks in the intrusion logs under review. In essence, our model is very useful for comparing patterns of intrusions from different segments of computer networks together.

Nevertheless, we have not explored the patterns associated with failed attacks and attacks that can achieve the objectives of the intruders in this paper. One method of achieving this is to isolate failed attacks from other category of attacks and then validate each category with TSA-Log analyzer to ascertain the patterns of attacks in each subset.

The experiments described in this paper may erroneously mismatch patterns of true positives with false positives in a set of intrusion logs. All these are potential research areas that call for future investigations.

Finally, the TSA-Log analyzer proposed in this paper may not be sufficient enough for a human operator to thwart all network intrusions at once. Hence, collaborative usages of IDS models are strongly recommended.

REFERENCES

[1] A. Lazarevic, J. Srivastava and V. Kumar, “Intrusion detection: A survey”, Computer Science Department, University of Minnesota (2005).

[2] Alder, R., Baker, A.R., Carter, E.F., Esler, J., Foster, J.C., Jonkman, M., Keefer, C., Marty, R. and Seagren, E.S. Snort: IDS and IPS Toolkit, Syngress publishing, Burlington, Canada, 2007

[3] CTFC (Capture the flag contest) defcon datasets, http://cctf.shmoo.com/data/, 2011.Accessed 09 January 2011.

[4] DARPA Intrusion Detection Scenario Specific Datasets http://www.ll.mit.edu/mission/communications /ist/corpora/ideval/data/2000data.html,.Accessed 09 January 2011.

[5] D.E. Denning. An Intrusion Detection Model, *IEEE Trans. Software Eng.,* Vol. 13, pp. 222-232, 1987.

[6] G. Brebner. *Computers in Communication*, McGraw-Hill, UK, 1997

[7] G.V. Glass, V.L. Willso and J.M. Gottman. Design and Analysis of Time-Series Experiments, Information Age Publishing Incorporations, USA, 2008

[8] H. Debar and A. Wespi. Aggregation and Correlation of Intrusion-Detection Alerts, *Proc. Int'l Symp. Recent Advances in Intrusion Detection*, pp. 85-103, 2001.

[9] J.D. McCabe. *Network Analysis, architecture, and design*, 2nd edition, Morgan Kaufmann publisher, US, 2003.

[10] J. Viinikka. Debar, H. Debar, L. Me and R.S. Supelec. Time Series Modeling for IDS Alert Management, In *proceedings of the ACM Symposium on InformAtion*, Computer and Communications Security (AsiaCCS), March 2006

[11] J. O. Nehinbe Methods for reducing workload during investigations of Intrusion Logs, PhD thesis, University of Essex, Colchester, UK, 2011

[12] J.D. McCabe. *Network Analysis, architecture, and design*, 2nd edition, Morgan Kaufmann publisher, US, 2003.

[13] J. Han and M. Kamber. *Data mining: concepts and techniques*, 2nd edition, Morgan Kaufmann publisher, US, 2006.

[14] K. Scarfone and P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS)”, Recommendations of the National Institute of Standards and Technology, Special Publication 800-94, Technology Administration, Department of Commerce, USA, 2007.

[15] P. Ren, Y. Gao, Z. Li, Y. Chen and B. Watson. IDGraphs: Intrusion Detection and Analysis Using Histographs, Department of Computer Science
Northwestern University, USA, 2005

[16] R. Yusof, S.R. Selamat and S. Sahib (2008). Intrusion Alert Correlation Technique Analysis for Heterogeneous Log, International Journal of Computer Science and Network Security (IJCSNS), Vol.8, 2008
.

[17] R. Rehman, *Intrusion Detection Systems with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP and ACID*, Prentice Hall PTR Upper Saddle River, New Jersey, 2003.

[18] W.A. Shay. *Understanding communications and networks*, 3rd Edition, Brooks/Cole, Belmont, CA, 2004

[19] X. Qin and W. Le. Statistical Causality of INFOSEC Alert Data, in p*roceedings of Recent Advances in Intrusion Detection, 2003*