IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 5, Ver. I (Sep - Oct 2018), PP 15-22
www.iosrjournals.org

Log Based Intrusion Detection System

Umesh K. Raut[1]
[1](*Computer Engineering, MIT Pune, India)*

Abstract: *The idea of making everything readily available and universally has led to a revolution in the field of networks. In spite of the tremendous growth of technologies in the field of networks and information, we still lack in preventing our resources from cyber-attacks. This may not concern small organizations but it is a serious issue as far as industries, companies or national securities are concerned. Since many different mechanisms were opted by organizations in the form of intrusion detection and prevention systems to protect themselves from these kinds of attacks, there are many security breaches which go undetected. A host-based intrusion detection system (HIDS) is a system that monitors a computer system on which it is installed to detect an intrusion and/or misuse, and responds by logging the activity and notifying the designated authority. In this paper, we develop a HIDS using logs generated by services running on the systems. We will discuss about the client-server architecture used in HIDS.*

Index Terms: *IDS, HIDS, security, threat detection, log analysis*

Date of Submission: 29-08-2018 Date of acceptance: 13-09-2018

I. Introduction

A log file is a file that records either events that occur in an operating system or other applications or it is a message between different users of a communication software. Almost every software running on the system generates logs, mostly for debugging purposes. A log is generated whenever an activity occurs in the software. These logs can be used for different purposes. The logs can be used as a forensic evidence, audit trials or to understand how the system behave after certain input. Recognizing the importance of logs, the National Institute of Standards and Technology, USA issued best practices and recommendations for computer security log management [2]. Current generation log analyzers work well with logs from know applications. They are able to parse and interpret logs from know formats but fail to parse logs from a new application or a different and unknown source. It is important to parse logs appropriately in order to use them later.

Log files provide vital information about systems and the activities occurring on them. For instance, database logs can help trace how data was added and modified, while web server logs reveal patterns of how resources are accessed from the Web. Analyzing log files can provide valuable insights into system configurations as well as potential vulnerabilities. It is possible to use this information pro-actively to secure a system against potential malicious activities.

Large company networks consists of a variety of software, including security software, running on multiple hardware devices generating multiple log files. The logs generated daily by these devices is massive. Traditional log file analyzers rely on knowing the format of the file. Existing tools either focus on a specific type of log file or several but a fixed set of log types. Log files typically lack a common format with each log encoding data in its unique and often proprietary style. It is even more difficult to keep track of log file formats in large heterogeneous networks in which software and devices are dynamic in nature; often new software and hardware are added to the network, each generating a log file with a unique schema. Existing approaches designed to deal with a fixed number of log files thus fail to scale in such scenarios.

Recently cyber-attacks have been fairly sophisticated affecting different parts and applications as well as several systems simultaneously and are carried out over a period of several days. Detecting and preventing such low and slow vector attacks would require to generate a holistic view rather than analyzing each log individually. Threat detection in dynamically adapting scenarios requires integrating data from multiple traditional and non-traditional sources.

```
127.0.0.1 - - [19/Nov/2017 18:44:02] "GET /index.php HTTP/1.1" 404 -
127.0.0.1 - - [19/Nov/2017 18:44:27] "GET /image HTTP/1.1" 404 -
127.0.0.1 - - [19/Nov/2017 18:44:46] "GET /index.html HTTP/1.1" 404 -
127.0.0.1 - - [19/Nov/2017 18:44:57] "GET /favicon.ico HTTP/1.1" 404 -
```

Figure 1: Sample web logs

We present an approach that infers the structure and the schema of the log file and uses the schema to generate semantic representation of the log file content. Consider the example web server log in figure 1. Each line in the log file records the same type information for a fixed set of fields. In our example, each line is recording IP address of the requesting host, time of request, requested resource, etc. We use this information by splitting it into distinct columns, with each column consisting of similar type of values. For example, the web server log can be split into columns such as requesting IP address, requested resource and so on.

Organizations are facing an increasing number of threats everyday in the form of viruses, malware, intrusions, etc. Since many different mechanisms were opted by organizations in the form of intrusion detection and prevention systems to protect themselves from these kinds of attacks, there are many security breaches which go undetected.

A HIDS is to prevent the host system from being com-promised by intruders. To prevent the execution of malicious codes on the host, HIDS monitors the system audit and event logs.

A HIDS is used for security detection, visibility, and compliance monitoring. It's based on a multi-platform agent that forwards system data (e.g log messages, file hashes, and detected anomalies) to a central manager, where it is further analyzed and processed, resulting in security alerts. Agents convey event data to the central manager via a secure and authenticated channel.

II. Literature Survey

A literature survey represents a study of previously existing material on the topic of the paper.

- Existing theories about the topic which are accepted universally.
- Books written on the topic, both generic and specific.
- Research done in the field usually in the order of oldest to latest.
- Challenges being faced and ongoing work, if available.

The literature survey should be structured in such a way as to logically represent the development of ideas in that field.

[1] In this paper, the author describes the logs generated by intrusion detection systems, web servers, anti-virus and anti-malware systems, firewalls and network devices are usually intended for diagnostic and de-bugging purposes, but their data can be extremely useful in system audits and forensic investigations.

[2] This paper describes a framework that can process any log file with unknown source and format, and automatically generate a semantic interpretation of both its schema and contents in the form of RDF linked data triples. They develop a splitting algorithm which can successfully generate a table like structure for log files. They also extended existing techniques to successfully annotate every column in a log file with a semantic class and map selected pairs of column to relation from the given ontology.

[3] Joshi describe an automatic framework that generates and publishes a RDF linked data representation of cyber-security concepts and vulnerability descriptions extracted from several structured vulnerability databases and unstructured text. This linked data collection was intended for vulnerability identification and to support mitigation efforts. The prime idea was to leverage the Linked Data representation of vulnerability databases to detect and prevent potential zero day attacks.

[4] In the area of "Log analysis", Nascimento use ontologies to analyze security logs. They create an ontology model from the Modsecurity logs and demonstrated that it was easier to find co-relations of events in log files when they were modelled using their ontology.

[6] Splunk is a enterprise log analysis and management tool developed to analyze a large corpus of logs of logs providing features such as searching, management, storage and visualization. It analyzes structured and unstructured log files and identifies fields in a given log file. It works well with log files whose structure it already knows. It is able to split a log file into fields, but fails to generate headers for log files from unknown sources.

[9] To prevent the execution of malicious codes on the host, HIDS monitors the system audit and event logs. But the design of HIDS is very challenging due to the presence of high false alarm rate. This paper mainly focuses on reducing the problem of false alarm rate, using semantic based system call patterns. This paper make use of the semantic approach to apply on the underlying kernel level system calls which can help understand the anomaly behavior. In this concept, the semantic approach is applied on the system call patterns for detecting the intrusion on the host system using ADFA-LD dataset. The decision engine used is the ELM, which is well known for its high learning speed and it requires only one time training. But the cost of processing time is high.

[10] In this paper they have made a survey on the overall progress of intrusion detection systems. They had done survey of the existing types, techniques and architectures of Intrusion Detection Systems in the literature. This survey paper gives a description of some intrusion detection approaches based on two basic techniques. Some approaches work better in one environment but then prove to be weak in other environments. The approaches discussed in this paper includes Statistical Models, Data Mining Based Methods, Signature analysis, Rule based systems, Genetic Algorithms etc.

[11] Host Intrusion Detection Systems (HIDS) have recently gained a noticeable amount of interest. These defensive systems detect malicious activities on host-based applications. This paper reviews types of architecture in intrusion detection systems and describes a threat-aware HIDS architecture model. In this paper, they have reviewed traditional Intrusion De-tection architectures and then proposed an HIDS architecture model. Some of the significant components in this model are: Dispatcher; according to its policy distributes the input traffic to the analyzers. This, could affect on detection time and accuracy. Equalizer; necessitates in host web application data, which supports data normalization. Correlation engine; responsible to reduce the total number of alerts and messages that need to be observed by the system administrator to as few as possible by merging similar events into groups.

[12] Most existing solutions for detecting these attacks use log analysis, and employ either pattern matching or machine learning methods. This paper proposes a multi-stage log analy-sis architecture, which combines both pattern matching and su-pervised machine learning methods. It uses logs generated by the application during attacks to effectively detect attacks and to help preventing future attacks. The architecture is described in detail; a proof-of-concept prototype is implemented and hosted on Amazon AWS, using Kibana for pattern matching and Bayes Net for machine learning. Experiment results have shown that the two-stage system is able to detect significantly more SQL injections that a single-stage system. When Bayes Net model (a supervised machine learning method) precedes Kibana (a pattern matching sys-tem), the combined system has achieved the best result. Since Kibana provides the final output with visualization, it is easy for analysts to further understand, interpret, and take further actions.

[13] Intrusion Detection System (IDS) and Intrusion Pre-vention System (IPS) are the standard measures to secure computing resources mostly in a network. They are deployed in a network for assuring an intrusion free computing envi-ronment. This paper discusses the two technologies in details, their functionality, their performances and their effectiveness to stop the malicious activity over a computer network. Having IPS and IDS technologies are only two of many resources that can be deployed to increase visibility and control within a corporate computing environment. IDS and IPS are to provide a foundation of technology that meets the requirement of tracking, identifying network attacks to which detect through logs of IDS systems and prevent an action through IPS systems. If the host is with critical systems, confidential data and strict compliance regulations, then its a great to use IDS, IPS or both in network environments.

[14] The human labeling of the accessible network audit information instances is generally tedious, expensive as well as time consuming. This paper focuses on study of existing intrusion detection task by using data mining techniques and discussing on various issues in existing intrusion detection system (IDS) based on data mining techniques. The data mining techniques that were discussed in this paper are K-Means, ID3 (Iterative Dichotomiser 3), Naive Bayes, K-NN (K-Nearest Neighbour).

III. Intrusion Detection System

In today's booming e-commerce economy age, virtually every business is connected to compete for market share in the cyberspace. Enterprise's networked systems are inevitably exposed to the increasing threats from the external hackers as well as from internal. The consequences can be loss or modification of critical business data, disruption of services (availability), compromise of proprietary business plans or processes (confidentiality and integrity).

To counter these threats, information security organization has deployed many methods, tools and technology to defend the legitimacy of the systems. Methods like implementing policies and procedure, user awareness, deploying firewall and authentication systems, control systems access contain, eradicate recover and serve as a lesson learnt.

We examine IDS, one of the relative new technologies in information security. It aims to explore, in high level, the intrusion detections systems available today, as well as new developments in the technology.

At its core, IDS for computer network systems resemble burglar alarm systems to a physical building, it is capable of detecting and alerting the systems administrator on potential intrusion, providing guidance against any potential loss of integrity and confidentiality to the enterprise's valuable intellectual assets.

Firewalls and authentication are effective in protecting and preventing unauthorized access to the systems but lacks capabilities to monitor the network traffic where majority of attacks are taking place. These attacks could be initiated by disgruntled employees and others who have legitimate network access and use that privilege to do harm.

Firewall and authentication systems are vital, but they work at the point of entry to the network, if an attack able to breach the firewall, he can roam freely throughout the whole network. To keep a constant eye on network traffic and to know anything out of ordinary is happening, network security should be supplemented with IDS.

IDS can be active or a passive IDS. Using an active IDS, suspected attacks are automatically blocked, based on pre-programmed rules. This type of IDS is also referred to as intrusion detection and prevention

system (IDPS). IDPS offers real-time protection whereas passive IDS only monitors the activities, logs the suspected activities and reports it to the administrator for action.

These are the two ways intrusion detection works. IDS is host based, network based or the hybrids of the two. Each type of the intrusion detection systems has its own merits and legitimate shortcoming. Regardless of the type of systems deployed, it should include the following key features:

- **Robust:** IDS is expected to run continually in the background without human intervention, it should also be fault tolerant, meaning that in the event of a crash or failure the product won't have to be rebuilt or reconfigured. It should remain impervious to attacks.
- **Flexibility and Scalability:** IDS should be configurable and is flexible in response to changes on the network environment, it should also be able to cope with the growth of the network traffic while maintaining fairly high accuracy in performance (scalable).
- **Easy of use:** IDS is to be managed without consuming too much of bandwidth or high overhead from the organization. However, balance should be sought between ease of use and system effectiveness. In a real situation, it requires considerable resources to manage and operate the devices.

A. Host-based IDS:

Host based IDS typically reside in the hosts they are monitored, the system agent will record important system file attributes, including hashes of the files. The agent will periodically scan log files for anomalous activity and notify the system administrator if they have detected suspicious pattern of systems access on the hosts. Another host-based approach monitors all packets as they enter and exit the host, just like a personal firewall.

Host based IDS is popular in that it can tell you if an attack actually happened and if it is significant enough to warrant action. For example the IDS can detect changes in system files and knows if someone tries to install potentially malicious software such as back doors. Back doors are highly specialized programs that let hackers remotely control a server and either steal information from or modify it in a harmful way.

Also, the host is the best location to any attacks. It can also get a fine granularity of information, such as who is accessing what files and when the users log in and out of servers. It is appropriate for protecting and individual computer systems and the information it contains. However, it doesn't provide data on the network as a whole. Also the security systems take on considerable processing resource of the host (CPU, RAM and storage).

B. Network-based IDS:

As its name suggests, network-based IDS monitor and analyze network traffics on a designated segment. Network based IDS can be categorized as knowledge or behavior based.

For knowledge based NIDS, the systems searches for know "attack signatures" that indicates the packets represent in intrusion. Signatures could be based on actual packet contents, and are checked by comparing the bits against known patterns of attacks, for example, attempts to modify systems files. Other known network attacks are protocol based where attackers seek weaknesses of a poorly administered web, file or other servers in a network. These port attack signatures will monitor and identify attempts to connect to network ports associated with service that are often vulnerable. Another protocol signature the systems monitors is the abnormal or illogical TCP/IP packet headers, which is identified as denial-of-service pattern of attacks.

Behavior based NIDS identify attacks by monitoring systems or network traffic patterns and flagging any activity that looks suspicious. The systems capture and analyze packets to define patterns of usage on the network. Once the IDS have constructed statistics or the traffic patterns, it will audit network traffic and analyzing them for any abnormality of the traffic pattern, which is deviated from the normal statistic.

Network based IDS provides real time monitoring and provides faster turn around time in detecting and responding to potential attacks. Also, it is a system of probes that can be deployed at different points of the network which are deemed critical and is prone for attack so that it can capture those attacks packets at early stage, for example, these monitors can be located at a gateway or firewall between a corporate intranet and the outside internet (known as router based monitoring) or inside the intranet and between the dedicated server farm segment with other sub-segments (known as network based monitoring).

IV. Approach

Our approach implements security log analysis. It stores the alerts, and not every single log. Though, it is recommended for log management and long term storage of ALL logs. Security Log Analysis can be called as Log-based IDS (LIDS)

Log-based IDS (LIDS):

Log Analysis for intrusion detection is the process or techniques used to detect attacks on a specific environment using logs as the primary source of information.

LIDS is also used to detect computer misuse, policy viola-tions and other forms of inappropriate activities.
1) Benefits of LIDS:

- Cheap to implement: Does not require expensive hardware
- High visibility of encrypted protocols: SSHD and SSL traffic are good examples
- Visibility of system activity (kernel, internal daemons)
- Every application/system can be a part of it:

– They all have some kind of logs.
– Including firewalls, routers, web servers, applica-tions, etc.

V. Architecture

The HIDS can be used in two ways:

- Local (Figure 2)
  – When there is just one system to monitor.
- Client/Server (Figure 3)
  – When there are more than one machines connected.
  – The client/server architecture is recommended.

Various daemons are used throughout the analysis of threat detection. Some of the important daemons would be:

- Analysisd - Used for log analysis.
- Remoted - Receives remote logs from agents. Maild - Used to send email alerts.
- Logcollector - Reads log files (syslog, apache logs, sshd logs, etc)
- Monitord - Monitors agent status, compresses and signs log files, etc.
- Agentd - This daemon is would run at the agent side.

A. Local Architecture

Generic log analysis flow breakdown (for local architecture)

- Log collecting is done by logcollector.
- Analysis and decoding are done by analysisd. Alerting is done by maild.
- Active responses are done by execd.
- Active responses are done by execd.

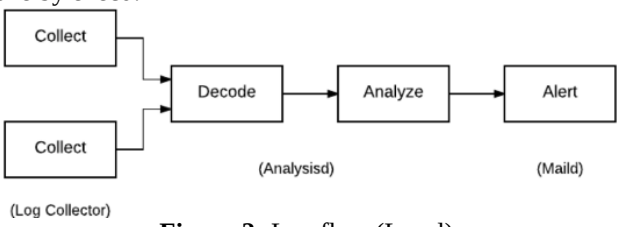

Figure 2: Log flow (Local)

B. Agent/Server Architecture

Generic log analysis flow for agent/server architecture

- Log collecting is done by logcollector.
- Analysis and decoding are done by analysisd.
- Alerting is done by maild.
- Active responses are done by execd

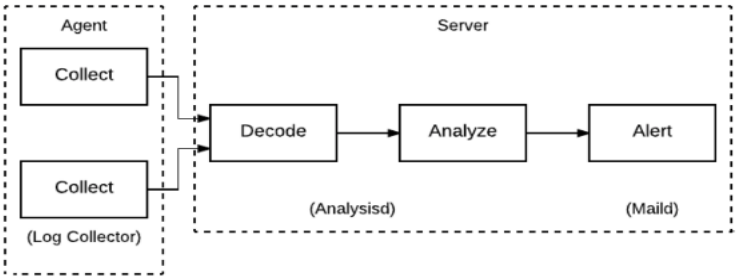

Figure 3: Log flow (agent/server)

C. Network Communication

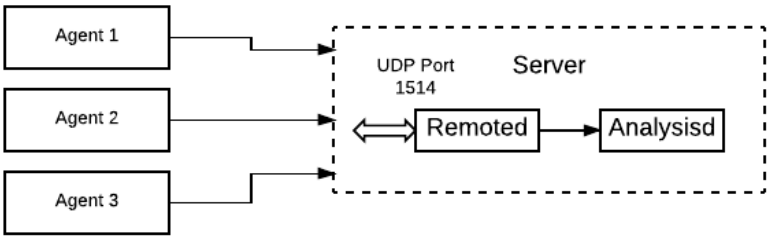

Figure 4: Agent/Server Network communication

Figure 4 shows the communication between agent and server. The data transmitted is compressed and sent to the server. The transmitted data is encrypted using pre-shared keys with Blowfish encryption algorithm [6]. The data is transmitted via UDP port 1514.

VI. Log Analysis

In this section, we'll discuss about how the logs are analysed in the HIDS. All the logs are analysed by the daemon named analysisd. This daemon does the log decoding and analysis.
The log flow inside analysisd comprises of 3 main parts:
1) Pre-decoding - Extracts known fields, like time, etc.
2) Decoding - Using user-defined expressions
3) Signatures - Using user-defined rules

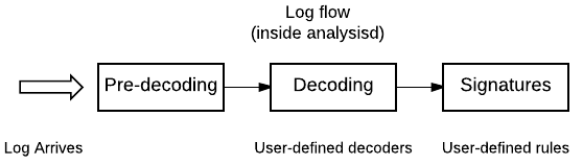

Figure 5: Internal log flow

A. Log pre-decoding

Log pre-decoding extracts generic information from the logs, such as hostname, program name and time from syslog header. These logs must be well formatted.
For example, log comes in as:

Apr 13 13:00:01 enigma syslogd: restart

After parsing this log, the HIDS will store it as: time/date -> Apr 13 13:00:01
hostname -> enigma program name -> syslogd log -> restart

Let us take an example of a SSHD message:
Log comes in as:

Apr 14 17:32:06 enigma sshd[1025]: Accepted password for root from 192.168.2.190 port 1618 ssh2
This is how it will look like after pre-decoding: time/date -> Apr 14 17:32:06
hostname -> enigma program name -> sshd
log -> Accepted password for root from port 192.168.2.190 port . . .

B. Log Decoding

Log decoding is the process to identify key information from logs. Most of the time you don't need to worry about it. Generally, we want to extract source ip, user name, id, etc. The list is user-defined and stored in a file named decoders.xml.
Let's take a look at how the log will look like after being decoded:
Original log from SSHD:
Apr 14 17:32:06 enigma sshd[1025]: Accepted password for root from 192.168.2.190 port 1618 ssh2
Log after being decoded: time/date -> Apr 14 17:32:06 hostname -> enigma program name -> sshd
log -> Accepted password for root from 192.168.2.190 port . . .
srcip -> 192.168.2.190 user -> root

C. Log Rules

Next step after decoding is to check the rules. The rules are internally stored in a tree structure. The rules are user-defined in an XML file which is very easy to write. It allows match, based on decoded information. The rules are independent of the initial log format, because of the decoders.

The rules are of 2 types:

1) Atomic - Based on a single event.
2) Composite - based on patterns across multiple logs.

To write a rule, we need the following attributes:
A Rule ID (any integer)
A Level - from 0 (lowest) to 15 (highest) Level 0 is ignored, not alerted at all
Pattern - anything from ”regex” to ”srcip”, ”id”, ”user”, etc
The following rule is written for the sshd: <rule id = ”111” level = ”5”> <decoded as>sshd</decoded as>
<description>Logging every decoded sshd message</description>
</rule>
This is the second rule written which is dependent on the first rule.
The severity is high (level 7)
It will only execute if the first one matches (if sid)
Match is a simple pattern matching (looking for Failed pass)

<rule id = ”111” level = ”5”> <decoded as>sshd</decoded as>
<description>Logging every decoded sshd message</description>
</rule>
<rule id = ”122” level = ”7”> <if sid>111</if sid> <match>^Failed password</match>
<description>Failed password attempt</description> </rule>

D. D. Rule Structure

The rules are internally stored in a tree structure using an user-defined XML format. It is very easy to write rules in XML format. It allows to match, based on the decoded information. The advantage of using these rules is, that it is independent of the initial log format, because we use decoders to parse the logs and store it in a format which is consistent. Figure 6 shows structure after first five rules. Not a flat format (like most log analysis tools)

The rules are matched very fast. Non-sshd messages are only checked against the first rule(111), not the sub ones.

Average of only 7/8 rules per log, instead of 400 default rules.

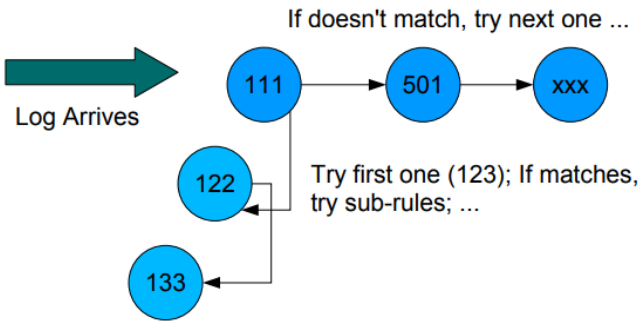

Figure 6: Rule structure

VII. Hardware and Software Requirements

Hardware and software requirements are very minimal.

A. Hardware requirements

Hardware requirements include: A computer with an operating system installed in it. RAM with more than 2GB storage.

B. Software requirements

Software requirements include: GCC compiler for compiling the source code.

The HIDS server installed at the server side. The HIDS agent installed at the client/agent side. The HIDS server should be installed on a Linux operating system. The HIDS client/agent can be installed on either Linux or Windows operating system depending on the systems to be monitored.

VIII. Conclusion

The logs generated on any system play a crucial role. The system to be developed makes use of the logs generated by various processes and services. The logs are monitored and analyzed and any suspicious activity that results in the alteration of the logs, is reported in real time.

The project aims at developing an effective and efficient HIDS using log analysis. Light-weight processes known as Agents can be installed on the host machines which will run as a service in the background. The Agents in turn communicate with the server and send the necessary data from the host.

The system focuses on logs predominantly, along with a rule-set of actions. It also consist of a web-panel which provide an easy to use interface and an overall view of the hosts. The web panel can be used to access the server remotely and monitor the system.

References

[1]. Piyush Nimbalkar, Varish Mulwad, Nikhil Puranik, Anupam Joshi and Tim Finin, ”Semantic Interpretation of Structured Log Files”, 2016 IEEE 17th International Conference on Information Reuse and Integration.

[2]. K. Kent and M. Souppaya, ”Guide to computer security log management recommendations of the national institute of standards and technology” NIST, 2006.

[3]. A. Joshi, R. Lal, T. Finin, and A. Joshi, ”Extracting cyber-security related linked data from text” in Seventh International Conference on Semantic Computing (ICSC). IEEE, 2013,pp. 252259.

[4]. C. H. do Nascimento, R. E. Assad, B. F. Loscio, and S. R. L. Meira, ”Ontology: A security log analyses tool using web semantic and ontology,” in 2nd OWASP Ibero-American Web Applications Security Conference, 2010, pp. 112.

[5]. Sreenivas Sremath Tirumala, Hira Sathu, Abdolhossein Sarrafzadeh, ”Free and open source Intrusion Detection Systems: A study”, Interna-tional Conference on Machine Learning and Cybernetics, Guangzhou, 12-15 July 2015.

[6]. ”Splunk”, http://www.splunk.com

[7]. ”Blowfish Cipher”, https://en.wikipedia.org/wiki/Blowfish (cipher).

[8]. Ho, Swee Yenn (George), ”Intrusion Detection-Systems for today and tomorrow”, SANS Institute InfoSec Reading Room.

[9]. M.Anandapriya, Mr.B.Lakshmanan, ”Anomaly Based Host Intrusion Detection System Using Semantic Based System Call Patterns”, IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO)2015

[10]. Uzair Bashir, Manzoor Chachoo, ”Intrusion Detection and Prevention System: Challenges & Opportunities”, IEEE 2014

[11]. Atefeh Torkaman, Marjan Bahrololum, M.H.Tadayon, ”A Threat-aware Host Intrusion Detection System Architecture Model”, 2014 7th Inter-national Symposium on Telecommunications (IST’2014)

[12]. Melody Moh, Santhosh Pininti, Sindhusha Doddapaneni, Teng-Sheng Moh ,”Detecting Web Attacks Using Multi-Stage Log Analysis”, 2016 IEEE 6th International Conference on Advanced Computing

[13]. Nilotpal Chakraborty, ”Intrusion Detection System and Intrusion Preven-tion System: A comparative study”, International Journal of Computing and Business Research (IJCBR), Volume 4 Issue 2 May 2013

[14]. Sanjay Sharma and R. K. Gupta, ”Intrusion Detection System: A Review”, International Journal of Security and Its Applications, Vol. 9, No. 5 (2015), pp. 69-76

Umesh K. Raut "Log Based Intrusion Detection System " IOSR Journal of Computer Engineering (IOSR-JCE) 20.5 (2018): 15-22.