J. Basic. Appl. Sci. Res., 2(5)4413-4322, 2012

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Logs Correlation: Current Approaches, Promising Directions, and Future Policies

Sayed Omid Azarkasb[1] and Saeed Shiry Ghidary[2]

Artificial Intelligence MSc, Faculty of Computer Engineering, Qazvin Branch Azad University, Tehran, Iran
Assistant Professor of Computer Science Department, Amirkabir University of Technology, Tehran, Iran

ABSTRACT

This paper provides a comprehensive policy of the approaches achievements in the research area of intrusion detection, especially logs correlation, an area that has been so active and prosperous in the past few years. Large volume and low quality of produced logs justify the need for further log processing. Network administrators are able to correlate log file entries manually. The problem with this approach is lack of flexibility. It is time consuming, and one doesn't get the general view of the log files in the network. Without this general view it is hard to correlate information between the network components. Events seemingly unessential by themselves can in reality be a piece of a larger threat. In this regard, different log correlation methods are proposed to improve alert quality and to give a comprehensive view of system security. In this paper, we show how different attacks are reflected in different logs and argue that some attacks are not evident when a single log is analyzed. We propose a new method to correlate intrusion logs with the main processes of intrusion detection system. This is based on a centralized log correlation system which is composed of six components: data provider, preprocessor, analyzer, manager and controller, responder, and evaluator. The proposed architecture satisfies essential requirements of centralization, normalization, consolidation, aggregation, correlation, visualization and remediation. The results of our experiments show that the centralized logs file correlation improve the effectiveness of intrusion detection.

KEYWORDS: Computer Security, Intrusion Detection, Logs Correlation, Misuse Detection, Anomaly Detection.

1. INTRODUCTION

The increasing reliance on networked computers, and the growing expertise in subverting such systems, makes intelligent and adaptive threat detection vital.

Computer security revolves around confidentiality, integrity and availability. Integrity refers to the trustworthiness of data or resources, and is usually phrased in terms of preventing improper or unauthorized change. Integrity mechanisms fall into two classes: prevention or detection. Prevention mechanisms try to maintain the integrity of data by blocking unauthorized attempts to change data. On the other hand, detection mechanisms do not try to prevent violations of integrity, but simply report that data integrity can no longer be assumed. Intrusion Detection Systems (IDSs) attempt to detect intrusion and attacks through analyzing events in computer systems or networks. They detect intrusions and attacks through analyzing audit and log file data. Based on the data source, IDSs can be classified into host-based and network-based. IDSs are also categorized into anomaly detection or misuse detection depending on how they analyze data [4].

Misuse detection systems detect known attacks using attack patterns and signatures known a priori, while anomaly detection systems detect attacks by observing deviations from normal behavior of the system, network, or users [3].

Attacks that categorized in three categories with different behavior: Denial of Service (DoS) attacks, User-to-Root (U2R) & Remote-to-Local (R2L) attacks and Probing [8]. Depend on a category, specific security policy should be taking.

The problem faced today by system administrators are the correlation of firewall logs, intrusion detection system event logs, operating system event logs, mail system logs, database logs, web server logs, antivirus logs, router/switch logs etc. All these logs can identify a threat, and may receive hundreds or thousands of entries a day. The examination of these logs is often performed by staff short of time and knowledge, and for a company; resources may be a limitation. Despite the resource limitations we want log file correlation because it improves intrusion detection [1].

This paper is structured as follows: Section 2 provides historical background for this work. Section 3 focus on the different types of attacks that are categorized in three categories: Denial of Service (DoS) attacks, User-to-Root (U2R) & Remote-to-Local (R2L) attacks, and Probing. Section 4 covers the many types of logs and different fields that could potentially be anonymized. Section 5 shows how multiple logs are affected by common attacks and give

***Corresponding Author:** Sayed Omid Azarkasb, Artificial Intelligence MSc, Faculty of Computer Engineering, Qazvin Branch Azad University, Tehran, Iran. Email: Azarkasb@ymail.com. Tel.:+98 21 66741274; fax: +98 21 66741387

several specific examples of attacks and their attack traces in heterogeneous logs. Section 6 discuses system architecture and main components of a log correlation system. Section 7 presents experimental results, and Section 8 draws conclusions.

2. Related Works

Logs data have beneficial information to security operation teams. Rinnan [11] demonstrated benefits of centralized log file correlation which can be summarized as:

- Eliminating manual device monitoring,
- Resolving security events in real-time,
- Security events that seem unimportant by themselves, when put in context of all other events monitored, can suddenly illuminate a major threat that may be caught too late,
- Centralization enables IT staff to integrate real time and historical information, improving visibility into events, trends, and recurrences. With the ability to centralize events also comes the ability to handle enterprise correlation.
- Accelerating identification of threats and events,
- Achieving a general view of the system.

He claimed that a complete system should implement the following criteria: Centralization, Normalization, Consolidation, Aggregation, Correlation, Visualization and Remediation.

In the research that is accomplished by Abad and et.al [1], they showed that correlating log information is useful for improving both misuse detection and anomaly detection by using two complementary approaches to map attacks and the attack traces in logs: top-down(attacks to logs) and bottom-up(logs to attacks). They suggested facilitating processing of the millions of entries found in typical logs, it is useful to apply data mining techniques.

Bahreyni [2] in his thesis proposed a causal probabilistic method to correlate intrusion alerts. In his approach, the uncertainty in prerequisite-attack, attack consequence, and attack-alert relations are involved in correlation process. He used expert knowledge in the form of probabilities of attack occurrence, success, and detection to improve correlation accuracy. Probabilities gathered from experts are used for probabilistic inference in a causal (Bayesian) network. Using probabilistic inference, he guessed attacks not detected by intrusion detection systems, and attacks probably coming next. Correlation functionalities can be configured using three parameters: attack threshold, correlation threshold, and prediction threshold.

Researches demonstrated difficulties and problems of centralized log file correlation can be summarized as follows:

- Automated security event collection and analysis when managing a security infrastructure requires a lot of people without the tools for automatic security event collection and analysis. This makes it an error-prone process [14].
- Most data collected from Intrusion Detection Systems (IDS) and security devices may be treated as noise; separating this noise from risks is nearly impossible. This overwhelms IT staffs with data and makes it nearly impossible to correlate the data [11].
- Audit log data are formatted differently; the communication of logging is standardized but not the content or format [11].
- Legislation often requires "adequate" protection [6].

[5, 9, 10] are some research in this area.

3. Attacks Categories

Table.1 shows the 88 different attacks that were throated operating systems. This table presents the attacks broken up into three categories. These three groups are: Denial of Service (DoS) attacks, User-to-Root (U2R) and Remote-to-Local (R2L) attacks, and Probing [8]. The next three sections present detailed behavior descriptions of each class of attacks.

a. Denial of Service Attacks

This type of attack is an attack in which causes the system cannot be answered legitimate user requests. The attacker makes some computing or memory resource too busy or too full to respond.

b. User-to-Root (U2R) and Remote-to-Local (R2L) Attacks

This type of attack is an attack in which the attacker access to local system from a remote machine. Achieving root user's permission, invalid and unauthorized access to system is done.

Table.1. Attacks and Their Categories.

Attach Name	Attack Category
Apache2-Arppoison-Back(Bonk)-Crashiis-Dosnuke (NetBios) Echochargen-Jolt-Killwin-Land-LinuxICMP-MailBomb-Moyari13 Nestea-NewTear-Octopus-Oshare-PingOfDeath(POD) ProcessTable-Saihyousen-SelfPing-Sesquipedalian-Smurf SshProcessTable-SYNDrop-SYNFlood(Neptune)-Syslogd-TcpReset TreaDrop-Twinge-UdpStorm-Winnuke-1234	Denial of Service
Anypw-Buffer Overflow-Casesen-Dictionary-Eject-Fdformat Ffbconfig-FtpWrite-GuessPassword-Guest-HttpTunnel-Imap LoadModule-MultiHop-Named-Ncftp-Netbus-Netcat-Ntfsdos-Perl Phf-Ppmacro-PS-RootKit-Sechole-SendMail-SMBdie-SNMPGet SNMPGuess-Spy-SshTrojan-WarezClient-WarezMaster Xlock-Xsnoop-Xterm-yaga	Remote to Local & User to Root
InsideSniffer-IpSweep-LsDomain-Mscan-Nmap-NTinfoscan PortSweep-Queso-ResetScan-Saint-Satan-SynScan TcpWindowScan-XmasTreeScan	Probing

c. Probes Attacks

This type of attack is an attack in which the attacker probes and scans a network of computers to gather information or find known vulnerabilities.

4. Log Varieties

A computer network contains a variety of different infrastructure devices which may be instrumented to produce multiple audit logs. A brief survey of some of the different types of logs is an important starting point in understanding security research. The fact that the audit logs are different is significant because it promotes multiple views for discovery, robustness against attack, interoperability, extensibility and flexibility. While one type of log may not be enough to break the anonymization system, information may be inferred from multiple logs that can be used in a successful attack against anonymized data. Thus we seek to create anonymization schemes for many types of logs.

What follows is a description of commonly implemented network and system logs summarized from [13]. These logs provide situational awareness of what is happening where and when on networks/systems by auditing system activities, transactions performed and network signaling. These logs are useful for detecting network problems, malicious activity and recovery from accidental or intentional failures.

5. Logs Correlation

As discussed in previous section, there are many independent logs that store information for their own purposes. They are sometimes not well organized, and information overlap exists among different logs. Instead of concentrating on system vulnerabilities, each of which can be exploited in several different ways, we will analyze specific attacks and the way they affect the different logs. We have created log correlation tables according to the logs described in previous section and the attacks behavior described in section 3. It needs to be noted that as in the actual networks, the volume of attacks has the following descending order: Denial of Service Attacks, User-to-Root (U2R) & Remote-to-Local (R2L) Attacks, and Probes Attacks, we also have taken this priority into the table 2. Table 2 shows complete and comprehensive information between three attacks categorized behavior and their effects in some important logs such as:

a. Log Inspected Priority Based on Attacks Category

After analyzing how attacks affect each log, we are able to identify six important logs which are affected by more than half of attacks. They need to be checked first. According to figure 1 the most important logs are: DNScache, NetFlow, Syslog, Dial-up, SNMP and Firewall. Analyzing these six logs and correlating the information found in them will help improving IDS performance.

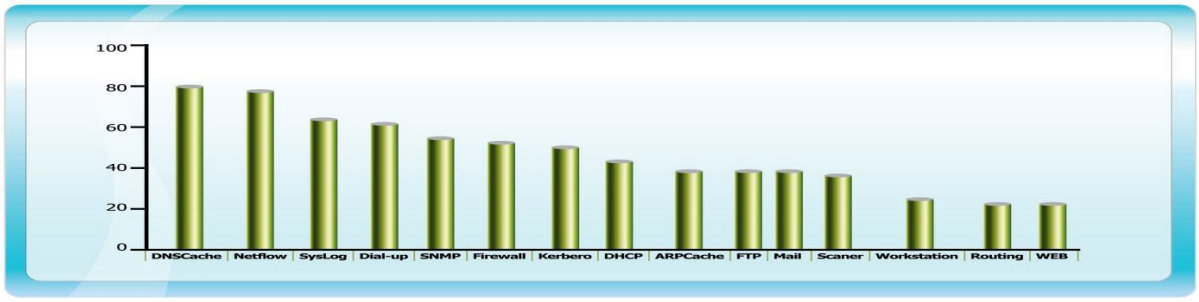


Fig. 1. Average logs effectiveness with common attacks.

Table. 2. Log Correlation Table.

Attack Category	Attack Name	ARPCache	DHCP	Dial-up	DNSCache	Firewall	FTP	Kerbero	Mail	Netflow	Routing	Scaner	SNMP	SysLog	WEB	Workstation	Total Reference / Reference Rate
Denial of Service	Apache2		✓			✓		✓		✓			✓	✓	✓		7 / 46.66
	Bonk(Back)		✓	✓				✓					✓	✓	✓		6 / 40.00
	Dosnuke(NetBios)				✓				✓	✓		✓	✓	✓	✓	✓	7 / 46.66
	EchoChargen		✓	✓			✓			✓			✓				5 / 33.33
	Jolt				✓	✓			✓	✓			✓				5 / 33.33
	KillWin			✓			✓						✓	✓		✓	5 / 33.33
	Land	✓	✓		✓				✓	✓			✓				6 / 40.00
	MailBomb		✓	✓		✓			✓	✓			✓	✓	✓		8 / 53.33
	Moyari13				✓				✓	✓			✓				4 / 26.66
	Nestea		✓		✓	✓				✓		✓					5 / 33.33
	NewTear				✓	✓				✓	✓					✓	5 / 33.33
	Oshare		✓			✓			✓		✓		✓		✓	✓	7 / 46.66
	PingOfDeath(POD)		✓	✓		✓				✓			✓			✓	5 / 33.33
	ProcessTable	✓		✓	✓	✓		✓	✓	✓		✓					8 / 53.33
	Smurf			✓	✓			✓	✓	✓			✓				6 / 40.00
	SYNFlood(Neptune)			✓	✓	✓		✓		✓		✓		✓			7 / 46.66
	TearDrop	✓	✓		✓	✓				✓			✓	✓			7 / 46.66
	UdpStorm		✓	✓	✓					✓			✓	✓			6 / 40.00
	Winnuke			✓	✓	✓		✓		✓	✓	✓				✓	8 / 53.33
	Access Number	9	4	11	16	8	11	12	7	12	25	5	7	13	5	5	117
	Access Rate	52.94	32.52	64.70	94.11	47.05	64.70	70.58	41.17	70.58	29.41	29.41	41.17	76.41	29.41	29.41	41.05
Remote to Local & User to Root	AnyPW			✓	✓	✓	✓	✓								✓	6 / 40.00
	CaseSen	✓			✓		✓	✓			✓					✓	6 / 40.00
	Dictionary	✓		✓	✓	✓	✓	✓		✓				✓	✓		9 / 60.00
	FrameSpoofer					✓	✓	✓	✓	✓			✓	✓	✓		8 / 53.33
	FtpWrite	✓		✓	✓		✓							✓			5 / 33.33
	HttpTunnel			✓	✓			✓	✓	✓					✓		6 / 40.00
	Imap				✓	✓	✓		✓	✓	✓	✓		✓			8 / 53.33
	Named			✓	✓			✓	✓	✓				✓	✓	✓	8 / 53.33
	NetBus	✓	✓		✓		✓	✓	✓	✓		✓	✓	✓			10 / 66.66
	NetCat			✓	✓	✓	✓	✓	✓	✓		✓	✓	✓			10 / 66.66
	NTFsDos	✓		✓	✓	✓	✓	✓		✓	✓		✓	✓		✓	11 / 73.33
	Perl	✓	✓	✓	✓						✓	✓					6 / 40.00
	Phf				✓		✓	✓		✓	✓			✓	✓		7 / 46.66
	Sechole	✓		✓	✓		✓	✓					✓	✓			7 / 46.66
	SendMail	✓		✓	✓	✓		✓	✓	✓		✓	✓	✓			10 / 66.66
	Xsnoop		✓	✓	✓	✓				✓				✓			6 / 40.00
	Yaga	✓	✓		✓	✓				✓			✓	✓		✓	7 / 46.66
	Access Number	3	9	10	12	11	2	6	8	16	3	4	14	8	5	6	130
	Access Rate	15.78	47.36	52.63	63.15	57.89	10.52	1.57	42.10	84.21	15.78	21.05	73.36	42.10	26.31	31.57	50.98
Probing	InsideSniffer	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓			11 / 73.33
	IPSweep	✓	✓	✓	✓		✓			✓		✓	✓	✓			9 / 60.00
	LsDomain	✓	✓	✓	✓							✓					5 / 33.33
	Mscan	✓	✓		✓							✓		✓			5 / 33.33
	Nmap	✓	✓	✓		✓				✓		✓		✓			5 / 33.33
	NTinFoScan			✓	✓	✓	✓	✓		✓		✓		✓			8 / 53.33
	Queso		✓		✓			✓	✓	✓			✓	✓			7 / 46.66
	ResetScan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			13 / 86.66
	Access Number	5	6	6	7	4	4	4	2	6	2	7	3	7	0	0	63
	Access Rate	62.50	75.00	75.00	87.50	50.00	50.00	50.00	25.00	75.00	25.00	87.50	37.50	87.50	0	0	52.50
	Access Number	17	19	27	35	23	17	22	17	34	10	16	24	28	10	11	310
	Access Rate	38.63	43.18	61.36	79.54	52.27	38.63	50.00	38.63	77.27	22.72	36.36	54.54	63.63	22.72	25.00	46.49

By analyzing table 2 the following new approaches are proposed for system security policy based on particular type of attacks category:

Priority based on dos Attacks

As shows in figure 2, we are able to identify five important logs that are affected by more than half of attacks. They need to be checked first as left to right priority: Netflow, SNMP, DNSCache, Firewall, Dial-up. And other logs arranged with left to right priority: DHCP, Mail, Syslog, Kerbero, Workstation, Web, Scanner, ARPCache, Routing, and FTP.

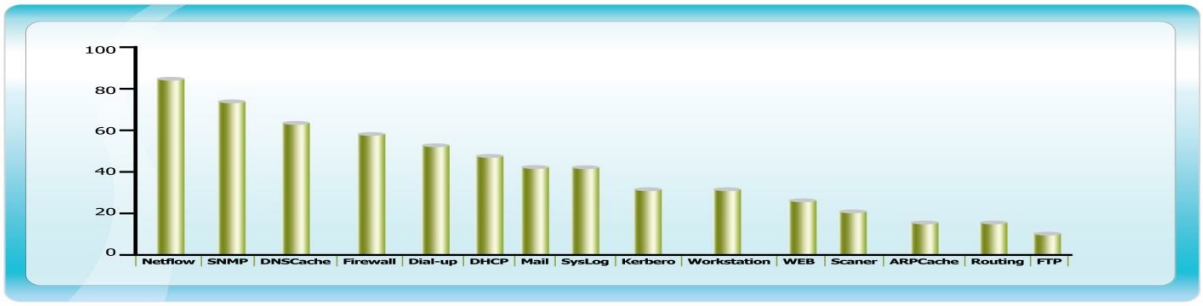


Fig. 2. Average DoS attacks affect in different logs.

Priority based on U2R&R2L Attacks

As shows in figure 3, we are able to identify seven important logs that are affected by more than half of attacks. They need to be checked first as left to right priority: DNSCache, Syslog, Kerbero, Netflow, Dial-up, FTP, ARPCache. And other logs arranged with left to right priority: Firewall, Mail, SNMP, Routing, Scanner, Web, Workstation, and DHCP.

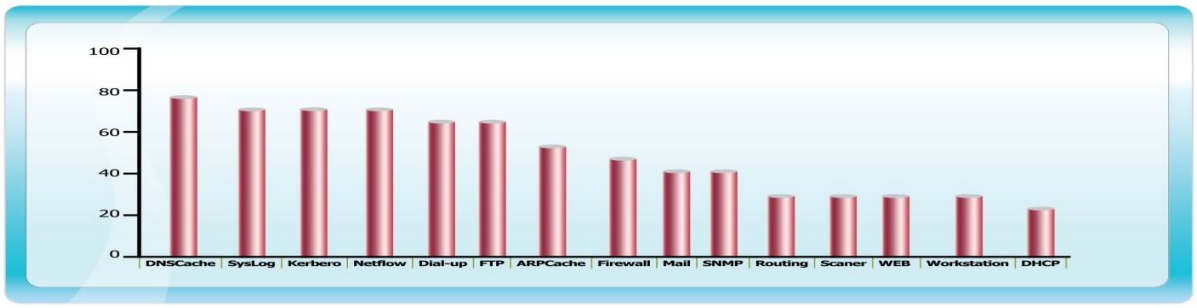


Fig. 3. Average U2R&R2L attacks affect in different logs.

Priority based on probe Attacks

As shows in figure 4, we are able to identify seven important logs that are affected by more than half of attacks. They need to be checked first as left to right priority: DNSCache, Scanner, Syslog, DHCP, Dial-up, Netflow, ARPCache. And other logs arranged with left to right priority: Firewall, FTP, Kerbero, SNMP, Mail, Routing, Web, and Workstation.

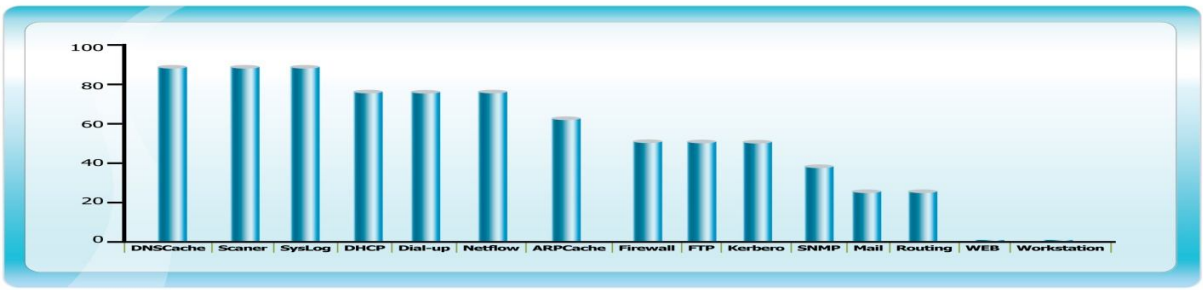


Fig. 4. Average probe attacks affect in different logs.

As we saw, ARPCache log that is not important in DoS detection is very important in U2R&R2L and probe detection.

As it is shown in figure 5 the Probe category has the greatest logs affect, however they are less allocating on system. For example ResetScan can be traced in 13 logs.

Fig. 5. Average attacks category affect in different logs.

Figure 6 demonstrates attacks effective rate in different logs.

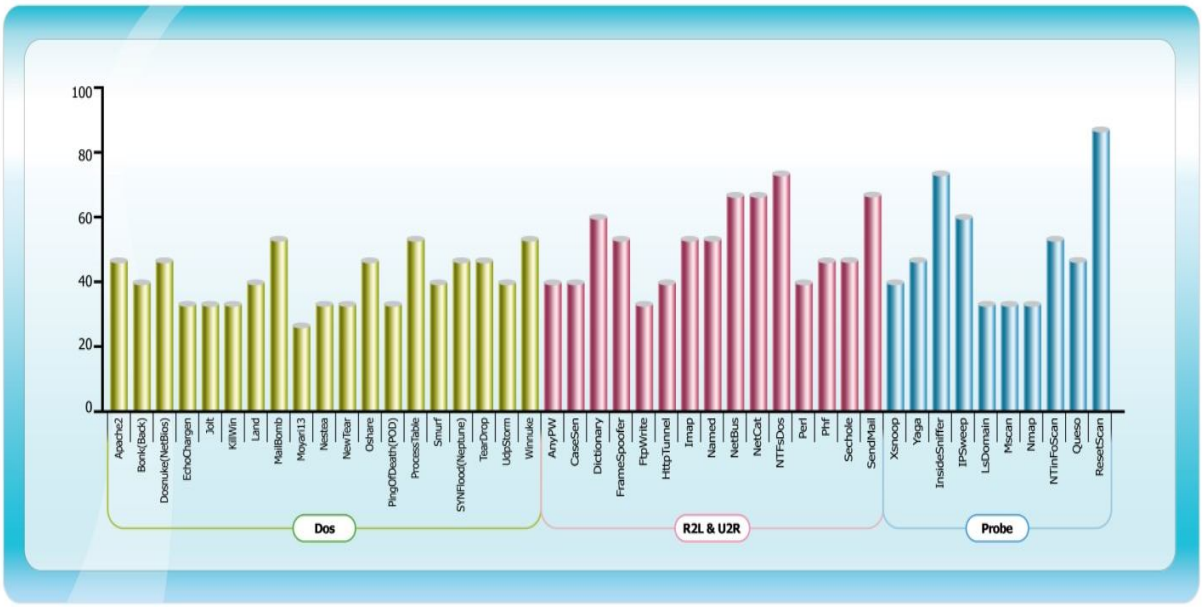


Fig. 6. Average attacks effective in system logs.

6. Log Correlation System Architecture

A complete log correlation system should implement the following criteria: Centralization, Normalization, Consolidation, Aggregation, Correlation, Visualization and Remediation [11].

1. Centralization: The principle is to collect logs which are selected in previous section from their devices. The consequence is that it has to incorporate components such as routers, managed switches and other dedicated hardware with logging capabilities.

2. Normalization: XML seems to be the preferred choice for structuring the data. The log entries should be structured by all possible fields. The most appropriate fields are dynamic fields such as time, date, src/dst IP and port, user, etc.

3. Consolidation: By naming already known common vulnerabilities and exposures it would be possible to compare and analyze the gathered data. By naming an event, it is possible to determine which events are significant and are related to a particular attack.

4. Aggregation: It means grouping similar events together and giving answer to how many times an attack happens over a certain time period.

5. Correlation: A product should be able to use a top-down, and bottom-up approach for correlation. How good we are at incorporating normalization, consolidation, and aggregation will affect our ability to correlate. There exist a lot of correlation techniques, but the point is that we want to be able to receive an alert and trace an event, or to get

to the level of a single log file entry, and trace events from that entry. The quality of the correlation step relies on the level of automation of the tool.

6. Visualization: The ultimate goal is to reduce complexity thus the visualization part of the tool should be as simple as possible. We need to have a view of all networks and components, and at the same time to be able to look at entries from a single log file from a single component. Designing graphical representations is desirable, but such representations must simplify the view, not make it more complex.

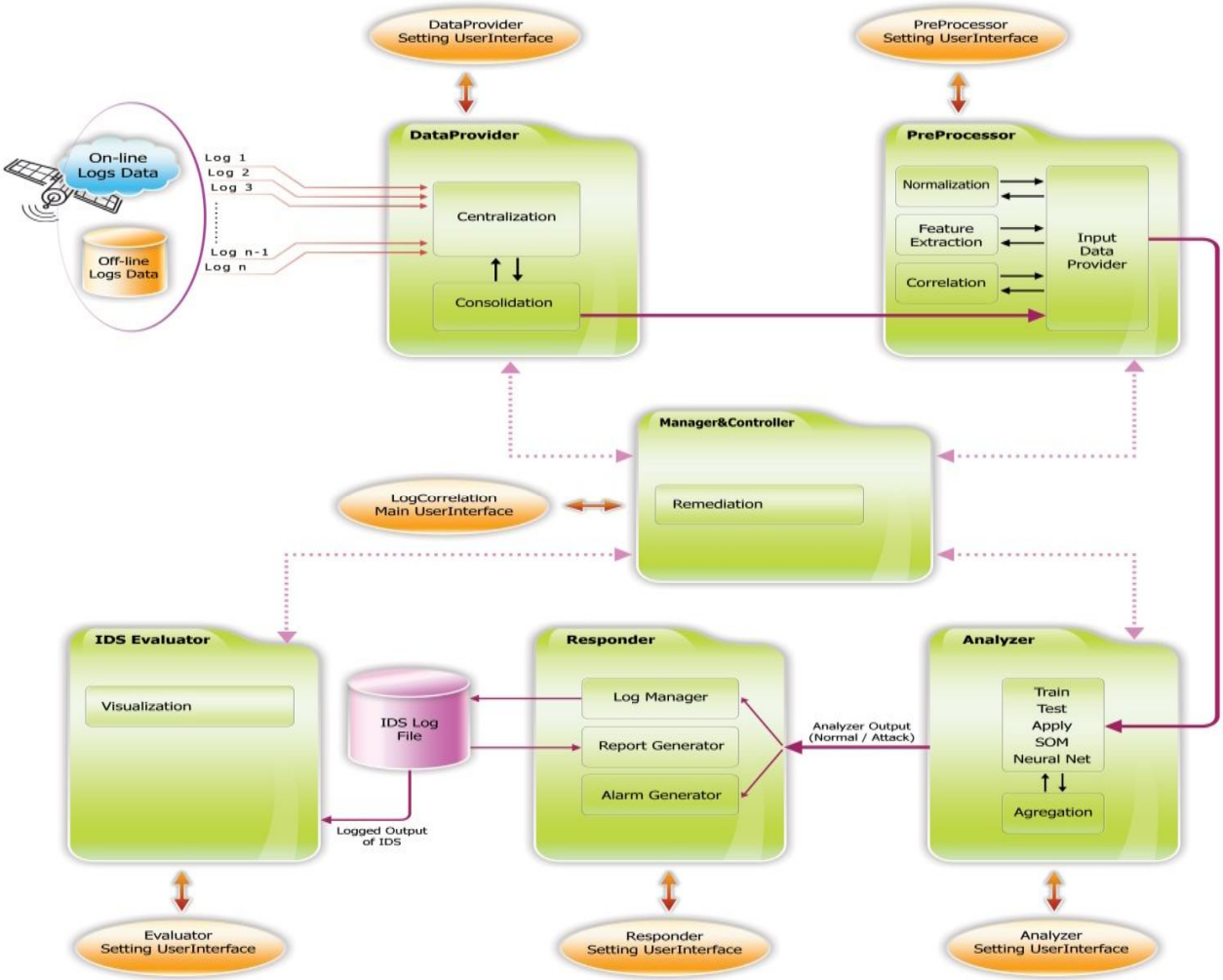
7. Remediation: Remediation may not be the responsibility of such products, but can rather be done manually. Reactive responses are certainly possible, but such capabilities should be limited, because we do not want to block legitimate use. Maybe reactive logging is more appropriate to gather more information in the event of probable malicious activity [11].

The architecture and main components of our log correlation system is shown in figure 7.

In our proposed system, Data Provider collects data from network logs audited data file (off-line mode) or live network logs (on-line mode) and sends text data to the Processor component.

Preprocessor converts text data into numeric one and if necessary converts numeric data into binary or normalized form, and sends them to a SOM Neural Net Based Analyzer. In Preprocessor, after extracting features from each record, each feature is converted from text or symbolic form into numerical form. The next step in preprocessing is converting data into binary, or normalized and scaled form. For normalizing feature values, a statistical analysis is performed on the values of each feature, based on the existing data from dataset and then acceptable maximum value for each feature is determined.

Fig. 7. Log Correlation Intrusion Detector System Architecture.



The analyzer uses data either for training and testing its SOM neural net or for analyzing and detecting intrusions/attacks. Moreover, this component provides facilities for training and testing an unsupervised neural net for intrusion detection purpose (before bringing the system into application in the real environment). As unsupervised neural nets can classify input data based on their similarity, SOM nets are used in our system for clustering and classifying network traffic into normal and intrusive. The analyzer output (normal or attack type) is given to Responder for generating alarm (in different ways such as sending e-mail to system administrators and displaying the appropriate message on the screen) and recording detailed or statistical reports on the collected data in IDS log files.

The IDS Evaluator component provides a facility for reporting true detection rate (only separating normal traffic from attack), true type detection rate (detecting normal traffic from attack and recognizing the known attack type), false positive detection rate (miss detecting attack), false negative detection rate (failing to detect attack when it is occurred), and other criteria such as detection rate of three attacks categories, to evaluate our log correlation based intrusion detection system.

In brief, log correlation system works in four modes: 1) off-line training, 2) off-line testing, 3) as a real on-line IDS, and 4) as a real off-line IDS.

The Manager & Controller component, manages and directs other component to work in one of the above modes based on the command and parameters delivered from the operator.

7. Experimental Results

For our experiment we applied some of the existing attack tools to generate a group of attacks against a local network server and collected the produced traffic. To gather the normal traffic, we recorded samples of the usual traffic of the network within a 4 days period. Thus, we had a training data collection of 5114 instances including 23 known attack types and a test data collection of 3919 instances including the aforementioned 23 attack types plus 8 other attack types. The attacks existed in the test data set along with the corresponding tools for their generations are presented in Table 3.

Table. 3. Train and test data attack types with their tools and instances.

Attack Name	Attack Generation Tools	Train	Instance	Test	Instance
Bonk	targa2.c	✓	30	✓	30
BrKill	BrKill.c	✓	40	✓	40
Dosnuke(NetBios)	FireHack	✓	50	✓	100
Imap	Imap4.c	✓	4	✓	4
Jolt	targa2.c	✓	189	✓	84
KillWin	Killwin.c	✓	13	✓	23
Land	targa2.c	✓	15	✓	45
Neptune(SYN Flood)	FireHack	✓	60	✓	200
Nestea	targa2.c	✓	90	✓	45
Newtear	targa2.c	✓	30	✓	30
Octopus	Octopus.c	✓	103	✓	100
Oshare	targa2.c	✓	50	✓	31
Saihyousen	targa2.c	✓	106	✓	23
SMBdie	Smbdie.exe	✓	8	✓	20
SYNDrop	targa2.c	✓	30	✓	30
SYNScan	Nmap	✓	217	✓	510
TcpWindowScan	Nmap	✓	202	✓	147
TearDrop	targa2.c	✓	30	✓	30
Twinge	Twinge.c	✓	93	✓	62
UdpPortScan	Nmap	✓	285	✓	285
Winnuke	targa2.c	✓	150	✓	50
XmassTreeScan	Nmap	✓	106	✓	72
1234	targa2.c	✓	50	✓	23
Apache2	Apache2.c	–	–	✓	98
EchoChargen	FireHack	–	–	✓	100
LinuxICMP	linux-icmp.c	–	–	✓	202
Moyari13	moyari13.c	–	–	✓	30
OpenTear	opentear.c	–	–	✓	100
OverDrop	overdrop.c	–	–	✓	40
Sesquipedalian	sesquipedalian.c	–	–	✓	60
Smurf	smurf4.c	–	–	✓	50

We determined the best values of important parameters using the SOM neural net, before evaluating the system. These include the number of cluster units in the output layer, the number of epochs for training, learning

rate, neighborhood type, neighborhood number and its distance. For this purpose, 50 epochs, 750 cluster units, learning rate 0.8, neighborhood type rectangular, neighborhood number 7 and Euclidian distance were achieved.

After determining appropriate parameter values for SOM, we evaluated their performance in real-time detection of network based attacks using the log correlation system. The evaluation results are shown in Table 4 based on the True detection Rate (TR), Exact True Type detection Rate (ETTR), False Positive detection Rate (FPR), and False Negative detection Rate (FNR). To compare the performance of proposed system we evaluated it two other approaches as shown in table 5.

Table. 4. Detection Performance of Log Correlation Intrusion Detection System.

TR	ETTR	FPR	FNR
97.75	73.82	1.53	0.71

Table. 5. Comparison of detection rate of proposed log correlation intrusion detection system and best result of learning approach in [12] and [7].

Attack Name / Approach	Dos	U2R & R2L	Probe
Best Result of Machine Learning In [Sabhnani 2003]	97.30	29.80	88.70
Result In [Jazzar 2008]	98.36	80.25	88.40
Proposed Approach	99.04	85.00	98.90

The results show that log correlation intrusion detection system has a superior performance for all attack categories. Low performance in the case of R2L&L2R attacks is due to the fact that it is difficult to detect them by a network based intrusion detector.

8. Conclusions

In this paper, we introduced a Log Correlation Intrusion Detector in real-time intrusion detection system, which are fed with live network logs. Logs are vital to both security operations and researchers. Security professionals analyze them on a daily basis. Using centralized log correlation in intrusion detection has many advantages over other systems. By analyzing and correlating the information found in multiple logs, intrusion detection systems are able to improve the effectiveness of IDS alarms. In this paper, we have emphasized the importance of logs, attacks and have discussed how different logs are affected by three categories of attacks. We applied logs correlation table with 44 common attacks and 15 important logs that represented beneficial and valuable information to researchers in this area. Specifically, we performed experiments to successfully correlate data from multiple logs for cases of anomaly detection and misuse detection. Therefore, this system is able to detect known attacks with their type as well as new unknown attacks. Using SOM neural nets in analyzer have many advantages too. The main advantage is the capability of unsupervised nets to improve their analysis of new data without retraining over all the previous/new data and have higher speed and lower response time in the deployment phase. We report empirical results showing improved IDS accuracy by correlating logs information.

REFERENCES

1. Abad. C., J. Taylor, C. Sengul, W. Yurcik, Y. Zhou, & K. Rowe, 2003. Log correlation for intrusion detection: A proof of concept, In Proc. of the 19th Annual Computer Security Applications Conf, ACSAC.
2. Bahreyni. P., & R. Jalili, 2007. An Intrusion Alert Correlation Method, Master's thesis, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.
3. Cannady. J., 1998. Artificial Neural Networks for Misuse Detection, Proceedings of the National Information Systems Security Conference.

4. Coolen. R., & H. A. M. Luiijf, 2002. Intrusion Detection: Generics and State-of-the-Art, Research and Technology Organization (RTO) Technical Report 49.

5. Debar. H., & A. Wespi, 2001. Aggregation and correlation of intrusion-detection alerts, In Proc, of the 4th Intl. Symposium on Recent Advances in Intrusion Detection، RAID.

6. Fisma. S., 2003. Overcoming Persistent FISMA Weaknesses through Security Compliance Management, NETFORENSICS WHITE PAPER .

7. Jazzar. M., & A. Jantan, 2008. A Novel Soft Computing Inference Engine Model for Intrusion Detection, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.4.

8. Kendall. K., 1999. A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, Master's thesis, MIT.

9. Ning. P., Y. Cui, & D. S. Reeves, 2002. Constructing Attack Scenarios through Correlation of Intrusion Alerts, In Proc. of the 9th ACM Conference on Computer & Communications Security, pages 245–254.

10. Porras. P. A., M. W. Fong, & A. Valdes, 2002. A Mission-impact-based Approach to Infuse Alarm Correlation. In Proc. of the 5th Intl. Symposium on Recent Advances in Intrusion Detection, RAID, pages 95–114.

11. Rinnan. R., 2005. Benefits of Centralized Log file Correlation, Master's Thesis Master of Science in Information Security, ECTS Department of Computer Science and Media Technology Gjøvik University College.

12. Sabhnani. M., & G. Serpen, 2003. Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context http://www.eecs.utoledo.edu/~serpen/professional/Research/Publication/MLMTA 2003 Manuscript Submission Version.pdf.

13. Slagell. A., & W. Yurcik, 2004. Sharing Computer Network Logs for Security and Privacy: A Motivation for New Methodologies of anonymization, SECOVAL: The Workshop on the Value of Security through Collaboration, Athens, Greece.

14. Walker. J., 2001. Security Event Correlation: Where Are We Now? NetIQ white paper, http://www.netiq.com/products/sm/whitepapers.asp.