

System Application Behavioural Profiling using Log Analysis

E.V.Soundariya¹, S.Kalaiselvi², Dr.K.Narasima Mallikarjuna³

¹UG Student, Dept. of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, Tamil nadu, India

²UG Student, Dept. of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, Tamil nadu, India

³Assistant Professor, Dept. of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, Tamil nadu, India

Abstract - Log analysis is a widely accepted concept with a collection of various log information not only about system faults, but also about security of the system. Since the data centers and networks create large number of logs, the manual review of that data is not easy for normal users. For automate this work, a number of tools and algorithms have been suggested in recent researches. The rapid development in the computer network brings both advantages like a great convenience to the users and disadvantages like security threats for system users and data. An application can be installed, updated and deleted from the system. Those data were recorded in the system as logs which are not known by the normal users. An uninstalled application may not completely delete all its packages. It may leave some pending packages, which are not known by the normal user. This research work proposes a LOGxy application for the usual users to review and understand the computer-generated logs.

Keywords: Log file analysis, Bash file, System security.

1. INTRODUCTION

The rapid development of a computer network brings both advantages such as user usability, better connectivity, data availability and disadvantages such as device user security threats and data theft. The log files are primarily used in the system to protect against unwanted code changes, intrusion and misuse. Log files contain valuable device information such as data use, who, what, where, how it was used, alerts and errors. There are two types of logs, logs for systems and logs for applications. It can be used to track a program, abuse and investigate irregularities from logs. When a device has been breached the log file is used as evidence.

Linux logs provide a visual history of everything that happens on the linux operating system. When something goes wrong, they will give us a helpful run down of all incidents to help the admin locate the culprit. Linux logs files should be easy to find, as the text format is stored in the directory and subdirectory of /var / log.

```
root@kalai:/home/kalai# cd /var/log
root@kalai:/var/log# ls -la
.
..
alternatives.log
alternatives.log.1
alternatives.log.2.gz
apt
auth.log
auth.log.1
auth.log.2.gz
auth.log.3.gz
auth.log.4.gz
btm
btm.1
cups
daemon.log
daemon.log.1
daemon.log.2.gz
daemon.log.3.gz
daemon.log.4.gz
debug
debug.1
debug.2.gz
debug.3.gz
debug.4.gz
dpkg.log
dpkg.log.1
dpkg.log.2.gz
faillog
fontconfig.log
gdm3
hp
installer
kern.log
kern.log.1
kern.log.2.gz
kern.log.3.gz
kern.log.4.gz
lastlog
messages
messages.1
messages.2.gz
messages.3.gz
messages.4.gz
speech-dispatcher
syslog
syslog.1
syslog.2.gz
syslog.3.gz
syslog.4.gz
syslog.5.gz
syslog.6.gz
syslog.7.gz
unattended-upgrades
user.log
user.log.1
user.log.2.gz
user.log.3.gz
user.log.4.gz
wtmp
wtmp.1
Xorg.0.log
Xorg.0.log.old
Xorg.1.log
Xorg.1.log.old
Xorg.pid-602.log.old
```

Fig: 1 List of linux log files

The above Figure 1 shows a list of log files that are commonly found inside Linux. All log files on linux servers are usually located under the directory / var / log. However these locations can vary depending on the server setup.

In a Linux based environment, there are four types of log files generated and they are: application logs, event logs, service logs, and system logs. Linux log files are created by processes in the system. Log files thus not have any specific limit. We can generate log files, reset log files, rotate log files and log files for archive. Locating the log files over vendors and daemons is very inconsistent. Currently found at different location such as /var/log, /var/adm or /usr / adm. Read device initialization scripts

such as `/etc/rc *`, `/etc/rc.d/ *`, or `/etc/Init.d/*.`. These commands can be used to see if logging is on, which file is being used and to find log files. In

`/etc/syslog.conf` the position of the log file is typically specified. Table 1 shows a short description of what kind of information each file contains.

Table 1 Log files and their explanations

Name of Log files	Description
<code>/var/log/auth.log</code>	All events related to authentication are logged here including successful and failed attempts.
<code>/var/log/boot.log</code>	The boot log stores all booting-related information.
<code>/var/log/kern.log</code>	It contains the kernel logged-in information. Helps to correct kernel errors and warnings.
<code>/var/log/daemon.log</code>	The daemon log contains information about Linux running events.
<code>/var/log/mail.log</code>	The mail log stores mail server information and the archiving of emails.
<code>/var/log/debug</code>	The debug log stores accurate debugging messages and is useful to troubleshoot particular system operations.
<code>/var/log/syslog</code>	It includes generic system activity logs and system messages that are not important.

An example of log file is system accounting file, which holds a record for each process such as username running the command, command name, used CPU time, process completion timestamp and completion status flag. Files not to be handled are the last login of the user's `/var / adm / lastlog` records and are in sparse file format which will expand alarmingly when copied. While accessing the latest log files are displayed as shown in Figure 2.



Fig : 2 Log files in file window format

In this Figure 2 few files have a lock and wrong symbol indicating those are system locked files which are not accessible for editing and that other files could be accessed by the user. Syslog, a comprehensive logging system used by the kernel and system utilities to manage the information generated. There are two essential functions, freeing programmers from write log files mechanics and allowing administrators to efficiently manage logging. Syslog allows messages to be sorted to log file, user terminals, or other machines by their important level and messages to be routed to. Syslog consists of three parts, the logging daemon and its configuration file, library routines used by programmers to send data to syslog and logger, which means a user-level command to send log entries.

Linux logs provide a visual history of everything that happens on the linux operating system. When something goes wrong, they will give us a helpful rundown of all incidents to help the admin locate the culprit. Linux logs files should be easy to find, as the text format is stored in the directory and subdirectory of `/ var / log`. It encompasses all kinds of framework, kernel, package manager, MySQL etc. There is still no analyzer in the system to evaluate an application's behavior, so the user can't evaluate or predict which application is better for the system and its data.

Log analysis is a commonly accepted term with a set of different log information not only about system vulnerabilities but also about system security. Because the data centers and networks generate a large number of logs, it is not easy for regular users to manually review the data. In this project, we introduce an application for reviewing and understanding the computer generated logs by normal user. It is achieved by providing the structured view of the application behavior and system and memory status.

2. RELATED WORK

(Phong H. Nguyen et al, 2019) has proposed an approach that involves innovative visual designs and techniques of interaction, along with a data mining algorithm to provide analysts with a multi-level analysis of user sessions and ways of performing in-depth multi-faceted comparative sessions of interest. To gain deep understanding of both anticipated and unpredictable user activities by observing their sequences of action. Also, they aim to boost the efficiency of our sequential mining algorithm to every production size without missing specific patterns. (Itkin et al, 2019) has focused on the User-assisted log analysis technology for quality assurance of fin-tech distributed applications. They have also presented the experience of a semi-automated study of the system's clearing and settlement actions by using its logs to define and recognize the errors. (Schipper Daan et al, 2019) has carried out state-of-the-art log parsing work in their logging environment and has assessed its accuracy and efficiency. This approach was used in industries and in implementations. There are logging applications in businesses, and tooling needs to adapt to it. Their implementation provides developers an simple way to add support for the various log libraries. This method is used to enhance our application log. (Nehal G. Karelia et al, 2014) has proposed a Web Usage Mining technique to analyze the behavior and activity of the user's results. The users' click streams are stored in Web log files. But since the data in the log files are not preprocessed, web-use mining techniques are used to pre-process the data and organize it into some organized information. To determine which user is visiting which sites and by which browser, the raw web log files generated by the web server need to be examined. The real raw web log file size has been reduced to 80 percent by applying the data cleaning technique and the accuracy of the data also improves. (Ke Yu et al, 2018) has focused on analyzing mobile user behavior and predicting it based on data from mobile Internet traffic. They create a bipartite network of User-Apps to reflect the pattern of traffic interaction between users and App servers. Knowing human behaviors using mobile phones is critical for developers of mobile systems, optimizing the human-centered system and delivering better service. So they suggest two optimistic and unlabeled (PU) learning methods, namely spy-based PU learning and K-means-based PU learning, to forecast Device use and to classify mobile video traffic. The proposed methods of PU prediction can increase both efficiency and accuracy with respect to F score and accuracy on classifiers.

(Kaikai Deng et al, 2019) has proposed a customized user behavioral analysis algorithm based on the regular pattern mining. To solve, the accuracy of the traditional algorithm for user identification being

unsatisfactory because it ignores the position of user-generated data in matching identity. They follow the subsequent weight allocation method of information entropy based on probability which improves the precision rate and recall rate compared with the method of empirical weight allocation. This proposed algorithm, which enhances the precision rate, the recall rate and the matching results of the user's F1. (Huiqi Zhang et al, 2011) has proposed a socioscope model based on cell phone call-detail data for social-network and human-behavior research and a new index to calculate the degree of reciprocity between users and their contact partners. They used multiple probability and statistical methods for quantifying social groups, relationships and patterns of communication to detect every attribute that arises when people engage in social behaviour. This model offers high precision. In their future studies, they expect to explore more information such as terminology, user experiences from other sources, other events, dynamics and evolution of social networks.

(Mallikarjunan K.N., et al, 2018) has discussed about Current protection mechanisms and concentrate on attack features, rather than the attacker's. Analysis of Attacker activity is a challenging problem, because relevant data can not be easily identified. So, the author uses cognitive analysis to identify the attacker group on the network traffic data logs and infer his actions and suggests a Fuzzy-rule-based method for categorizing the attacker. The model could be further expanded in many directions. Sophistication in modeling the actions of the intruder constitutes a promising area for future research. This involves building upon the current literature to incorporate new behavioral models with nine types of attackers into the existing model. After reviewing these research papers, all attempts are made to analyze user behavior, user performance on the web browser, attacker profiling, and to gather valid logs from raw log files. So they were all trying to know about the behavior of the user and to find out what they were doing. Some of the techniques suggested in the research papers are used in this proposed work to analyze the application behaviour.

3. TECHNIQUES

An application i.e. LOGxy is developed to identify and classify the system and user data. So multiple commands are used to get valid application information. Sqlite3 database is used to store these information into the database. This database is connected to the front end of the Qt creator application that was created to view the log information collected. This application which is useful for regular users will access the application logs in a standardized format. The process of log analysis is shown in the figure [3].

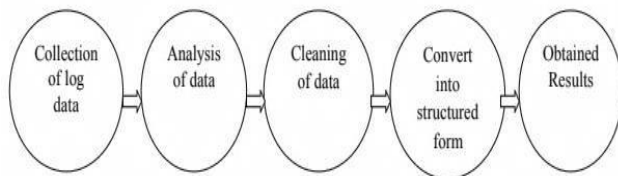


Fig: 3 Log analysis process

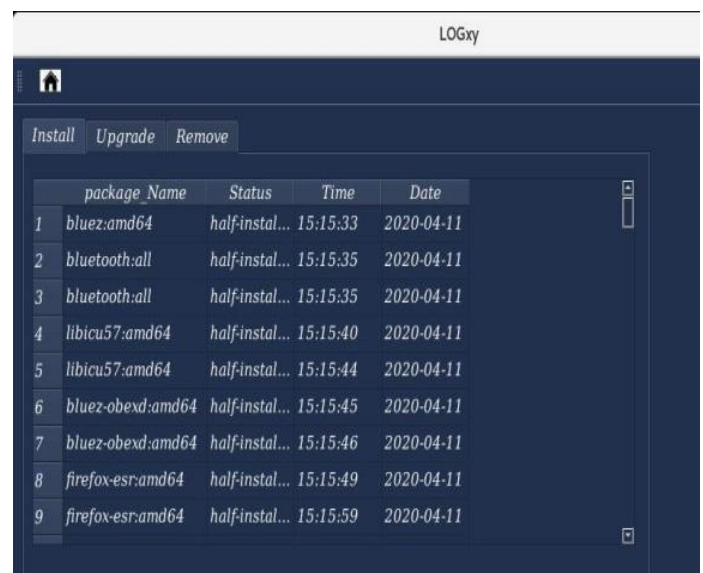
The user does not know which application is being installed, changed or removed, and the user can see the details about them using this app. This app is useful to easily figure out, any pending packages are not deleted from the system after the application has been removed. /var/log/dpkg will contain the collection of installed upgraded and deleted application of the current month. When you run out of space on your system, you may want to check who is the biggest contributor to get out of the problem. That can be looked at in two ways: Initially everybody wants to figure out the large data, it may be log data, etc and other choice is enabled size application / packages these problem is solved in this app. Foreground and background process, highest memory use process, newly installed, upgraded and uninstalled programs, port information opening and listening, and existing user information are all detected using this application. This is one way to identify the device that has been compromised, as hackers mostly come through the application. So the applications actions are evaluated using suitable commands.

In profiling application activity various commands are grouped together from different references to achieve the desired log profile. This app is implemented to save user time when searching on the internet for the commands Commands to include all relevant application information inside the system is used. For regular users, this app is useful to know the application information, memory details, port details, process details and device behavior. So a regular user can easily find out whether or not the device is secure. The application's output is achieved through the bash file commands. This application displays the log of the system's operations, as well as the processes list currently running in the system. Using this

LOGxy app, the regular user may recognize background processes and the Foreground processes.

4. RESULTS AND DISCUSSION

This application aim's to provide the user-friendly log analyzer for the user of the normal system. Using this program, the regular user can get system and application behavior information, so that the user can predict whether the application is secure and not secure. Here the LOGxy application is created using Qt creator, and sqlite3 is used to create the database. Bash file is the source to get the desired result, it contains multiple commands. The bash file is run using the application. This provides information about the application in the system. Some of the screenshots of the application was given below.



	package Name	Status	Time	Date
1	bluez:amd64	halfinstal...	15:15:33	2020-04-11
2	bluetooth:all	halfinstal...	15:15:35	2020-04-11
3	bluetooth:all	halfinstal...	15:15:35	2020-04-11
4	libc57:amd64	halfinstal...	15:15:40	2020-04-11
5	libc57:amd64	halfinstal...	15:15:44	2020-04-11
6	bluez-obexd:amd64	halfinstal...	15:15:45	2020-04-11
7	bluez-obexd:amd64	halfinstal...	15:15:46	2020-04-11
8	firefox-esr:amd64	halfinstal...	15:15:49	2020-04-11
9	firefox-esr:amd64	halfinstal...	15:15:59	2020-04-11

Fig: 4 Recently installed package details

After executing the **grep-i "install"/var / log / dpkg.log** command in bash file. It displays the recently instilled package name, package removed time and date, their current status as shown in figure 4. This details is used to identify the system configuration changes and installed Package status through the application.

package_Name	Status	Time	Date
1 bluetooth.all	upgrade	15:15:34	2020-04-11
2 libc6:amd64	upgrade	15:15:38	2020-04-11
3 bluez-usb:amd64	upgrade	15:15:45	2020-04-11
4 firefox-esr:amd64	upgrade	15:15:48	2020-04-11
5 libbluetooth3:amd64	upgrade	15:16:01	2020-04-11

Fig: 5 Recently updated package details

After executing the **grep-i "upgrade"/var / log dpkg.log** command in bash file. It displays the recently updated package name, the time and date of updation, their current status as shown in figure 5. This details is used to identify the last seven days updated package details.

package_Name	Status	Time	Date
1 gnome:amd64	remove	16:08:58	2020-04-12
2 task-gnome-des...	remove	16:09:00	2020-04-12
3 gnome-core:am...	remove	16:09:02	2020-04-12
4 chrome-gnome-...	remove	16:09:04	2020-04-12
5 gcj-6-jre-lib.all	remove	16:09:07	2020-04-12
6 gimp:amd64	remove	16:09:09	2020-04-12
7 gnome-control-...	remove	16:09:12	2020-04-12
8 gnome-tweak-t...	remove	16:09:15	2020-04-12
9 gvfs-backends:...	remove	16:09:18	2020-04-12
10 inkscape:amd64	remove	16:09:21	2020-04-12

Fig: 6 Recently removed package details

After executing the **grep-i "remove"/var / log / dpkg.log** command in bash file. It displays the recently removed package name, package removed time and date, their current status as shown in figure 6. This details is used to identify the attackers removed package details and removed package status through the applicant

Net_id	state	Recv_Q	send_Q	LocalAddress_and_port	peerAddress_and_port	User_Details	process_id	file_id
1	udp	UNCONN	0	0	*:1900	**	users:(("minissdpd"	pid=712 fd=4)
2	udp	UNCONN	0	0	*:631	**	users:(("cups-browsed"	pid=463 fd=7)
3	udp	UNCONN	0	0	*:56218	**	users:(("avahi-daemon"	pid=452 fd=14)
4	udp	UNCONN	0	0	*:5353	**	users:(("avahi-daemon"	pid=452 fd=12)
5	udp	UNCONN	0	0	:::44665	:::*	users:(("avahi-daemon"	pid=452 fd=15)
6	udp	UNCONN	0	0	:::5353	:::*	users:(("avahi-daemon"	pid=452 fd=13)
7	tcp	LISTEN	0	128	*:22	**	users:(("sshd"	pid=525 fd=3)
8	tcp	LISTEN	0	5	127.0.0.1:631	**	users:(("cupsd"	pid=453 fd=10)
9	tcp	LISTEN	0	128	:::22	:::*	users:(("sshd"	pid=525 fd=4)
10	tcp	LISTEN	0	5	:::1631	:::*	users:(("cupsd"	pid=453 fd=9)

Fig: 7 Opening ports information

After executing the **ss -tulpn** command in the bash file. It displays the opened ports details such as process id, Net id, port number, local and peer address as shown in figure 7.

user	process_id	CPU_usage	Memory_usage	Virtual_memory_size	Physical_memory_size	Connecting_terminal	Process_State_code	Command
1	root 1	0.0	0.1	204720	6996	?	Ss	20:17
2	root 2	0.0	0.0	0	0	?	S	20:17
3	root 3	0.0	0.0	0	0	?	S	20:17
4	root 5	0.0	0.0	0	0	?	S<	20:17
5	root 7	0.0	0.0	0	0	?	S	20:17
6	root 8	0.0	0.0	0	0	?	S	20:17
7	root 9	0.0	0.0	0	0	?	S	20:17
8	root 10	0.0	0.0	0	0	?	S<	20:17
9	root 11	0.0	0.0	0	0	?	S	20:17
10	root 12	0.0	0.0	0	0	?	S	20:17
11	root 13	0.0	0.0	0	0	?	S	20:17
12	root 14	0.0	0.0	0	0	?	S	20:17
13	root 15	0.0	0.0	0	0	?	S	20:17
14	root 16	0.0	0.0	0	0	?	S	20:17
15	root 18	0.0	0.0	0	0	?	S<	20:17
16	root 19	0.0	0.0	0	0	?	S	20:17

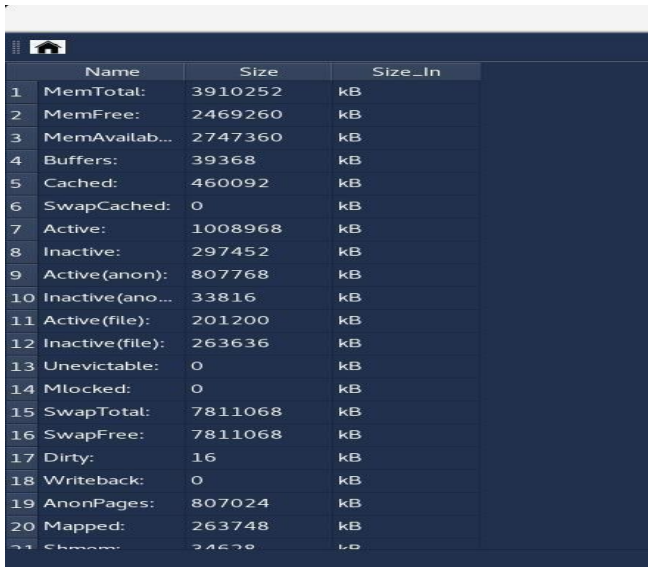
Fig: 8 Details of all the process in the system

After executing the **ps -aux** command in the bash file. It Displays the foreground and background process details such as Visual memory size, physical memory size, process state code, name of the terminal connected to the process as shown in figure 8.

Process_id	Parent_Process_id	Package_Name	Memory_usage	CPU_usage
1 864	768	/usr/bin/gnome-shell	7.4	8.2
2 633	604	/usr/bin/gnome-shell	4.1	0.2
3 757	755	/usr/lib/xorg/Xorg	1.8	1.2
4 1298	745	/usr/bin/gedit	1.3	0.7
5 991	768	/usr/lib/evolution/evolutio	1.3	0.0
6 1280	745	/usr/bin/nautilus	1.1	0.0
7 992	768	/usr/bin/gnome-software	1.1	0.0
8 594	592	/usr/lib/xorg/Xorg	1.0	0.0
9 943	768	/usr/lib/gnome-settings-dae	0.9	0.0

Fig: 9 Process which are using highest memory

After executing the `ps -eo pid,ppid,cmd,%mem,%cpu--sort=-%mem | head` command in the bash file. It displays the list of processes which use the highest system memory space, that process id, cpu usage, memory usage and process name as shown in figure 9.



	Name	Size	Size_In
1	MemTotal:	3910252	kB
2	MemFree:	2469260	kB
3	MemAvailab...	2747360	kB
4	Buffers:	39368	kB
5	Cached:	460092	kB
6	SwapCached:	0	kB
7	Active:	1008968	kB
8	Inactive:	297452	kB
9	Active(anon):	807768	kB
10	Inactive(ano...	33816	kB
11	Active(file):	201200	kB
12	Inactive(file):	263636	kB
13	Unevictable:	0	kB
14	Mlocked:	0	kB
15	SwapTotal:	7811068	kB
16	SwapFree:	7811068	kB
17	Dirty:	16	kB
18	Writeback:	0	kB
19	AnonPages:	807024	kB
20	Mapped:	263748	kB
21	Shmem:	24678	kB

Fig: 10: Total memory used for all application in the system

After executing the `cat /proc/meminfo` command in the bash file. It Displays the Memory Status for understanding the system's used and unused memory space as shown in figure 10.

5. CONCLUSION AND FUTURE WORK

In this application, a log file analysis is implemented and evaluating system configuration details in the standardized format is achieved. So that the usual user can recognize each application's actions in the system. This identifies each application's behavior in the system and offers statistical details about the system's applications. Using this application, normal user can identify the security status of the installed application. It also helps in identifying the system ports that are been used, the application using those ports, list of current open ports and their process id mapping. This mapping aids the user to identify unused and suspicious ports. Therefore this application could be used as a decision-maker. It gives more information while using commands in the terminal. But the result after all the commands have been aggregated is slightly different from the terminal output. It truncates some data which could be used for better analysis. Future work focuses on overcoming these limitations and identifying all processes and sub processes initiated in the system to have better status analysis using this application.

6. REFERENCES

[1] Nguyen, P.H., Turkey, C., Andrienko, G., Andrienko, N., Thonnard, O. and Zouaoui, J., 2018. Understanding user behaviour through action sequences: from the usual to the unusual. *IEEE transactions on visualization and computer graphics*, 25(9), pp.2838-2852.

[2] Itkin, I., Gromova, A., Sitnikov, A., Legchikov, D., Tsymbalov, E., Yavorskiy, R., Novikov, A. and Rudakov, K., 2019, April. User-assisted log analysis for quality control of distributed fintech applications. In *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)* (pp. 45-51). IEEE.

[3] Schipper, D., Aniche, M. and van Deursen, A., 2019, May. Tracing back log data to its log statement: from research to practice. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)* (pp. 545-549). IEEE.

[4] Karelia, N.G. and Shukla, S., 2014. Data Preprocessing: A Pre requisite for Web Log Files. *International Journal of Engineering Research & Technology (IJERT)*.

Yu, K., Liu, Y., Qing, L., Wang, B. and Cheng, Y., 2018. ositive and unlabeled learning for user behavior analysis based on mobile internet traffic data. *IEEE Access*, 6, pp.37568-37580.

[5] Deng, K., Xing, L., Zheng, L., Wu, H., Xie, P. and Gao, F., 2019. A user identification algorithm based on user behavior analysis in social networks. *IEEE Access*, 7, pp.47114-47123.

[6] Zhang, H., Dantu, R. and Cangussu, J.W., 2011. Socioscope: Human relationship and behavior analysis in social networks. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6), pp.1122-1143.

[7] Mallikarjunan, K.N., Shalinie, S.M. and Preetha, G., 2018. Real Time Attacker Behavior Pattern Discovery and Profiling Using Fuzzy Rules. *Journal of Internet Technology*, 19(5), pp.1567-1575.

[8] Dai, H., Li, H., Shang, W., Chen, T.H. and Chen, C.S., 2020. Logram: Efficient Log Parsing Using n-Gram Dictionaries. *arXiv preprint arXiv:2001.03038*.

[9] Diederichsen, L., Choo, K.K.R. and Le-Khac, N.A., 2019, December. A Graph Database-Based Approach to Analyze Network Log Files. In *International Conference on Network and System Security* (pp. 53-73). Springer, Cham.

[10] El Hadj, M.A., Khoumsi, A., Benkaouz, Y. and Erradi, M., 2019, June. Efficient Security Policy Management Using

Suspicious Rules Through Access Log Analysis. In International Conference on Networked Systems (pp. 250-266). Springer, Cham.

[11] Guo, S., Liu, Z., Chen, W. and Li, T., 2018, June. Event extraction from streaming system logs. In International Conference on Information Science and Applications (pp. 465-474). Springer, Singapore.

[12] Hanka, S., 2019. A Grammar Based Approach to Distributed Systems Fault Diagnosis Using Log Files.

[13] Hadi, A.S. and Ali, S.H., 2019. Resource Description Framework Representation for Transaction Log File. Journal of Computational and Theoretical Nanoscience, 16(3), pp.1093-1099.

[14] He, P., Zhu, J., He, S., Li, J. and Lyu, M.R., 2017. Towards automated log parsing for large-scale log data analysis. IEEE Transactions on Dependable and Secure Computing, 15(6), pp.931-944.

[15] Lin, Q., Zhang, H., Lou, J.G., Zhang, Y. and Chen, X., 2016, May. Log clustering based problem identification for online service systems. In 2016 IEEE/ACM 38th International Conference on Software Engineering Engineering Companion (ICSE-C) (pp. 102-111). IEEE.

[16] Nagaraj, K., Killian, C. and Neville, J., 2012. Structured comparative analysis of systems logs to diagnose performance problems. In Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12) (pp. 353-366).

[17] Pickard, J.J. and Foster, W.W., Red Hat Inc, 2019. Managing and archiving system and application log files. U.S. Patent 10,318,477.

[18] Vedaprakash, M.P., Prakash, M.P.O. and Navaneethakrishnan, M.M., 2016. Analyzing The User Navigation Pattern From Weblogs Using Data Pre-Processing Technique. International Journal of Computer Science and Mobile Computing, ISSN, pp.90-9.