International Journal of Computer Engineering and Information Technology

VOL. 9, NO. 2, February 2017, 24–29
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

# Security and Privacy in Big Data

**Mohammed S. Al-kahtani**

Dept. of Computer Engg., Prince Sattam bin Abdulaziz University, Saudi Arabia

**ABSTRACT**

Providing security and privacy in big data analytics is significantly important along with providing quality of services (QoS) in big data networks. This paper presents the current state-of-the-art research challenges and possible solutions on big data network security. More specifically, we present network security approaches (i.e., intrusion detection, network threat monitoring systems), classify and compare threats and their defense mechanisms that help mitigate the network vulnerabilities from big data and software defined networks (SDN).

***Keywords:*** *Big Data, MapReduce, Hadoop, SDN, Intrusion Detection, Vulnerabilities, Threats.*

## 1. INTRODUCTION

As everyday data are being collected from applications, networks, social media and other sources Big Data is emerging. Studies have shown that by 2020 the world will have increased 50 times the amount of data it had in 2011, which was currently 1.8 zettabytes or 1.8 trillion gigabytes of data [14]. The basic reason for the sharp increase in data being stored over the years simply comes down to cost of storage. The IT industry has made the cost of storage so cheap that applications are capable of saving data at exponential rates. This brings the challenge of having existing network infrastructure learn how to manage and process this big data so that it can be utilized into useful information [12].

Many big data applications work in real-time. Hence, these applications need to create, store and process large amount of information which produces a great deal of volume and demand on the network. When looking at data from a networking perspective, many different areas are needed to be explored These include network topology optimization, parallel structures and big data processing algorithms, data retrieval, security, and privacy issues [10]. The topic of big data is still a new exciting area of research among the IT community and will be requiring much attention for the years to come. A typical organization has a limited network infrastructure and resources capable of handling these volumes of traffic flows which cause regular services (e.g., Email, Web browsing, video streaming) to become strained. This can reduce network performance affecting bandwidth and exposing hardware limitations of devices such as firewall processing being overwhelmed [10], Providing security and privacy has also become a major concern in Big Data as many critical and real-time applications are developed based on Big Data paradigm.

This paper presents a comprehensive survey on big data network security. This work starts by introducing the distributed architecture of big data networks in Section 2. Section 3 focuses on the network security technologies and classifying threats related to security and privacy issues and what type of defense mechanism can be implemented to help mitigate the network vulnerabilities from Big Data and SDN. Finally, the paper concludes with a brief security analysis on big data in Section 4.

## 2. BIG DATA ARCHITECTURE

In this section, we present architecture of MapReduce and Hadoop that are used to processing big data. Moreover, understanding the architecture of Big Data is crucial to design approaches that provide security and privacy in Big Data.

### 2.1 MapReduce and Hadoop

MapReduce is the core component of the Hadoop Apache software framework and is a type of programming model that can be implemented in variety of languages (e.g., Java, C++) that is used for processing big data [13]. This type of software tool can divide applications into smaller fragments or blocks which are then sent out to nodes in a cluster or map. It uses a map function that will able to filter, sort, and distribute jobs to various nodes and also uses a reduce function to collect the results from those jobs so they can resolved into a single value to be used for efficient analysis (Figure 1).

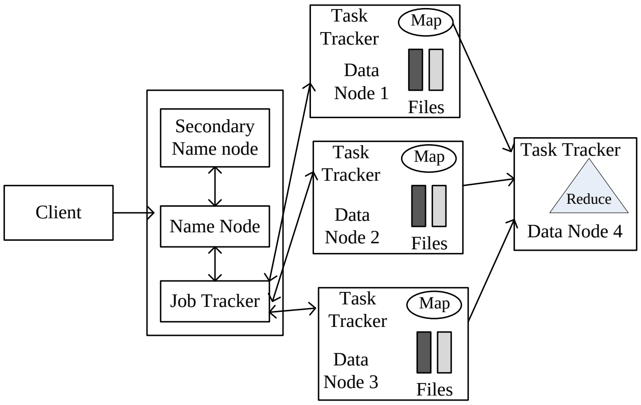

*Fig. 1. Hadoop based on MapReduce paradigm*

The MapReduce consists of a job tracker, task trackers, and sometimes a job history server [17]. The job tracker is used as the master node that is in charge of managing resources and jobs. The task tracker is used to be deployed to each node in order to run the map and help with some of the cluster task load. The job history server is used to track finished jobs and can be deployed as an independent function. MapReduce operates in parallel across vast cluster sizes, while jobs can be divided across many different servers [13]. MapReduce has fault-tolerance where each node sends status updates to the master node, who can re-assign jobs to functioning nodes in cases of node failure.

On the other hand, Hadoop is a management software framework that plays an important role in big data analytics. Hadoop is capable of cataloging, managing, distributing, and querying unstructured large data sets rapidly across many nodes within a distributed network environment [8]. It uses a Hadoop distributed file storage system (HDFS) for storage that divides data into blocks which is distributed and stored on multiple nodes. In order to process the data, Hadoop uses MapReduce to break down data for the nodes to process and sort in parallel (Figure 2). The map procedure is responsible for filtering and sorting and the reduce procedure focuses on summary tasks. It supports high speed transfer rates and is capable of resilient uninterrupted operation in situations when there is node failure [15]. The infrastructure divides up the nodes into groups or racks. Hadoop is an excellent framework for applications using large search engines such as Google and Yahoo.

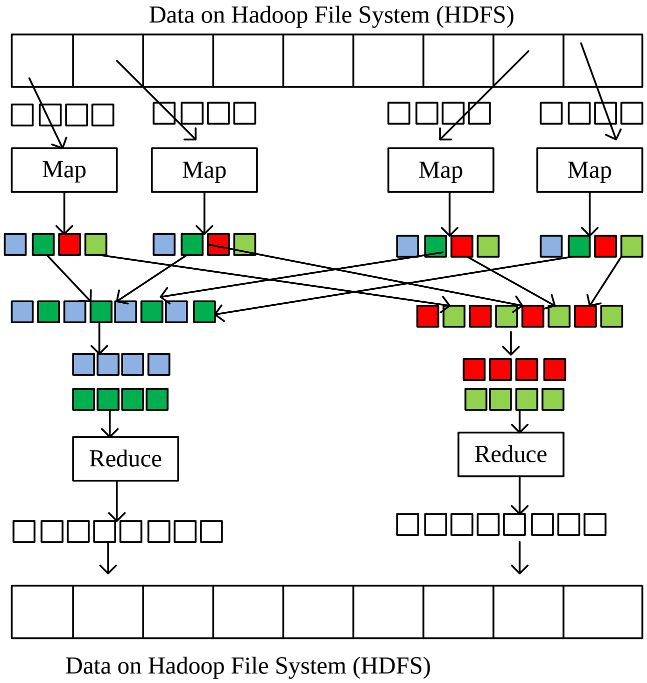
Data on Hadoop File System (HDFS)
Map
Map
Map
Map
Reduce
Reduce
Data on Hadoop File System (HDFS)

*Fig. 2. Hadoop that reduces data and processes in parallel*

Together Hadoop and MapReduce provide the current popular choice for implementation of big data infrastructure [7]. A typical Hadoop network structure consists of a slave (data node, task tracker), master (name node, job tracker) and a client [6]. The client is basically the user interface or query engine. The data nodes are used as storage for the data that contain smaller databases systems and are horizontally distributed across the network. The task tracker is used to process the broken down fragments of the task that has been distributed to a node. The name node maintains a location index of all the other nodes found in the network, so it knows where the specific data is located in which data node. The job tracker is the software job tracking mechanism that is used to transfer and aggregate request search queries (tasks) through to the task tracker nodes so the end user can perform information analysis on the result.

# 3. SECURITY AND PRIVACY

As big data becomes more popular and networks grow in size and complexity, the issue of security is becoming increasingly crucial. Many cyber-attacks have started to focus on big data and cloud architecture as the payoff can be quite rewarding and devastating at that same time. Being able to store and process the organizations and customer's information in a secure manner is much more difficult and complicated in a big data environment.

The repercussions of big data breach have the potential to cause much more damage than a traditional architecture. Both legal and privacy issues are areas of great concern to a large number of organizations. The obvious and general risks associated with big data environments include: discovering new vulnerabilities when organizations begin implementing a new technology into their work atmosphere, using open-source tools that may contain un-documented vulnerabilities and lack of update options (e.g., backdoors), large cluster node attack surfaces organizations are not prepared to monitor, inadequate knowledge of server hardening, poor user authentication and weak remote access policies, inability to handle large processing of audit/access logs, lack of data validation looking for malicious data input that can become lost in the large volumes of big data. Common infrastructure attacks can include false data injection, DoS, worm/malware propagation, and botnet attacks [10].

In this paper, we explore specific security areas which include big data network intrusion detection, network threat monitoring systems based on MapReduce machine-learning methods, and flow-based anomaly detection.

## 3.1 Intrusion Detection

There are many different types of intrusion challenges, (i.e., virus, malware, Trojans), found in big data security that threaten to exploit the integrity, confidentiality, and availability of network resources [10].

Most intrusion detection systems (IDS) focus on monitoring the individual host or network. They are classified into two types called (i) signature-based and (ii) anomaly-based IDS. The signature IDS detects the attack based on pre-defined maliciously established signatures and the anomaly-based IDS detects the attack by comparing a baseline of a healthy normal network profile to the current network activity looking for deviations in operations.

Each IDS approach has its own pros and cons. For instance, the signature-based IDS is not effective at detecting previously un-established threats, but has a much lower false positive (i.e., false alarm) rate than anomaly-based IDS. In most traditional networks, IDS processes all the network packet traffic and sometimes can end up overloaded. Hence, processing and storing data of big data networks become a great challenge as big data networks volume begins to increase. New studies have shown that "every day, 2.5 quintillion bytes of data are created and 90% of the data in the world today was produced within the past 2 years" [18].

Since IDS are dependent on quick response time and accurate network analysis, current IDS techniques and methods are having a difficult time processing big data's high volume (e.g., inefficiently produced large number of false positives to further impede network performance). Moreover, unstructured varieties of data sets cause traditional machine-learning techniques (i.e., anomaly algorithms) and selecting/extracting (i.e., signature) methods to fail, and the high speed at which the data is arriving into networks decreases network performance dramatically. These big data characteristics make it difficult for current IDS to perform efficiently and process packets with the real-time demands of big data applications. Hence, packets are missed or dropped, which obviously lead to security vulnerabilities in the form of potentially missed security attacks. Table 1 lists the challenges related to each of the big data characteristics.

*Table 1: Intrusion Detection Challenges for Big Data*

| **Big Data Characteristics** | **Related Challenge** |
|---|---|
| Data Volume, i.e., the size of data | Traffic overload and overhead in processing and throughput |
| Data velocity, i.e., the speed of data arrival | Packet loss, cannot analyze missed data packets, and real-time requirements |
| Data variety, i.e., different types of data and complexity of data | Difficult to select and extract appropriate data packets and machine learning, designing knowledge based system is also a great challenge |

### 3.1.1. Solutions

In order to mitigate big data intrusion detection challenge's, it is important to try different approaches to help solve the problem. It may be possible to explore being able to implement new features into intrusion detection that can help use signature matching more flexibly, such as utilizing dynamic parameters in IDS signatures [10]. Advancing stronger machine learning techniques are crucial to big data processing. It is important that knowledge extraction and analysis be in real time, or at least close to it, due to the fact that storage of all the relevant data may not be possible in some cases. In order for the data to have accurate data analysis, it must be processed efficiently at a fast response and real-time classification [10]. There needs to be more innovation in the area of machine-learning algorithms using relevant evaluation metrics to process large volumes of data efficiently.

Because big data is still a new concept that everyone is dealing with, much more research is needed to go into this area of development. It can be also be helpful to introduce additional intrusion detection mechanisms for example: context-aware list-based packet filter and frequency-based exclusive signature matching. The packet filter is located before the IDS to evaluate incoming traffic looking for specific criteria in the established lists (e.g., IP match on the black or white lists), then the traffic move on to the more efficient adaptive character signature matching scheme where each input string (non-sequenced) is analyzed looking for all fixed-size bit string that match an established malicious string signature [10]. Introducing both these additional intrusion security mechanisms actually enhance the performance of the IDS by decreasing the processing time of signature matching while increasing the accuracy of intrusion analysis and resulting in less false positives which waste network performance and time for the security staff.

## 3.2 Threat Monitoring System - MapReduce Machine Learning (MML)

Botnet is a common attack against big data infrastructure where an attacker can exploit various types of malware (e.g., Trojan virus) or even other system vulnerabilities to infect a host's computer and in turn gain control over other compromised systems [10]. By the end of the attack, the adversary can potentially take control over thousands and millions of computers (e.g., zombie network). Now the botnet (i.e., zombie) network can be used to perform other attacks such as Distributed Denial of Service attacks. These types of attacks can become a serious threat to mobile phone networks.

In order to guard against these types of attacks, it is important to implement a cyber threat-monitoring system that is capable of characterizing, tracking, and mitigating the threat in an efficient way [10]. These threat-monitoring detection systems monitors different segments of the system in order to evaluate and compare the states of these segments to pre-defined profiles looking for deviations that will sound an alarm. Threat-monitoring systems such as Advanced Intrusion Detection Environment (AIDE) function by monitoring behavioral changes in hosts and network devices.

The systems collects threat-monitoring logging data (e.g., system, security, application, traffic) created by hosts and network devices (e.g., routers, firewalls) to attempt and gain threat awareness among the environment. In some cases honeypots are installed to gain more interactive information about the attacks. As stated earlier in the report, once the volume of this data becomes too large, it becomes more difficult to process effectively.

### 3.2.1. Solutions

MapReduce Machine Learning (MML) is an effective threat-monitoring method using monitoring agents that are capable of processing real-time data streams and keeping status of hosts/network devices while paying attention for suspicious malicious activity [10]. MML was created to use a distribution method, where computational tasks are distributed across multiple nodes in a cloud environment to help increase the processing time efficiency of the machine-learning process.

In order to accurately and efficiently detect threats, these monitoring systems use the MML schemes to distinguish and profile the characteristics of traffic flows so a learned "classifier" can find the traffic anomalies. The system works by beginning to collect various relevant network metrics and measurable characteristics such as flow duration, average bytes per packet in the flow, and average bytes per second in the flow. Next the system will attempt to detect traffic anomalies using MML schemes.

These schemes are capable of profiling the dynamic characteristics of network traffic flow and detecting anomalies by using learned classifiers and algorithms called logistic regression (LR) and naïve Bayes [10]. Both LR and naïve Bayes are a type of probabilistic statistical classification model, commonly used in binary classification, where traffic flow extracted features/characteristics (e.g. average bytes per packet of the flows) are categorized into two categories (e.g., 1 = monitored data is normal, 0 = monitored data is malicious) [10].

The MML schemes make this threat monitoring method very efficient when dealing with big data because they spread the processing and computing load amongst many different machines. The learned computational results, from the multiple machines, are then processed and combined into an integrated a single classifier which is used to make a conclusion on the malicious vs normal status of the traffic flow [10].

The detection system is broken into an offline training and online detection phase. In the offline training phase, the collected network traffic, consisting of both normal flow and potential attack flows, is stored in a training set. These training sets are organized into different computers to perform the training process independently. Once the computational results from the training (learning) phase using various computers are concluded, they are combined into a single learned classifier. The online detection phase consists of using the learned classifier to evaluate if normal vs malicious traffic flow was found.

## 3.3 Flow-Based NIDS

Monitoring and traffic management of Big Data networks is essential in keeping large data sets moving efficiently in high-speed networks. The research of flow-based anomaly detection has proven to be a great approach due to a few bytes in packet headers that shorten processing time, decreasing privacy concerns, with no issue by encrypted protocols, supported by routers relying on networked systems. Its counterpart packet-based anomaly detection supports payload processing rendering it slow and inefficient for the needs of high-speed networks and great for small networks. Flows efficiently can meter and collect processes to export for monitoring or analysis [3].

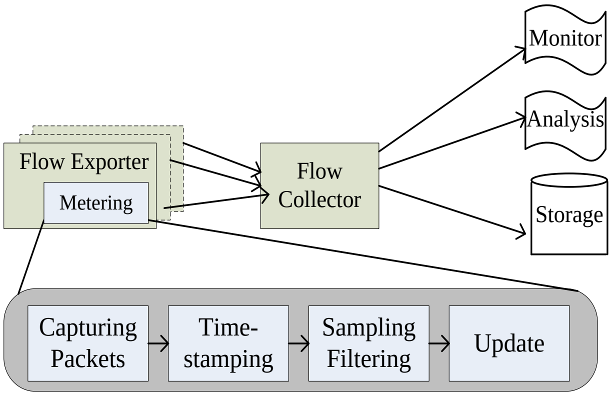

*Fig. 3. Exporting flows from metering specifics through collector architecture*

Flow export and collection is provided by the exporter and collector (shown in Figure 3). Flow exporter and monitoring point meter the creation of flows process to create records that are to be observed. Each packet headers are taken from the flow exporter, marked with a timestamp then processed by sampling-filtering module to be sampled and filtered by administrative specific requirements. By the end the module is updated, flows trigger updates flow entries in cache from packet headers and creating new flows if none exist. Expired flow records export to flow collector when it meets specifications of being idle, maximum allowed lifetime, FIN or RST flags showed, and flow-cache memory exhaustion. The flow collector then exports those flows as a flow record to further be exported to be monitored or analyzed. Attacks relevant to network flow-based methods are:

*Table 2: Attacks and Risk relevant to network flow-based approaches*

| Attack | Description | Risk |
|---|---|---|
| Botnets | Infection hosts controlled by master host | Take control of master node |
| Denial of Service (DoS) | Brute force DoS exhausts resources and overloads networks | Affect resource usages and overloads network blocking users from using services |
| Worms (or Virus) | They explore and find vulnerable network system | Identify vulnerable systems and spread virus into it. |
| Scans | Small probing packets | Create many flows through scanning (through single to many hosts scan) |

There is many published research has been done for flow-based network intrusion detection [3, 16] shows one of the way research can be further improved upon to implement structured approaches [4] using network flow aggregation monitoring to learn of flow-based attacks. Table 2 presents attacks and risks that are related to network flow-based approaches.

### 3.3.1. Solutions

In order to detect a taxonomy approach needs to be classified to focus on classifying[1] the approach taken to succession. In the approach is used with labeling [3] to help trace malicious types, structure its traffic and review malicious activities. Figure 4 shows the processes of packets and logs to labeled data sets.

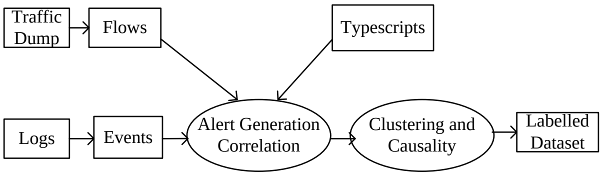

*Fig. 4. Capture process to labeling dataset*

Broken down into flows, alerts, honeypot alerts, and source behavior to process and learn about malicious traffic, side-effects of maliciousness, unknown traffic and uncorrelated alerts through aggregating packets of identical flow to identify any abnormal traffic patterns.

[1] Anna Sperotto has made the approached of flow-based intrusions detection [6] based two main contributors focusing on classification and elements of research by Debar and further built by Axelsson on characteristics, refining their works to better help research direction.

Due to research just touching the surface of flow based IDS many different methods are handy and accuracy and performance is improving. This Fig. 4 taken from [4] categorizes the start of the art flow-based solutions. With the appropriation of passive and centralized solutions along centralized data collection [4] was able to provide reliable results and lessons learned. “Once flow-based monitoring became an established technology; we can see how flows became also a source of data for intrusion detection.” [4] Detection on flow based studies still proves the need for additional research for the automatic tuning of the intrusion detection system.

## 4. DISCUSSION AND CONCLUSION

Along with the network requirements such as network resiliency, congestion, performance consistency, scalability, and partitioning, providing security and privacy must be considered while implementing an infrastructure for Big Data analytics. As current network infrastructures lacks the characteristics to implement Big Data network it is important to implement various tools, methods and techniques into networks in order to help support big data processing such as Map Reduce and Hadoop. These tools and approaches helps manage and process the data more efficiently by breaking up the work to distribute simultaneously across the network.

The current state of the art research in Big Data includes designing network topology, distributed algorithms, integration of software defined networks (SDN), scheduling, optimizations and load balancing among different commodity computers. We also researched security concerns related to big data and how they can be much more severe and difficult to manage than traditional networks. In many cases, traditional forms of security detection with nuanced ideas help effectively identify detecting attacks some prove to be very promising others need more research. Traffic management, monitoring, additional anomaly-detection security strategies such as MapReduce machine learning must be used to help mitigate collective threats such as botnet and DDoS attacks. In order to keep the system packet flows consistent flow-based intrusions detection is sought after. In conclusion, this paper presents an overview of the operation principle of Big Data in the context of security and privacy issues, especially explore network security concerns and mitigation strategies.

## REFERENCES

[1] Hadoop Cluster - Architecture, Core Components and Work-flow” http://saphanatutorial.com/hadoop-cluster-architecture-and-core-components/ Access on February 2016

[2] X. W. Chen and X. Lin, “Big Data Deep Learning: Challenges and Perspectives,” in IEEE Access, vol. 2, pp. 514-525, 2014.

[3] Anna Sperotto, Ramin Sadre, Frank Vliet, and Aiko Pras, “A Labeled Data Set for Flow-Based Intrusion Detection”, In Proceedings of the 9th IEEE International Workshop on IP Operations and Management (IPOM '09), Berlin, Heidelberg, 39-50, 2009.

[4] Anna Sperotto. Flow-Based Intrusion Detection, PhD Thesis, Centre for Telematics and Information Technology, University of Twente, 2010

[5] Kvernvik Tor and Matti Mona. “Applying big-data technologies to network architecture”, Ericsson Review, 2012

[6] Guohui Wang, T.S. Eugene Ng, and Anees Shaikh, “Programming your network at run-time for big data applications”, In Proceedings of the first workshop on Hot topics in software defined networks (HotSDN '12). ACM, New York, NY, USA, 103-108, 2012

[7] Idiro Analytics, “Hadoop”, http://idiro.com/about-idiro/our-technology/ Accessed Web in December 2016

[8] Introduction to big data: infrastructure and networking considerations, Juniper Networks White Paper 2012.

[9] Jain, Raj. “Networking issues for big data”, Class Lecture, Washington University in St. Louis 2013

[10] Lin Xiaodong, Misic Jelena, Shen Xuemin, and Yu Shui, “Networking for big data”, Boca Raton: CRC Press, 2015 Print

[11] Borovick, Lucinda and Villars and L. Richard, “The critical role of the network in big data applications”. Cisco White Paper, April, 2012.

[12] “Big data tutorial: Everything you need to know”. Searchstorage.techtarget.com. Web. 2015. http://searchstorage.techtarget.com/guides/Big-data-tutorial-Everything-you-need-to-know

[13] Rouse Margaret, “MapReduce definition”. Searchcloudcomputing.techtarget.com. Web Accessed on April 2016.

[14] Mearin, Lucas. “World’s data will grow by 50x in next decade, IDS study predicts”. Computerworld.com, Web Accessed on May 2016.

[15] Jeffrey Dean and Sanjay Ghemawat, “MapReduce: simplified data processing on large clusters”, The ACM Communication Magazine, Vol. 51, Issue. 1, pp. 107-113, January 2008

[16] Myung-Sup Kim, Hun-Jeong Kong, Seong-Cheol Hong, Seung-Hwa Chung and J. W. Hong, "A flow-based method for abnormal network traffic detection," Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP, Seoul, South Korea, 2004, pp. 599-612 Vol.1

[17] Jeffrey Dean and Sanjay Ghemawat, “MapReduce: simplified data processing on large clusters”. The ACM Communication Magazine, Volume 51, Issue 1, pp. 107-113, January 2008.

[18] IBM Big Data Report, “Bringing big data to the enterprise”, https://www01.ibm.com/software/in/data/bigdata/ Accessed Web on June 2016.