International Journal of Database Theory and Application
Vol. 10, No.11 (2017), pp. 27-46
hhtp://dx.doi.org/10.14257/ijdta.2017.10.11.03

Scalable Security Analytics Framework Using NoSQL Database

Rizwan ur Rahman and Deepak Singh Tomar

Maulana Azad National Institute of Technology, India
rizwan.rahman12@gmail.com,deepaktomar@manit.ac.in

Abstract

Enterprises generate an estimated ten to hundred billion events every day. Large enterprises collect over 500GB logs per day. Traditional systems are not capable to handle this massive amount of data and this becoming classic problem of Big Data. Security Analytics deals with these issues by utilizing the techniques from Big Data analytics to dig out valuable information for averting cyber attacks. In this paper the scalable framework for security analytics is proposed using MongoDB NoSQL database. An attack scenario is created to simulate the zero-day malware. Supervised and unsupervised learning techniques are applied for analytics on data collected from live application and experimental set-up. The outcome is 360^0 view of data by singling out an abnormal access behavior for given user. It is observed that False Positive rate has been reduced.

Keywords*: Big Data analytics, Security analytics, NoSQL, MongoDB, Cyber Security, Real-Time Analytics, Anomaly Detection, Machine Learning, Clustering, and Classification*

1. Introduction

Software organizations regularly collect a variety of multiple terabytes of security-related data. The data comes from a variety of sources such as application access logs, network logs, database logs, web access logs, firewall logs, and user's action events. Software organizations generate an estimated 40GB of logs per day. These records will increase as organization enables monitoring and surveillance in additional sources, install more security devices and increase number of employees. The variety of data generating with high velocity from these sources, rapidly become overwhelming. Conventional analytical and mining techniques do not work efficiently on large scales of security data and typically produce a lot of false positives [1].

On the other hand, Enterprises will continue to implement defensive methods such as Hardware and Software Firewalls, Intrusion Detection System, Scanning for well-known malicious programs, patching application vulnerabilities and software updating. However, using these conventional defensive methods alone is not sufficient in the cyber world. To enhance the security practices, the software enterprises has to additionally employ security analytics techniques for constant monitoring and examining for new threat detection and response. The massive volume of numerous types of security-related data generating with high velocity involving storage, the processing and analysis; security analytics quickly becoming a typical Big Data problem. The first problem with traditional analytics system is the limited processing speed, the data in the real world is generating at an exponential rate. The other is the variety of data such as semi-structured and unstructured data, in traditional relational databases the data is highly structured in tabular form, which follows the strict schema, so it is not able to handle unstructured and semi-structured data. Moreover, the conventional relational database systems are not

Received (June 10, 2017), Review Result (October 30, 2017), Accepted (November 20, 2017)

ISSN: 2005-4270 IJDTA

horizontally scalable. To overcome this, a notion of NoSQL database has emerged, which can handle a variety of data and NoSQL databases are horizontally scalable [2].

Hence, in this paper, scalable security analytics framework using NoSQL database is proposed, which can handle a variety of semi-structured and unstructured data in real-time built on top of NoSQL MongoDB document-oriented database. The gist of major contribution of this paper is as follows:

- Scalable Analytics System is proposed using MongoDB document-oriented database.
- A variety of semi-structured data is collected and generated from live web application and an Experimental environment.
- Data is adding in the horizontally scalable system in real time using an open source tool Fluentd.
- An attack scenario is developed and deployed to simulate the zero-day attack.
- Supervised and unsupervised learning are applied to classify and cluster the normal and attack traffic.
- Evaluate and validate the prototype system using different validation schemes.

The rest of chapter is organized as follows: Second section review the related work and background, third section introduces big data analytics and NoSQL, the fourth section explores the document-oriented database MongoDB, the fifth section presents the proposed system design and prototype implementation, the sixth section illustrates the experiments and results. Finally, the seventh section concludes the paper.

2. Related Work

In the last decade, numerous researchers have proposed solutions for mitigating security threats based on machine learning techniques. A lot of packet processing tools are available for web traffic analysis such as Wireshark, Nmap, netsniff-ng, TCP dump and Wireshark, and SNORT [3]. In general, these packet-processing tools have limitations that they run on a single machine with a limited amount of storage and computing resources. Furthermore, with a single machine, it is hard to give fault-tolerant analysis services against a node failure, which over and over again happens when read/write jobs are repeatedly performed on disks. Researchers are steadily shifting from SIEM based methods to Big Data approaches like in [4] developing DDoS attack detection system based on Hadoop using large network traffic for mitigation of security threats. In [5], authors devise the first packet processing technique for Hadoop that examines packet trace files using map reduce by analyzing packets across multiple HDFS blocks. Researchers [6] report their work on developing a novel traffic monitoring system that carries outflow analysis on terabytes of web traffic in a scalable manner. They developed Map Reduce algorithm with a new format able to handle libpcap files in a parallel manner. But there is limitation associated with this method. They hardcode the features that are extracted from the libpcap files and thereby the user is not allowed to decide the feature set based on the problem instance.

In [7] authors made a distinction based on certain features related to traffic volume and classify them as benign and malicious in network flow records. Authors in [8] developed Bayesian Regularized Neural Network based Botnet detection method which achieved high precision. Researchers in [9] build up scalable quasi-real-time intrusion detection system. It is used to detect Peer-to-Peer Botnet attacks using machine learning approaches. This is based on a distributed framework using Hive and Mahout. This [10] paper describes a scalable method for detecting P2P Botnet regarding the relationships

between hosts and it is based on a Hadoop cluster. Their evaluation illustrates reasonably high detection accuracy and a good efficiency.

Bro [6] is a security monitoring system, has been extended to support the cluster environment [13]. Even though, it gives only independent packet processing a teach one host for real time packet streaming, so it is not able to examine a large data in the cluster file system. Ttat [11] is a passive analysis application which involves TCP trace, and it gives a variety of analysis capabilities with regard to classification and TCP performance metrics.

Authors [12] developed a distributed Snort system that assembles warning messages from numerous Snort processes which execute warning messages on different machines in a parallel manner using MapReduce functionality by Hadoop. Their experiment shows that using two or more slave nodes has enhanced performance than a single node system, also system with eight slave nodes shows almost four times faster speed than that of a single node system. However, it is not capable of real-time analytics.

[13] is scalable NIDS (Network Intrusion Detection System) log analysis system based on cloud computing. The main purpose is to efficiently handle large volume of NIDS logs from server's machines using Map reduce and cloud computing. Analysis is carried out on Snort log report with file size of 4 GB. The running time for log analysis is found by evaluating with the Hadoop based system and with a single node without Hadoop. As number of nodes increases, the system performance increases as compared to the single system. When the number of nodes in the system are five, the performance of the system is almost double than that of the single node.

In [14] Hybrid intrusion detection system, analysis is performed by using Hadoop Hive ecosystem. Hive is used as a data warehousing tool, which efficiently analyses large size of data. Hadoop configured with Hive makes IDS very scalable by giving packet analysis with language HiveQL similar to SQL. Another Network Intrusion Detection System [15] is able to perform analytics over the behavior of intrusion and their patterns on the network, which assists administrator to configure policies and settings for network security. Analytics over intrusion is performed by using a Score-Weight approach known as Pattern Frequency Inverse Cluster Frequency (PF-ICF).

Nearly all of the earlier research was focused on detecting a specific activity and specific attack and their solutions were not described to be successful in identifying general malicious behavior and attacks, whose features were not used in the training data set.

Indeed, it can be observed that using a machine learning techniques for the detection of malicious behavior is far better than conventional signature based approach as the attackers redesign the scripts frequently and the behavior and functionality of attacks varies quite extensively with each version release of the virus. Signature based techniques depends a lot upon the existing virus signatures to identify any activity.

In particular when handling zero-day attacks, signature based technique fall short completely as there is no account of prior activity for that attack. As a result, a machine learning technique is preferred to identify suspicious activity based on the anomalous behavior of the request.

In the prior work, there has not been much study on developing the detection in a real-time system to monitor and mitigate malicious activity.

On the other hand, SEIM based tools provide real time analysis of security alerts generated by network hardware and applications.

SEIM has been a foundational tool for a long time on the market place and it is the core data repository for security events about most organizations. However, it has a number of limitations which cause researches to move from SEIM based tools to Security analytics. Limitations of SEIM based tools are briefly given as follows.

- SEIM does the limited analysis mainly based on correlating and normalization of alerts.

- SEIM only understands those events inside of its defined rules or policies because it is primarily rules based, policy-based and trigger-based.
- SEIM does not understand the application specific activity and behaviour or pattern (anomaly detection) to find out how a number of activities raise the threat level.
- SEIM only understands log entries and network information to compare events at network level and find out network alerts.
- For advanced notification SEIM requires manual investigation.

Consequently, the existing correlation capabilities of SIEM tools, primarily based on a single machine in centralized servers, have confirmed to be inadequate to process large volume of events and data.

For the above mentioned reasons, the scenario has been shifted from SEIM base approach to security analytics which can analyze security intelligence in real time. It incorporates current data being generated and the massive volumes of event data and existing logs.

By applying this it is possible to stop and distinguish advanced threats as they occur, for that parallel processing is necessary; the tools should have the intelligence to evaluate anomalies and make a decision whether they are true threats or not, to keep away from troubles and damage to the productivity. Salient features of Security analytics are as follows.

- The ability to analyze and process huge volumes of security related data (offline and online) in real time.
- Reduce false positives and alert volumes.
- Provide better prioritization and enhanced visualization.
- Increased ability to easily aggregate and cross analyze data from non-security sources, for instance web access logs and server logs.
- Sophisticated process engine for correlating the information from different sources.
- Advanced automated response capabilities.

Ability to use data from outside sources to give information on the new categories of threat that has been observed somewhere else.

3. Big Data Analytics and NoSQL

Big data is a term that describes huge volumes of high velocity, variable and unstructured data that need advanced techniques and technologies to capture, store, distribute, manage, and analysis of the information [18]. It comprises of both structured and unstructured data which grow large so fast that are not manageable by conventional database management systems or traditional statistical tools. A brief comparison of Big Data with conventional data is given in the table (Table1). Big Data is described in terms of three V's that is Volume, Velocity, and Variety.

Volume is the size of data. Big data size is in multiple terabytes and even petabytes. A survey done by IBM in 2012 exposed that just over half of the 1144 respondents considers data over one terabyte to be Big Data [19].

Table 1. Conventional Data Vs Big Data

S.No.	Attributes	Conventional Data	Big Data
1.	**Data Volume**	From GB (Gigabytes) to TB (terabytes)	From Petabytes to Zettabytes
2.	**Data Variety**	Data is Structured (DBMS and spreadsheets)	Data is Semi-structured (XML and Logs) and Unstructured (Multimedia and Text)
3.	**Data Model**	It has Strict Schema (ER-Diagram)	It has Flat Schema
4.	**Data Organization**	Data is stored in centralized machine	Data is stored in distributed machines
5.	**Data Relationship**	It has complex interrelationship	Almost flat with few relationship

Definitions of Big Data volumes vary by factor, for instance type and the time of data What could be considered Big Data today may not meet up the threshold in the future since storage space will increase, allowing bigger data to be captured.

Variety is the type of data in a dataset. Structured, unstructured, or semi-structured data types are used in a variety of applications. Structured data, which includes only five percent of all existing data [20] is the tabular data found in relational databases and spreadsheets. The semi-structured data does not necessarily follow strict standards. Web access logs and other logs, XML (Extensible Markup Language), HTML, languages for exchanging data on the Web applications are classic examples of semi-structured data. HTML tags and XML user defined data tags which makes easy for the machine to parse and process. Multimedia data such as images, audio, and video and text are example of unstructured data, which fall short the structural organization required by tools for the analysis.

Velocity is the speed at which data are generated and the rate at which it is supposed to be processed and analyzed. The increase of digital devices such as sensors, surveillance cameras, and smart phones has led to an extraordinary speed of data creation. Even retailers are generating high volumes of data. For example, Wal-Mart processes more than one million transactions per hour [20]. Facebook processes 500TB each day and hundred hours of video are uploaded to YouTube every minute [19].

Big data are insignificant and useless in a data repository. Its potential price is only when leverage to drive decision-making. To make such evidence based decision-making, firms require efficient methods to turn high volumes of speedy and diverse data into significant and useful insights.

Generally, the process of digging out insights from big data can be divided into two main steps [18]. Data management and analytics, data management further divided into three steps involves acquiring, preprocessing and storing of data. On the other hand, Analytics is a technique used to analyze and dig out intelligence and useful patterns from big data. Analytics outcomes are of four types.

- **Descriptive analytics:** Sometimes, it is also referred as fact analytics. It describes the data in a detailed way. The technique used in descriptive analytics is an unsupervised learning such as clustering. In security analytics, it is used for anomaly detection and visualization of data.
- **Diagnostics analytics:** It describes the cause or the reason of the occurrence of an event. In security analytics, it diagnosis the security breaches, vulnerability and threats associated with the applications.

- **Predictive analytics:** It predicts the future event in advanced. The technique used in predictive analytics is supervised learning such as classification and regression analysis. In the context of security analytics, it gives advanced notification and real time alerts.
- **Prescriptive analytics:** It gives the recommendation for the future. All the four types of analytics are shown in Figure 1.

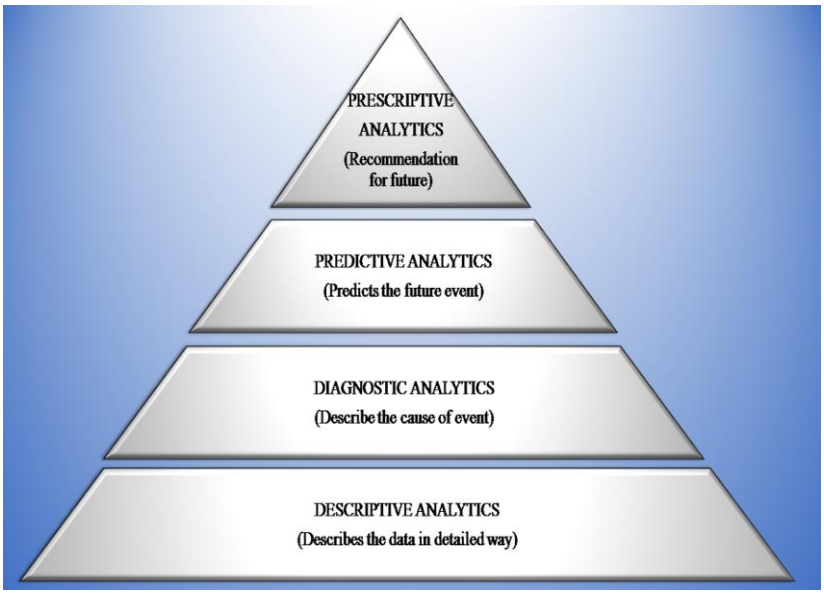

Figure 1. Types of Analytics

3.1. NoSQL

NoSQL or Not Only SQL refers to group of non-relational database management systems where databases are not built mostly on tables, and usually do not make use of SQL for data access [22]. NoSQL database management systems are useful when work with a large amount of data when the nature of data does not need a relational model. NoSQL database systems are distributed database systems that are designed for large volume of data storage and for parallel data processing across many commodity servers.

Relational databases are incapable of handling the Big Data and this is by the virtue of their design, they are not horizontally scalable even the Oracle database, Microsoft SQL Server or MySQL. To overcome this limitation, NoSQL databases have come up on the scene. The key idea behind NoSQL databases is horizontal scalability, which was not there in relational databases.

Horizontal scalability refers to keep on adding more and more computers as need of more and more power [22]. They are not any specialized computers; they are simply commodity computers so if there is a need of doubling the performance, so double the number of computers. Performance is linearly proportional to the number of computers. In contrast to that, conventional RDBMS systems are vertically scalable.

Vertical scalability refers to upgradation of the hardware. If the application is running on a single machine, it is generally possible to add a combination of CPU, faster disks, extra memory, for the easiness of any database bottleneck [23]. Horizontal and Vertical Scalability are shown in Figure 2. NoSQL databases used analytical processing of large volumes of datasets, providing increased scalability over commodity computers [24]. Storage and computational requirements of applications for instance Business Intelligence, Big Data Analytics and social networking over petabytes and zettabytes data have made centralized RDBMS to their limits [25].

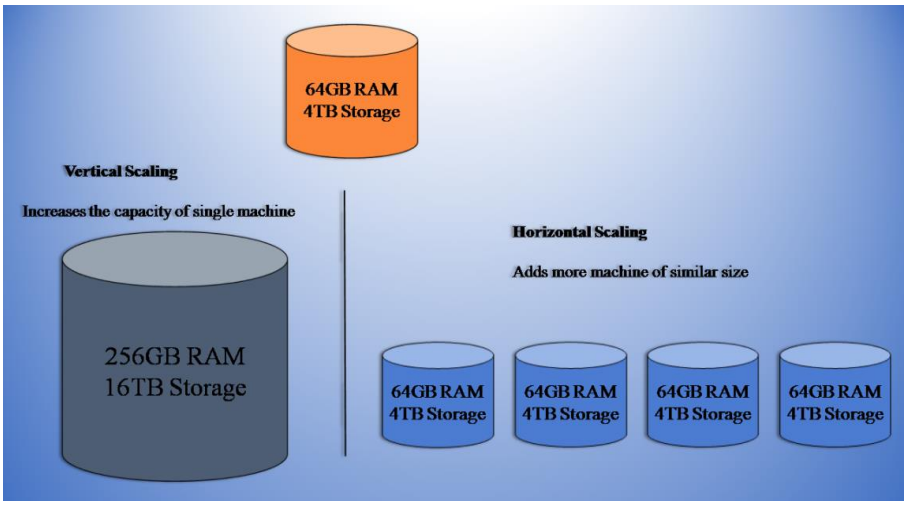

Figure 2. Horizontal and Vertical Scaling

Leavitt [23] categories NoSQL databases into three categories: key value stores, column oriented databases, and document oriented databases. Key-value stores for example SimpleDB items as alphanumeric keys and associated values in hash tables. The values could be simple strings or complex lists. Data searches usually performed against keys and not values. Column oriented databases for instance, Big Table from Google, Cassandra and HBase.

Document oriented databases as their name suggests, developed to store and manage documents. These documents are encoded in a data exchange format, for instance, JavaScript Object Notation, Binary JSON, and XML. Examples of Document databases are MongoDB and CouchDB. The details of each category are beyond the scope of a single paper. However, in the next section MongoDB is explored which is the database for proposed security analytics the framework.

3.2. MongoDB Introduction

MongoDB is a scalable open source high performance document oriented database [24]. It is an open source database and is developed, maintain, and supported by a 10gen in the USA. MongoDB is available under general public license for free and it is also available under commercial license from the manufacturer. According to one of the DBMS ranking website (DB Engines) [25], the MongoDB is ranked at fourth position among all DBMS.MongoDB usage is growing for a reason of business analytics, can easily integrate with web application, and for different log analysis.

In terms of structure and implementation, MongoDB and conventional RDBMSs are different. A document is the fundamental unit of data in MongoDB and is by and large is same as a row in RDBMS. In the same way, a collection is equivalent to a table having a dynamic schema. In MongoDB each document has a unique key (_id) and act as the primary key within a collection. In single machine, MongoDB could host numerous independent databases, and for each database there can be its own collections.

In MongoDB, Documents are stored in JSON-like (Java Script Object Notation) [24]. JSON is a straightforward representation of data and it is an open standard format for data transaction based on java script. JASON follows the notion of objects in JavaScript. In this format, an object is represented by a set of name-value pair. Every object commences with left brace and closes with right brace. Each name-value pairs are divided by comma. Value could either hold an object or an array. Array commences with left bracket and closes with a right bracket [24]. In Figure 3 entry of the web access log is represented in JSON format.

At any time, JSON document can be inserted into a MongoDB database since there is no fixed-schema unlike relational database management systems. As a result, MongoDB have a benefit that their data models could be modified at run time. Moreover, JASON format is reasonably good in handling semi structured data such as web access logs, firewall logs, IDS logs and database logs.

```
{
“IP" : "14.139.241.85",
“Date”: "25/05/2016 5:25:51 PM",
“USER" : "-",
“Request" : "GET",
“Path" : "/",
“Status" : "200",
“Size" : "56",
“Agent":"Mozilla/5.0 Windows NT 10.0”
}
```

Figure 3. Web Access Log Entry in JASON

3.2.1. Replication in MongoDB

Replication is a method of maintaining identical copies of data on multiple machines. Replication maintains the application and keeps the data protected. The replication is set up by creating a replica set. A replica set is a cluster of server nodes with one primary, and multiple secondary nodes. If the primary server breaks down, the secondary servers can choose a new primary server. In replication method, the data is synchronized automatically between primary node and secondary node, and replication guarantees service even if one or more server crashes. The minimum number of nodes should be three in replica set configuration, if a replica set having only two nodes there cannot be an election in case the primary server crashes. The Replication process in MongoDB is shown in Figure 4.

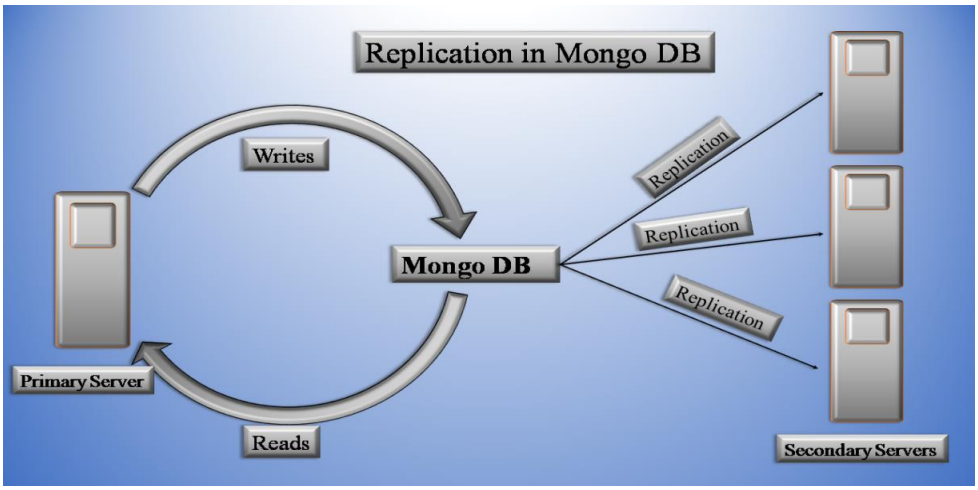

Figure 4. Replication in MongoDB

3.2.2. Sharding in MongoDB

Sharding is a process of divide the data across multiple machines; sometimes it is also referred as partitioning. By storing a part of data on different machine, it becomes feasible to put large data without the need of better or high configuration machines. With the growing speed of current applications, it is becoming increasingly expensive, and in a number of cases unfeasible, to get a single machine capable of handling the entire application data. Moreover, MongoDB provides transparency in sharding environment to all the application, *i.e.*, querying a sharded cluster is same as the querying a single mongoDB server instance or replica set. Sharding in MongoDB are of two types; manual sharding and auto sharding. Manual sharding is applied when an application requires a connection to many different databases and all of the databases are completely independent. The application stores different type of data on different machines and querying the particular server to get the desired result. On the other hand, auto sharding refers to automatically balancing of data across different clusters and adding and removing of clusters depending on load. Typical sharding process is shown in Figure 5. The primary database has four collections; the sharding process distributed the load into four machines.

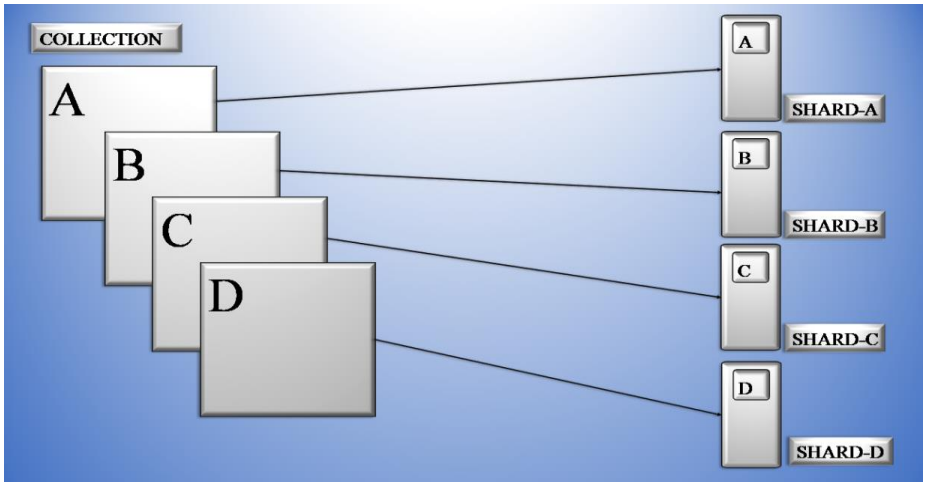

Figure 5. Sharding in MongoDB

3.2.3. Real-Time Security Analytics using MongoDB

With the help of fast indexing and powerful query model, MongoDB facilitates its users to directly run analytics in real-time. In contrast to relational database management systems which levels data into a two-dimensional tabular form of rows and columns, MongoDB powerfully models and stores a variety of structured and unstructured data as documents. MongoDB provides a wide array of query, update operators, and projection, including:

Queries based on Key / value pair: it returns results based on any key in MongoDB document, generally the unique key.

Queries based on Range operators: it returns results based on values defined in expression for instance operators such as equal to (=), not equal to (! =), greater than (>) and less than (<).

Queries based on Text search: it returns results in desired order based on arguments using Boolean operators such as AND operators and OR operators.

Distributed and parallel queries: it is based on Map-Reduce and Aggregation and queries.

4. Real-time Security Analytics Framework for Attack Detection

Users As shown in Figure 6 the proposed security analytics framework consists of five phases:

Data Extraction, Data parsing and storing in MongoDB, Feature Selection, Machine Learning, and Final Output.

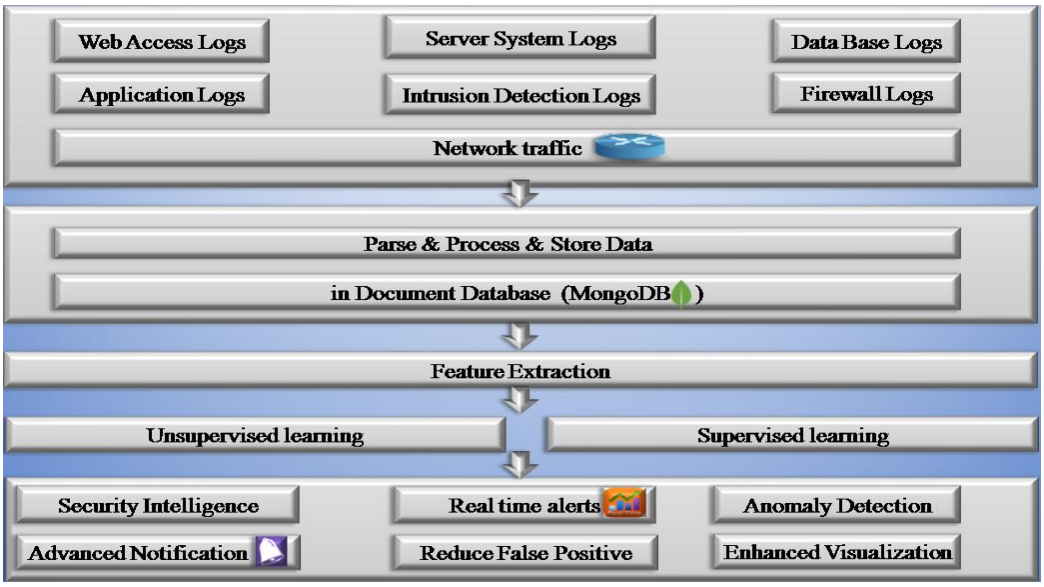

Figure 6. Real Time Security Analytics Framework

4.1. Data Extraction Module

This module consists of collecting and gathering massive amount of data from different security domains. The notion of data for security analytics is extensive, and can be classified into active and passive data sources [28]. Active data sources are associated with real-time data consist of; User identification such as ID and password, Digital Certificates, Biometric credentials such as face and voice recognition, Data from social media *etc*. Passive data source consists of; Data generated from access logs such as web access logs, Application logs, Firewall logs, Intrusion Detection System logs.

Security Analytics applied on active and passive data sources together would give a complete view of traffic, for instance by pointing out an abnormal behavior in the access pattern of a given request. Proper prevention methods can be applied, such as change of network settings, lock accounts, and two-way security. The data extraction module is further categorized into two parts.

- Log Generation and Collection
- Network Traffic Sniffing

Log Generation and Collection are fundamentally, gathering the passive data from numerous sources in the data repository. To achieve this objective a variety of large volume logs (over 25GB) are collected from different data sources. These data sources include live Institutional Websites and Experimental Setup. The collected logs including, the Apache server logs from multiple live websites; cover a period of three months from Jun'16 to Aug'16. The Apache web server generates massive amount of valuable information about access of users and errors. In particular, the Apache web server logs all the requests handled by the web server.

In the next step, the test-bed is created for gathering the malicious activity. It consists of isolated virtual machines and on top of every machine; samples of malicious program (Botnet) are installed. These Botnets include Slowhttptest, Slowloris, Pyloris and variants

of Agobot. Apart from these standard Botnets, one DDoS (Distributed Denial of Service) attack scenario is created and executed in test-bed. It is a malicious program written in C# for network Flooding. The key purpose is to identify zero-day attacks, as there is no account of prior activity for zero-day attack. All of these mentioned Botnets including new developed attack are then launched on a dummy web application created in the test-bed. In addition to that, genuine traffic is also created in test bed with normal user requests.

A second key source of data is Network Traffic Sniffing. For achieving this purpose Dumpcap [30] is used in test-bed for sniffing the data packets from the network in combination with other known tool Tshark [30] for taking out the data field, associated to feature set. After the extraction, the desired data fields are then stored in MongoDB.

4.2. Data Parsing and Storing in MongoDB

Once the massive volume of security related data is collected and generated, the next requirement is to store data in a standard database. Moreover, the demand of real-time system is to add continuously data in real-time. Numerous tools are there, for the purpose of parsing and storing the semi-structured data into one of the standard database format. The proposed framework is having the MongoDB document oriented database, so for inserting semi-structured data logs in real time Fluentd [29] is used.

Fluentd is an open-source semi-structured log-processing tool initially developed at Treasure Data, Inc. It is written in C and Ruby but its major code is in Ruby. Fluentd works in three basic steps.

- In the first step, it constantly tails the log file such as apache web access, server logs, and other logs.
- In the second step, it parses the log entry into meaningful records such as host, user, and code and then buffers the data.
- In the last step, it stores the buffered data to MongoDB documents periodically.

The working mechanism of Fluentd is shown in Figure 7.

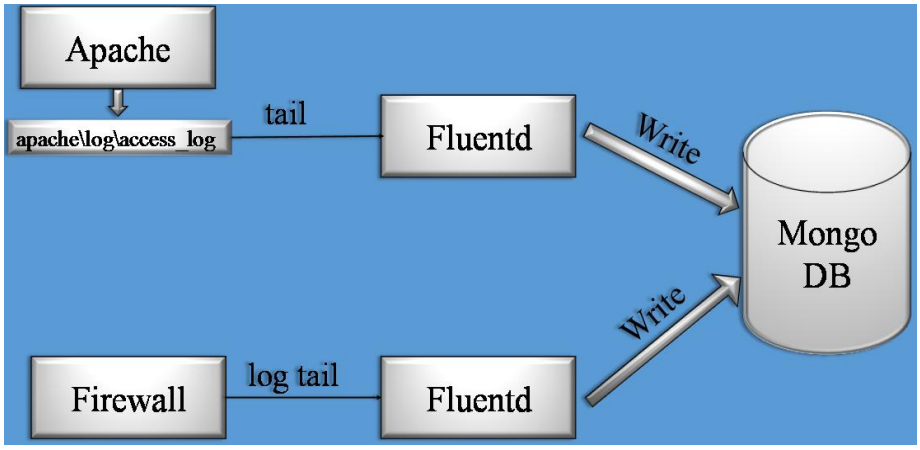

Figure 7. Working Mechanism of Fluentd

After parsing the semi-structured log, the log entry is stored in MongoDB collection as

```
{
"_id" :ObjectId("5cd1cd3a347895ab88750001"),
"host" : "14.139.241.85",
"user" : "-",
"method" : "GET",
```

"path" : "/",
"code" : "200",
"size" : "44",
"agent":"Mozilla/5.0 Windows NT 10.0”
"time" : ISODate("2016-7-26T09:28:27Z")
}

4.3. Feature Selection

Feature selection is a powerful and a necessary step in efficient, high dimensionality analytics applications. It is generally an essential step for data processing before applying the machine-learning algorithm. Reducing, the attribute size makes thing simpler for understanding the model and the handling of visualization technique. Principally, two methods are used for the purpose of feature selection; filter method and wrapper method. Filter method selects general characteristics of the data in a heuristic way. Filter method runs fast as compared to the wrapper method, since it does not use learning techniques. On the other hand, in wrapper method machine-learning techniques are used. In general, wrapper method gives useful features as compared to the selection method.

The key feature of security analytics framework is it can vary the features at run time. In the proposed framework, the feature set is divided into two parts based on Data Extraction and Collection in the first phase.

The first features set is obtained from access logs of live web applications and generated from test-bed. This feature set is used for detecting anomalous behavior of the user after applying the supervise machine learning techniques. Some of the features selected in this part are Distinct IP, timestamp range, request type, content type, and content size, and user agent, and response code, total number of requests and percentage of error requests. All of the fields are used to monitor the behavior of the user.

The second features set is obtained from the Network Traffic Sniffing module mentioned in Data Extraction phase. All the features selected in this part are based on the flow of data in the network. The flow based features set include source and destination IPs, total number of packets in forward and backward direction, size of the largest and smallest packet, maximum and minimum time between packets sent in forward and backward direction, number of bytes used for headers in forward and backward direction.

4.4. Machine Learning Phase

The machine learning module is the nucleus of security analytics framework. In this module, both supervised and unsupervised learning are implemented. In supervised learning, the class-label is known where in unsupervised learning the class-label is unknown. Classification and Regression are the examples of supervised learning. Clustering is an example of unsupervised learning. In the proposed framework, clustering the unsupervised learning is used for anomaly detection. In addition to that, supervised learning the classification technique such as K-nearest neighbors, Naive Bayes and Decision Tree is used for known attack detection. Both techniques together form a complete security analytics framework. These learning techniques are described in detail in following section.

4.4.1. Unsupervised Learning for Anomaly Detection

Clustering is the method of grouping the objects into clusters or classes, so that data objects within a one cluster have high similarity as compared to data objects in other clusters. The similarities are calculated based on the values of attribute that describes the object [31].

Clustering has drawn attention in the field of anomaly detection [32]. The key advantage of using a clustering technique is its ability to detect anomalies and zero-day

attacks from data without prior descriptions and class labels (attack signature). In this method, the model is trained from unlabeled data consisting of both normal and attack traffics. The main idea here is that the attack traffic or anomalous traffic that forms a tiny proportion of the entire data. Using above mentioned assumption, attack traffic, and anomalous traffic could be detected based on cluster volume, big data clusters correspond to normal data traffic and the remaining data objects correspond to the attacks [33].

Numerous clustering algorithms are there in the literature. Generally, the main clustering algorithm is classified into given categories

- Clustering based on partitioning such as k-means and k-medoid
- Hierarchical Clustering such as Divisive (Top down) and Agglomerative (Bottom up)

K-means and k-medoid are flat partitioning algorithm divides the unlabeled data into k-clusters. The advantage of partitioning clustering is less execution time as compare to hierarchical clustering. However, it has a limitation that it does not give in depth analysis such as the hierarchical structuring or deep association among the clusters.

On the other hand, Hierarchical clustering [31] partitions data objects in levels by creating a tree cluster. This tree structure is referred as dendrogram in hierarchical clustering. The tree is not a flat set of clusters, but it is having a multilevel hierarchy, in which the data clusters at one level are attached to the data clusters at the next level. This flexibility enables the application to decide the appropriate level in clustering. The only limitation is its execution time, which can be overcome using distributed parallel computing model such as map-reduce.

In a proposed model, Agglomerative hierarchical clustering is used in all of its variants. In the first step, the hierarchical clustering is applied on flow data and group the data into normal and anomalous. The characteristics of anomaly clusters are then further analyzed.

This algorithm starts by inserting each data object in its own cluster and then combines these clusters having single data object into bigger and bigger clusters, until all of the data objects are in a one cluster.

Given a set of "n" data objects $\{U_1, U_2, U_3, \ldots, U_n\}$. Each data object is an instance and they are usually treated as vectors. These vectors (instances) are representing the points in coordinate space. The first step is required to find out the proximity matrix having the distance between each vector using a distance function. Let the proximity matrix is "D", and then it can be represented in matrix form. The following four methods vary in how the distance among each data cluster is (Figure-8)

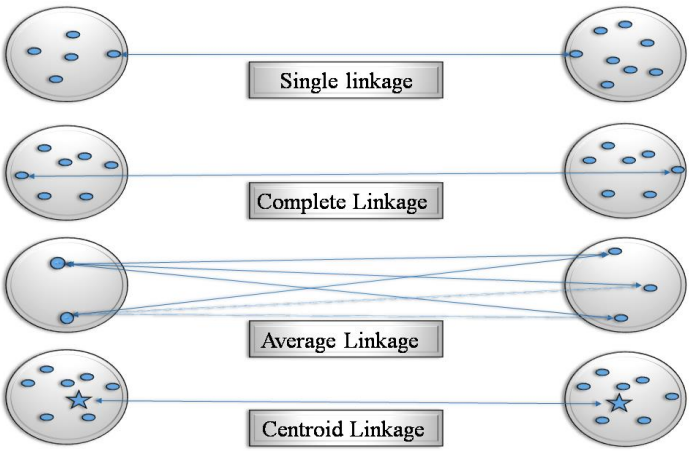

Figure 8. Distance Measures in Hierarchical Clustering

- **Single Linkage:** In this method, the distance between two clusters is the shortest distance between two vectors in each cluster.
- **Complete Linkage:** In this method, the distance between two data clusters is the longest distance between two vectors in each cluster.
- **Average Linkage:** In this method, the distance between two data clusters is the average distance between each vector in one data cluster to every vector in the other data cluster.
- **Centroid Linkage:** In this method, the distance between two data clusters is the distance between the centroid of one cluster to centroid of another cluster. All of the mentioned methods are show in Figure 8.

Following are the steps performed in an algorithmic approach to achieve hierarchical agglomerative clustering:

Initialization:

The proximity matrix is initialized using any one of the above-mentioned four methods for the distance calculation. All the distances are stored in proximity matrix "D".

1. For (i =1 to n)
   For (j =1 to n)
   Repeat Steps 2 to 3
2. Search the distance matrix 'D', for the nearest (most similar) pairs of clusters. For instance, consider 'U' and 'V' be the two nearest clusters and distance between them is d_{uv}.
3. Merge Clusters U and V into a single cluster. Update the proximity matrix 'D' by: x cx

3.1 Deleting the rows and columns containing the clusters U and V in the matrix and resizing the matrix.

3.2 Adding a new row & column containing the distance between the pair of cluster U and V as the merged value (U, V).

4. Return the distance matrix 'D'.

Time-Complexity Analysis

The time-complexity of the above algorithm comes out to be $O(n^3)$. This is because, time required to find out two nearest pairs of clusters in a matrix of order (n*n) is $O(n^2)$. This process is done N times, until only a single vector is left in the matrix. Hence, the complexity becomes $O(n^3)$.

4.4.2. Supervised Learning for Known Attack Detection

Once, the preprocessing of different logs taken from the Apache web server and extracting features desired for the classification, several classification algorithms were chosen for detection the known attacks. These algorithms include Random forest, Naive Bayes, K-nearest neighbors, and Decision Tree. All these algorithms belong to the supervised learning category. The precondition for all classification technique is that the basic dataset should be divided into two parts; first part for training the model and the remaining part for testing the performance.

Naive Bayes Classification:

It is a straight forward probabilistic classification technique, which uses Bayes theorem as basis by means of a naive assumption of independence. In this technique, a certain number of classes are predefined in the training phase. If a new instance comes, the

classifier computes the relative probability to every class, and then classifier will predict that instance belongs to the class having maximum probability.

The Naive Bayes classifier seems to be the appropriate solution in attack detection because of its predictability characteristic, which is useful in an uncertain network. It calculates the likelihood or probability value of the class labels and therefore it shows the safety percentage and the risk percentage.

K-nearest Neighbors:

K-nearest neighbors in short k-NN is instance-based method for classifying data objects based on nearest samples in the feature space. Nearest-neighbor technique is based on learning by comparing a given test object with the training data, if it is similar or not. The objects are having 'n' attributes. A vector in an n-dimensional space represents each object. When unknown objects come, this classifier look for the nearest space of the k training data objects that are nearest to the unknown object. If multiple objects are the same minimum distance to the test object, the object, which is found, first is used. For the purpose of an experiment, the value of k is one.

Decision Tree:

A decision tree is having a pair of true-false decision rules. The decision tree can be thought as a flowchart tree like structure. Each non-leaf node denotes a test on data attribute; each branch gives the result of the test, and each terminal node represents a class label.

Random Forests:

Random forests, sometime also referred as random decision forests are supervised learning methods for classification that runs by building a number of the decision trees at training phase and predicting the class of the new instance. Random decision forests is an improvement over decision by over fitting multiple decision trees in the training phase and randomly decide sub samples from the trees. The decision trees that are full-grown can learn extremely irregular patterns; they over fit the training data, since they have a very high variance. Random forests do the averaging of multiple decision trees by training the different parts of the same training data, with the objective of reducing the variance. In proposed model, random forest consists of ten trees.

5. Environment Setup and Result Analysis

The technologies underlying proposed framework are MongoDB, Fluentd and Python. Fluentd is used for parsing data from semi-structured source to MongoDB. Python is scripting language and becoming popular for NoSQL Data Connectivity. It has library for both connecting to MongoDB (PyMongo) and machine learning modules (scikit). It provides a mapping between JASON objects and python data structures. PyMongo and scikit are exploited in the proposed framework. The architecture diagram of the environmental set-up is shown in Figure 9.

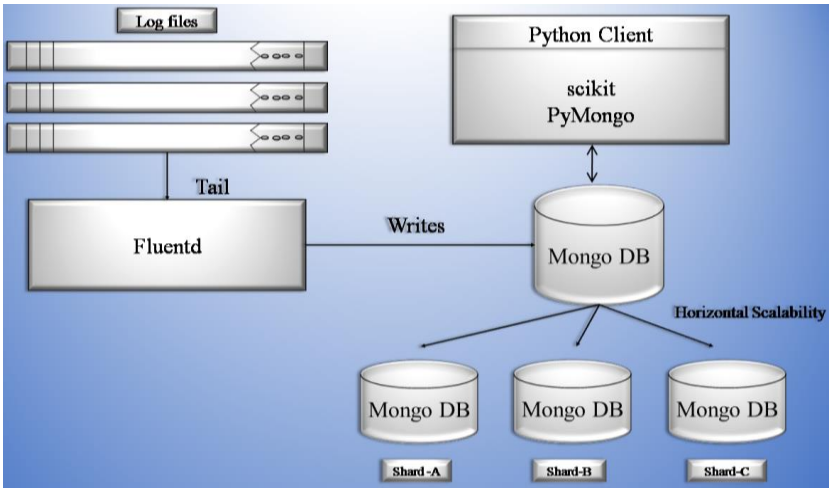

Figure 9. Environmental Set-Up

5.1. Result Analysis

Classification and clustering techniques are performed on the tuples extracted from log during log parsing phase. From clustering analysis, the data is parted into normal and anomalous clusters. Some of the attack instances (Slowhttptest, Slowloris, Pyloris) are not identified in clustering analysis. The reason could be, all these attacks operates on application layer and exploits HTTP traffic. On the other hand, supervised learning module correctly identifies these attack instances. After combining the results from both the techniques, it is possible to identify most of the attacks with minimal false positive rate.

Table 2. Hierarchical Clustering results

Algorithm	Detection Rate (True Positive Rate)	Accuracy
Single Linkage	0.8472	0.9624
Complete Linkage	0.8469	0.9611
Average Linkage	0.8520	0.9628
Centroid Linkage	0.8759	0.9662

5.2. Clustering Outcome

Here, the behavior of different hierarchical clustering methods in building an efficient framework for anomaly detection is presented. In Table II, the comparative analysis of different hierarchical clustering algorithms is presented**.**

Figure 10 shows the execution time of all the variants of hierarchical clustering. Figure 11 shows the dendrogram constructed from small data subset of clusters for better visualization. In this dendrogram, clusters are manually divided into normal clusters and anomalous clusters.

5.3. Classification Outcome

In supervised learning module, 10-fold cross validation scheme is used in order to validate and check the effectiveness of the model. In 10-fold cross-validation scheme, the given data set is divided into 10 of roughly equal size of data subset. Then one data subset is used for test set and the rest of nine data subsets are used for building the model.

Table 3. Classification Results

Algorithm	Recall	Precision
Random Forest	0.9137	0.8270
Naïve Bayes	0.8720	0.8118
Decision Trees	0.8725	0.7852
K-Nearest Neighbor	0.8756	0.7821

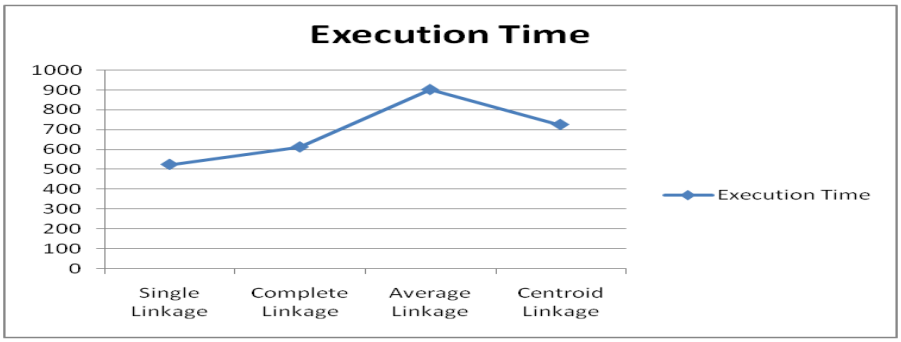

Figure 10. Execution Time of Hierarchical Clustering

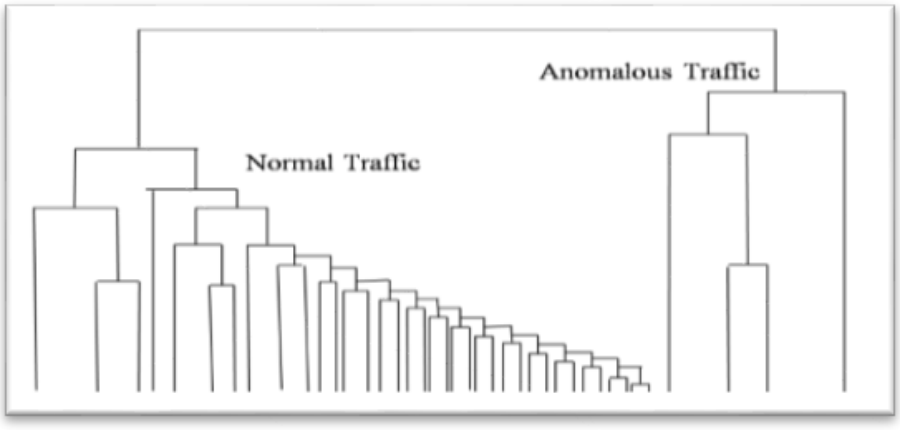

Figure 11. Dendogram of Hierarchical Clustering

This process is repeated 10 times in order to use each test subset at least once. The accuracy is then the average of all the accuracies calculated in each fold. Results from supervised learning module are shown in Table III. The results show that the Random

forest algorithm has the highest percentage of correctly classifying the attack. Remaining all the algorithms perform equally well for given data set.

6. Conclusion

As the diverse security-related data is generating at enormous speed by software organizations, the old security models are not sufficient to defense against new breed of cyber attacks. Moreover, the traditional relational databases are not capable of storing the security-related data in a single machine. In this paper, five-phase scalable security analytics framework has been proposed. The proposed framework is capable of storing and handling variety of unstructured data in real-time.

The proposed framework is evaluated and validated through experiments on set of dataset collected from live web applications and self-generated dataset from different botnets such as Slowhttptest, Slowloris, Pyloris and variants of Agobot. The experimental results illustrate that the framework system is capable of detecting the known and unknown malicious instances including zero-day malware efficiently irrespective of their patterns with low false positive rate and high detection rate.

References

[1] C. Zhang, X. Shen, X. Pei and Y. Yao, "Applying Big Data Analytics Into Network Security: Challenges, Techniques and Outlooks", 2016 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, **(2016)**, pp. 325-329.
[2] K. Saur, T. Dumitra and M. Hicks, "Evolving NoSQL Databases without Downtime", 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME), Raleigh, NC, USA, **(2016)**, pp. 166-176.
[3] N. Mandal and S. Jadhav, "A survey on network security tools for open source", 2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), Bangalore, **(2016)**, pp. 1-6.
[4] Y. Lee and W. Kang, "A hadoop-based packet trace processing tool. In International Workshop on Traffic Monitoring and Analysis", Springer Berlin Heidelberg, **(2011)**, pp. 51-63.
[5] D. J. Suryawanshi and U. A. Mande, "Parallel Processing of Internet Traffic Measurement and Analysis Using Hadoop", **(2014)**.
[6] Y. Lee, "Toward scalable internet traffic measurement and analysis with hadoop", ACM SIGCOMM Computer Communication Review, vol. 43, no. 1, **(2013)**, pp. 5-13.
[7] T. F. Yen and M. K. Reiter, "Are your hosts trading or plotting? Telling P2P file-sharing and bots apart", 2010 IEEE 30th International Conference on Distributed Computing Systems (ICDCS), **(2010)**, pp. 241-252.
[8] S. C. Guntuku, P. Narang and C. Hota, "Real-time Peer-to-Peer Botnet Detection Framework based on Bayesian Regularized Neural Network", **(2013)**.
[9] K. Singh, S. C. Guntuku, A. Thakur and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using random forests", Information Sciences, vol. 278, **(2014)**, pp. 488-497.
[10] J. Francois, S. Wang, W. Bronzi, R. State and T. Engel, "Botcloud: Detecting botnets using mapreduce", 2014 IEEE International Workshop on Information Forensics and Security, **(2014)**, pp. 1-6.
[11] J. Cheon and T. Y. Choe, "Distributed processing of snort alert log using hadoop", International Journal of Engineering and Technology, vol. 5, no.3, **(2013)**, pp. 2685-2690.
[12] M. Kumar and M. Hanumanthappa, "Scalable intrusion detection systems log analysis using cloud computing infrastructure",2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), **(2013)**, pp. 1-4.
[13] P. G. Prathibha and E. D. Dileesh, "Design of a hybrid intrusion detection system using snort and hadoop", International Journal of Computer Applications, vol. 73, no.10, **(2013)**.
[14] S. R. Bandre and J. N. Nandimath, "Design consideration of network intrusion detection system using Hadoop and GPGPU", 2015 International Conference on Pervasive Computing (ICPC), **(2015)**, pp. 1-6.
[15] D. Miller, S. Harris, A. Harper, S. VanDyke and C. Blask, "Security information and event management (SIEM) implementation", McGraw Hill Professional, **(2010)**.
[16] G. Cybenko and C. E. Landwehr, "Security analytics and measurements", IEEE Security & Privacy, vol. 3, no.10, **(2012)**, pp. 5-8.
[17] R. Mahanty and P. K. Mahanti, "Unleashing Artificial Intelligence onto Big Data: A Review", Handbook of Research on Computational Intelligence Applications in Bioinformatics, **(2016)**, pp. 1-16.
[18] M. Schroeck, R. Shockley, J. Smart, D. Romero-Morales and P.Tufano, "Analytics: The real-world use of big data", IBM Global Business Services, **(2012)**, pp. 1-20.

[19] K. Cukier, "Data, data everywhere: A special report on managing information", Economist Newspaper, **(2010)**.
[20] Banerjee, T. Bandyopadhyay and P. Acharya, "Data analytics: Hyped up aspirations or true potential",Vikalpa, vol. 38, no. 4, **(2013)**, pp. 1-11.
[21] Dasadia and A. Nayak, "MongoDB Cookbook", Packt Publishing Ltd, **(2016)**.
[22] R. Cattell, "Scalable SQL and NoSQL data stores", AcmSigmod Record, vol. 39, no. 4, **(2011)**, pp. 12-27.
[23] N. Leavitt, "Will NoSQL databases live up to their promise?", Computer, vol. 43, no. 2, **(2010)**, pp. 12-14.
[24] P. Membrey, E. Plugge and D. Hawkins, "The definitive guide to MongoDB: the noSQL database for cloud and desktop computing", Apress, **(2011)**.
[25] L. Schuff, Y. R. Choe, and V. S. Pai, "Conservative vs. Optimistic Parallelization of Stateful Network Intrusion Detection", IEEE International Symposium on Performance Analysis of Systems and software, **(2008)** April, pp. 32-43.
[26] J. Landbris, "A Non-functional evaluation of NoSQL Database Management Systems", **(2015)**.
[27] V. N. Gudivada, D. Rao and V. V. Raghavan, "Nosqlsystems for big data management", 2014 IEEE World Congress on Services, **(2014)**, pp. 190-197.
[28] T. Mahmood and U. Afzal, "Security Analytics: Big Data Analytics for cybersecurity: A review of trends, techniques and tools", 2013 2nd national conference on Information assurance (ncia), **(2013)**, pp. 129-134.
[29] Q. Lv and W. Xie, "A Real-Time Log Analyzer Based on MongoDB", Applied Mechanics and Materials, Trans Tech Publications, vol. 571, **(2014)**, pp. 497-501.
[30] M. Sabhnani and G. Serpen, "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context", MLMTA, **(2003)**, pp. 209-215.
[31] J. Han, J. Pei and M. Kamber, "Data mining: concepts and techniques", Elsevier, **(2011)**.
[32] L. Portnoy, E. Eskin and S. Stolfo, "Intrusion detection with unlabeled data using clustering", Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA), **(2001)**.
[33] Eskin, A. Arnold, M. Prerau, L. Portnoy and S. Stolfo, "A geometric framework for unsupervised anomaly detection," Applications of data mining in computer security, Springer US, **(2002)**, pp. 77-101.
[34] Orebaugh, G. Ramirez, J. Burke, L. Pesce, J. Wright and G. Morris, "Other Programs Packaged with Wireshark," Chapter 9.

Authors

Rizwan ur Rahman, he obtained B.E and M.Tech in Computer Science from Maulana Azad National Institute of Technology (MANIT), Bhopal with Hons grade. His programming experience includes C/C++, C#, SQL, ASP, ASP.NET, VB, VB.NET; Win Forms, Web Forms and Java. He has worked on government projects and R&D department of CRISP. Currently he is an assistant professor in Maulana Azad National Institute of Technology. His area of research includes web programming and web security.

Deepak Singh Tomar, he obtained his B. E., M. Tech. and Ph. D. degrees in Computer Science and Engineering. He is currently Assistant Professor of CSE department at NIT- Bhopal, India. He is co-investigator of Information Security Education Awareness (ISEA) project under Govt. of India. Currently, he is chairman of cyber security center, MANIT, Bhopal. He has more than 21 years of teaching experience. He has guided 30 M Tech and 3 PhD Thesis. Besides this he guided 70 B Tech and 15 MCA projects. He has published more than 54 papers in national & international journals and conferences. He is holding positions in many world renowned professional bodies. His present research interests include web mining and cyber security.

