A Distance-Based Method to Detect Anomalous Attributes in Log Files

Stefan Hommes, Radu State, Thomas Engel
University of Luxembourg, SnT
6, rue R. Coudenhove-Kalergi, L-1359 Luxembourg
Email: {stefan.hommes, radu.state, thomas.engel}@uni.lu

***Abstract*—Dealing with large volumes of logs is like the proverbial needle in the haystack problem. Finding relevant events that might be associated with an incident, or real time analysis of operational logs is extremely difficult when the underlying data volume is huge and when no explicit misuse model exists. While domain-specific knowledge and human expertise may be useful in analysing log data, automated approaches for detecting anomalies and track incidents are the only viable solutions when confronted with large volumes of data. In this paper we address the issue of automated log analysis and consider more specifically the case of ISP-provided firewall logs. We leverage approaches derived from statistical process control and information theory in order to track potential incidents and detect suspicious network activity.**

I. Introduction

The auditing of large log files is challenged by the ever-increasing volumes of data to be processed, and thus requires both processing facilities and conceptual solutions for this purpose. In incident response scenarios, a human analyst is required to process a large quantity of log data in order to find suspicious activities and correlate them with additional pieces of evidence. In many cases, this incident response is triggered off-line after some additional facts have been identified. Performing on-line incident detection is very difficult for several reasons. The many application-specific log formats also require deep domain-specific knowledge in order to properly configure existing rule-based event correlation engines. Secondly, a precise misuse model has to be given, or efficient detection algorithms are required.

We address these issues in our paper and propose a general framework that was implemented on the specific case of firewall log files. In the first phase, a sliding window over the log file compares successive windows by applying a set of appropriate distance functions. Furthermore the observed time differences are analysed with statistical quality process control rules. Suspicious time instants are identified and presented for examination by an analyst. However, the described method is general, and can be applied to many types of other log formats. Since many log formats include similar types of data (IP address, timestamps, actors and objects) and comply to the generic information model described in the seminal paper [1], a simple data type-based conversion is sufficient.

The paper is structured as follows: Related work is presented in section II. We then define a metric to specify the degree of anomalous connections in section III. An on-line method to analyse log files with control charts is presented in section IV. The results are presented in section V and we conclude our work in section VI.

978-1-4673-0269-2/12/$31.00 © 2012 IEEE

II. Related Work

To allow internet connected devices to communicate with each other in a securely manner, firewalls examine all packets that pass through them according to specific rules set up by the system administrator. These rules might block, allow or log certain connections. Dependent on the configuration of the firewall, all accepted and dropped connections are stored and archived in log files for later inspection. It is common practice that, due to the lack of real-time analysing tools, such log files are only analysed after an incident has occurred.

To find certain connection patterns that could be originated by an attack, the information content of these log files can be used to detect anomalous connection attempts. Since the size of log files and the amount of events in a system can grow very fast, analysing this data can be very challenging. Furthermore, a certain event by itself may be unimportant but be in fact a precursor of an attack (e.g. denial of service), or an error when occurring in conjunction with a second event.

The analysis of firewall configurations was pioneered by the papers [2], [3], where several types of potential conflicts and conflict resolution algorithms are proposed. While our work does not address the analysis of static firewall rules per se, the detection of firewall misconfiguration could leverage a cross-correlation between the static analysis of firewall rules and observed anomalies and events in the logs. The work described in [4], [5] is more similar to our approach in combining observed traffic with existing and deployed firewall configurations. There are however some major differences. We do not aim to optimize the storage of firewall rules or to reverse engineer a set of deployed rules, but focus on monitoring the operation of deployed firewall rules in order to detect potential incidents. Our approach, thus, is independent of the set of rules, but assumes that a working and operational configuration is in place. Some recent work leveraging association learning has been proposed in [6], [7], where sequences of events are mined in order to learn patterns related to an observed fault. This is complementary to our work, where we do not assume a known set of faults, but infer from differences between successive events that some

Attribute		Data type	Example	Metric
-	Number	number	4237	-
-	Date	string	16May2011	-
-	Time	number	0:05:11	-
-	Type	string	Log	-
a_1	Source Port	number	49954	J
a_2	Rule	number	298	KL
-	Current Rule Number	string	298-Standard	-
-	Information	string		-
-	Product	string	VPN-1 Power/UTM	-
a_3	Interface	string	eth-s1/s1p1c3	KL
a_4	Origin	string	IP1220-Gare1	KL
a_5	Action	string	Drop	KL
a_6	Service	number	694	KL
a_7	Source	number	192.168.8.183	J
a_8	Destination	number	192.168.8.255	J
a_9	Protocol	string	udp	KL
-	Rule Name	string		-
-	User	string		-

TABLE I
LIST OF ATTRIBUTES OF A FIREWALL LOG FILE WITH EXAMPLE RECORD. A DISTANCE METRIC (J=JACCARD SIMILARITY, KL=KULLBACK-LEIBLER DIVERGENCE) IS DEFINED FOR THE ANALYSED ATTRIBUTES IN THIS PAPER.

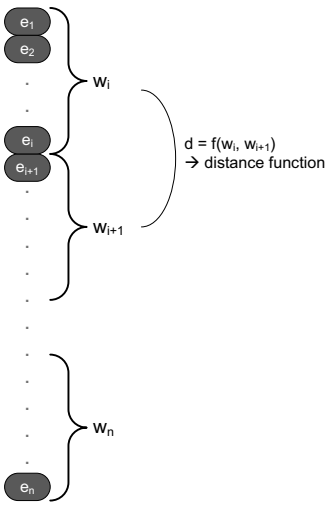

Fig. 1. Events are grouped into windows, which can be compared by a distance function

anomaly occurred. The mining of log data using a rule-based event correlator has been addressed in [8], [7]. Even though our approach leverages rules derived from statistical process control, the sequences of triggered alarms could be post-processed using a tool like SEC [9]. Although we have not yet implemented it, such post-processing could also use an unsupervised clustering method similar to the work described in [10]. From an operational perspective, [11] provides a rich and powerful application programming interface for event correlation, but human knowledge and expertise are required to use it.

In the intrusion detection community, [12] describes a method that monitors changes in event intensity and uses probabilistic techniques to evaluate events in network traffic data. That paper also uses Exponentially Weighted Moving Average (EWMA) control charts to search for anomalous changes in the event intensity of both correlated and uncorrelated data. In the context of supervised machine learning techniques, the authors of [13] leverage support vector machines to learn the predicting signs for failures. The approach is dependent on a labelled set of data and requires computational resources that are not appropriate for a large scale online monitoring of a firewall.

III. WINDOW-BASED SCORING OF LOG FILE EVENTS

In our approach, we consider all accepted or dropped connections from a firewall as data that must analysed from a log file. Each record is seen as an event e_i and described by a list of attributes: $e_i = \{id, date, time, a_1, ..., a_n\}$. The possible attributes and values from an example record are displayed in Table I. A log file can be considered as a sequence of such events, so that $L = \{e_i, ..., e_n\}$.

To reduce the amount of data and to incorporate the dependencies from temporally related events, we divide the log file into several windows w_i as presented in Figure 1. Each window contains a certain number of events and can be build in two different ways. A fixed window type contains a defined number of events w_s, where the time of occurence is not considered to be important. This can be useful if empty windows should be avoided, for instance during low traffic at night time. The fixed window does not consider event intensity, so we may miss anomalies that last only for a few seconds but have a high frequency. This can be avoided if we choose a window that contains events from a certain time period w_t.

A. Determining the distance between two windows

To determine if a window is anomalous or an outlier, we calculate the distance between two consecutive windows w_i and w_{i+1}. All values from each attribute (e.g. "TCP" (protocol)) in the first window are compared with the corresponding attribute in the second window. Since each attribute is different in data type and range (e.g. source IP and protocol), we utilize the Kullback-Leibler divergence and the Jaccard similarity coefficient to calculate this distance. Later is used if an attribute can have a high number of different values (e.g. IP address). To determine a single value that describes the similarity between two windows, we define a score s. It is calculated by summing all Kullback-Leibler distances and Jaccard similarities for all attributes (see equ. 1). The method used for each attribute is defined in table I, where the Kullback-Leibler divergence is calculated for $I_1 = \{a_2, a_3, a_4, a_5, a_6, a_9\}$ and the Jaccard similarity for $I_2 = \{a_1, a_7, a_8\}$.

$$s = \sum_{i \in I_1}^{n} \tilde{D_{a_i}} + \sum_{i = \in I_2}^{n} J_{a_i} \quad (1)$$

IV. ON-LINE DETECTION WITH CONTROL CHARTS

In on-line mode, the characteristic and trend of a series of scores from consecutive windows can give information about attacks, misconfigured networks or other

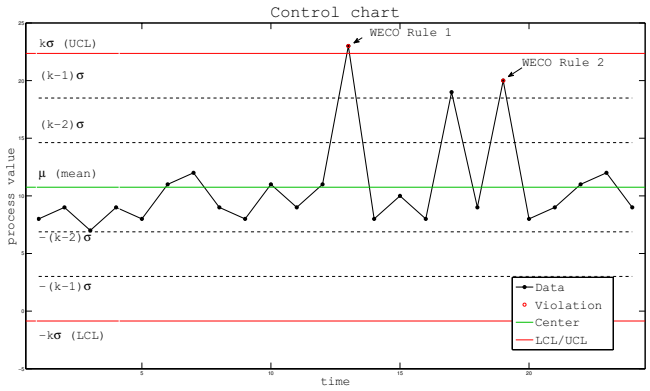

Fig. 2. Control chart with two detected data points enabling control rules Rule 1 (any point above or below $k\sigma$) and Rule 2 (2 out of the last 3 points above or below $(k-1)\sigma$)

types of network failure. A sequence of scores is calculated by comparing successive windows: $S = \{s_1(w_0w_1), s_2(w_1w_2), s_3(w_2w_3)...s_n(w_{n-1}w_n)\}$. Since each connection in the log file can last over several windows, we smooth the current window score using the formula 2. The smoothing constant λ describes the weight of the current data compared to past data, where $\lambda = 1$ only weights the current, and $\lambda = 0$ only weights past data.

$$\tilde{s}_i = \lambda s_i + (1-\lambda)\tilde{s}_{i-1} \tag{2}$$

Control charts are an important tool to analyse a sequence of values and detect if certain limits are exceeded. They are deployed in statistical process control, and are used to determine if a process variable is within certain limits. The advantage of control charts is that a human operator can observe the current and past trend of the process value. As shown in Figure 2, the chart consists of a centre line, the upper control limit (UCL) and a lower control limit (LCL). The centre line is the mean $\mu_{\tilde{S}}$ of the historical data, and is updated each time a new score is added. UCL and LCL are determined by:

$$\begin{aligned} UCL &= \mu_{\tilde{S}} + k\sigma_{\tilde{S}} \\ LCL &= \mu_{\tilde{S}} - k\sigma_{\tilde{S}} \end{aligned} \tag{3}$$

The constant k is defined as a factor of standard deviation $\sigma_{\tilde{S}}$. The value $k = 3$ is an accepted standard in industry; alternative values can be chosen using the Lucas and Saccucci [14] tables. Besides generating an alarm when the control limits (LCL/UCL) are exceeded, further control rules can give additional information about the process state since these also use the $(k-1)\sigma_{\tilde{S}}$ and $(k-2)\sigma_{\tilde{S}}$ ranges. There exist several well known rules, known as Western Electric Company Rules (WECO) [15]. The most common are presented in the table II. The WECO rules increase the sensitivity in detecting trends or drifts of the control variable, but also increase the number of false alarms. Therefore, an alarm raised by an out of control signal considered carefully when activating all rules on an new dataset.

Rule	Description
1	Any point above or below $k\sigma$
2	2 out of the last 3 points above or below $(k-1)\sigma$
3	4 out of the last 5 points above or below $(k-2)\sigma$
4	8 consecutive points above or below mean

TABLE II
DEFINITION OF WECO RULES [15]

	Dataset 1		Dataset 2	
Number of Events	49550		1016811	
Time period	1h 39 min		24h	
Date	15.05.2011		05.06.2011	
Data size	10.4 MB		196.7 MB	
	Unique Values		Unique Values	
	total	$ws = 100$	total	$ws = 100$
Source Port	21696	70,7	62982	84
Rule	34	7,1	52	6,4
Interface	26	7	32	6,4
Origin	2	1,9	2	1,5
Action	4	2	5	2
Service	3293	19,2	65536	23,6
Source	2657	47,8	28727	42,1
Destination	449	34,1	4000	26,4
Protocol	5	3	5	2,9

TABLE III
DATASET STATISTICS: THE LOWER PART SHOWS THE UNIQUE VALUES FOR EACH ATTRIBUTE IN THE THE DATASET AND THE AVERAGE FOR A WINDOW SIZE OF $w_s = 100$

V. EXPERIMENTAL RESULTS

Our industrial partner, the Luxembourg P&T company, provided us with two datasets from a Checkpoint firewall cluster, which is used as an internet firewall for their internet-connected servers. The dataset contains a smaller sample dataset and a dataset that represents the traffic from a normal working day (24 h). The dataset statistics are presented in Table III. The experiment was done with all WECO rules enabled and a fixed window size of $w_s = 100$. For both datasets, the number of activated WECO rules are shown in Table IV. The following records are a summary of suspicious and anomalous connection attempts to the company network:

A. Results for Dataset 1

The following record occurred 73 times and was identified by WECO Rule 1 and 2:

```
"27536" "16May2011" "0:47:50" "Log" "sip_any"
"298" "298-Standard" "" "VPN-1 Power/UTM"
"eth-s2/s2p2c1" "⋆ ⋆ ⋆" "Drop" "sip_any" "⋆ ⋆ ⋆"
"⋆⋆⋆" "udp" "" ""
```

The reason for this kind of connection attempts can either be a badly configured VoIP configuration, or a fraudulent use of SIP. In this specific case, many SIP Register requests occurred without success, and the system identified this anomaly correctly.

WECO rules	Dataset 1	Dataset 2
1	10	1111
2	4	1300
3	4	1521
4	8	1540

TABLE IV
NUMBER OF ACTIVATED WECO RULES FOR DATASET 1 AND DATASET 2

The next record occurred 72 times and was identified by WECO Rule 1:

```
"38023" "16May2011" "1:10:41" "Log" "4561"
"298" "298-Standard" "" "VPN-1 Power/UTM"
"eth-s2/s2p2c1" "⋆ ⋆ ⋆" "Drop" "telnet" "⋆ ⋆ ⋆"
"⋆⋆⋆" "tcp" "" ""
```

The telnet service is known to be less secure then SSH, because it lacks integrity and confidentiality protection. Inbound telnet traffic is a clear sign of malicious activity and was correctly identified by the system.

The following record occurred 119 times and was identified by WECO Rule 1 and 2:

```
"45284" "16May2011" "1:27:50" "Log" "X11"
"298" "298-Standard" "" "VPN-1 Power/UTM"
"eth-s2/s2p2c2" "⋆ ⋆ ⋆" "Drop" "MySQL" "⋆ ⋆ ⋆"
"⋆⋆⋆" "tcp" "" ""
```

In this case, an attempt is made to access an internal database server. Since inbound access to database servers is a clear violation of deployed access control mechanisms, such an incident is highly relevant for potential data stealing.

B. Results for Dataset 2

The following record occurred 157000 times and was identified by WECO Rule 1, 2, 3 and 4:

```
"230236" "6Jun2011" "7:54:34" "Log" "36446"
"301" "298-Standard" "" "VPN-1 Power/UTM"
"eth-s2/s2p2c2" "⋆ ⋆ ⋆" "Drop" "26870" "⋆ ⋆ ⋆"
"⋆⋆⋆" "tcp" "" ""
```

The huge volume of attempted SMTP connections triggered these four WECO rules. Manual inspection of the concerned address showed that it was listed as a *comment spammer* in Project Honey Pot[1].

VI. CONCLUSION AND FUTURE WORK

In this paper we have described an automated approach to the analysis of log files. In a first phase, a chart tracking algorithm identifies the most obvious suspicious activities. A human operator may confirm the findings in a second optional phase. The first phase is based on tracking temporal differences between successive windows of grouped events. The difference is computed, taking into account information theoretical measures and relevant set-specific distances. Outliers that might be anomalies are then identified with statistical process control techniques. The most difficult part consists in evaluating the performance of our approach when a large-scale labelled ground truth is available. We could not perform this task, because such labelled dataset does not exist, though our approach could be used for such a purpose. It might perform an initial labelling, followed by a human-driven validation. Although we focussed specifically on firewall logs, the proposed method should be applicable to many types of logs, since the individual data entries share common features (timestamp, IP addresses, actions). We plan to validate our method also on different log formats (syslog, SNMP, Netflow) and assess their pertinence in future work.

[1] http://projecthoneypot.org

ACKNOWLEDGEMENT

The present project is supported by the National Research Fund, Luxembourg.

REFERENCES

[1] D. E. Denning, "An intrusion-detection model," *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, vol. 13, no. 2, pp. 222–232, 1987.

[2] E. Al-Shaer, C. R. Kalmanek, and F. Wu, "Automated security configuration management," *J. Network Syst. Manage.*, vol. 16, no. 3, pp. 231–233, 2008.

[3] E. Al-Shaer, "Designing, optimizing, and evaluating network security configuration," in *NOMS*. IEEE, 2008.

[4] M. Abedin, S. Nessa, L. Khan, E. Al-Shaer, and M. Awad, "Analysis of firewall policy rules using traffic mining techniques," *IJIPT*, vol. 5, no. 1/2, pp. 3–22, 2010.

[5] E. Al-Shaer, A. El-Atawy, and T. Samak, "Automated pseudo-live testing of firewall configuration enforcement," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 3, pp. 302–314, 2009.

[6] R. Vaarandi and K. Podins, "Network ids alert classification with frequent itemset mining and data clustering," in *Network and Service Management (CNSM), 2010 International Conference on*, oct. 2010, pp. 451 –456.

[7] R. Vaarandi, "Mining event logs with slct and loghound," in *NOMS*. IEEE, 2008, pp. 1071–1074.

[8] J. P. Rouillard, "Refereed papers: Real-time log file analysis using the simple event correlator (sec)," in *Proceedings of the 18th USENIX conference on System administration*. Berkeley, CA, USA: USENIX Association, 2004, pp. 133–150. [Online]. Available: http://dl.acm.org/citation.cfm?id=1052676.1052694

[9] R. Vaarandi, "Platform independent event correlation tool for network management," in *NOMS*. IEEE, 2002, pp. 907–909.

[10] A. Makanju, A. N. Zincir-heywood, and E. E. Milios, "Clustering event logs using iterative partitioning," in *In Proceedings of KDD 09*, 2009.

[11] J. Hwang, "Splunk, innovation behind," in *Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology*, ser. CHiMiT '09. New York, NY, USA: ACM, 2009. [Online]. Available: http://doi.acm.org/10.1145/1641587.1814304

[12] N. Ye, S. Emran, Q. Chen, and S. Vilbert, "Multivariate statistical analysis of audit trails for host-based intrusion detection," *Computers, IEEE Transactions on*, vol. 51, no. 7, pp. 810 –820, jul 2002.

[13] E. W. Fulp, G. A. Fink, and J. N. Haack, "Predicting computer system failures using support vector machines," in *Proceedings of the First USENIX conference on Analysis of system logs*, ser. WASL'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 5–5. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855886.1855891

[14] J. M. Lucas and M. S. Saccucci, "Exponentially weighted moving average control schemes: Properties and enhancements," *Technometrics*, vol. 32, no. 1, pp. pp. 1–12, 1990.

[15] W. Electric, *Statistical Quality Control Handbook*. Western Electric Corporation, Indianapolis, Ind., 1956.

[16] *IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubiquitous Networks and Services, NOMS 2008, 7-11 April 2008, Salvador, Bahia, Brazil*. IEEE, 2008.