IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4 , Issue-1 E-ISSN: 2347-2693

Big Data Security – Challenges and Recommendations

Renu Bhandari[1], Vaibhav Hans[2*] and Neelu Jyothi Ahuja[3]

[1,2*]*University of Petroleum and Energy Studies, India*
[3]*Centre of Information Technology, University of Petroleum and Energy Studies, India*

www.ijcseonline.org

Received: Dec/11/2015 Revised: Dec/23/2015 Accepted: Jan/12/2016 Published: Jan/30/ 2016

Abstract— This paper focuses on key insights of big data architecture which somehow lead to top 5 big data security risks and the use of top 5 best practices that should be considered while designing big data solution which can thereby surmount with these risks. Big data architecture, being distributive in nature can undergo partition, replication and distribution among thousands of data and processing nodes for distributed computation thus supporting multiple features associated with big data analytics like real time, streaming and continuous data computation along with massive parallel and powerful programming framework. These series of characteristics are put into effect via a key setup that somehow leads to certain crucial security implications. The challenges induced by this can be handled via big data technologies and solutions that exist inside big data architecture compound characterized for specific big data problems. Big data solutions should provide effective ways to be more proactive against fraud, management and consolidation of data, proper security against data intrusion, malicious attacks and many other fraudulent activities. In particular, this paper discusses the issues and key features that should be taken into consideration while undergoing development of secured big data solutions and technologies that will handle the risks and privacy concerns (e.g. Data security, insecure computation and data storage, invasive marketing etc.) associated with big data analysis in an effective way to increase the performance impact, considering that these risks are somehow a result of characteristics of big data architecture.

Keywords— Big Data; Hadoop; MapReduce; Secure Computation

I. INTRODUCTION

The term big data is coined to describe voluminous amount of unstructured and semi-structured data with three characteristics: 'volume, variety and velocity' considering that the instance, the volume (amount of data), variety (complexity of multiple data types) and velocity (data in motion) of the data produced increases, data is defined as Big Data. Big Data architecture is distributive in nature scaling upto thousands of data and processing nodes. Among these thousands of nodes, the data gets partitioned, replicated and distributed for powerful computation, and because of performance reasons, data is also segmented into classes. Features like auto-tiering, real-time processing and streaming of data have been major trends in big data analysis. A growing number of companies are using the technology to store and analyze petabytes of data including web logs, click stream data and social media content to gain better insights about their customers and their business. Hence, classification of information is becoming more critical.

A number of software firms have been working on applications and solutions related to big data basically designed to bring the power of analytics to the masses. One of the chief risk factor surrounding big data computation and management is the unawareness of the potential future downsides being associated with failure to manage it, clearly making the risk factors transparent for all big data sets – unstructured, structured and all grey areas in between – becoming the top business priorities.

In this paper, we highlight top five security and privacy challenges that are specific to big data. We went through a number of journals, studied published researches, and big data related books to initialize a list of high-priority security and privacy problems and finally arrived at the top five challenges to big data computation along with basic recommendation of how to deal with these problems. The list of challenges is as follows:

1. Insecure Computation
2. Input Validation and Filtering
3. Granular Access Control
4. Insecure Data Storage
5. Privacy concerns in Data Mining and Analytics.

II. KEY INSIGHTS OF BIG DATA ARCHITECTURE

Big Data Architecture is premised on a set of skills for the development of scalable, reliable, and completely automated pipelines of huge amount of data. Big data is still till date quite a young field which therefore leads to 'no standard big data architecture available' that has been used for a long time. Properties like latency, volume, velocity, variety, veracity, capability for ad-hoc queries, scalability,

robustness and fault tolerance have become key features that are mandatory for choosing any big data architecture. Although some intrinsic properties like auto- tiering, easy shift of code through disks for analysis have been into practice for a long time, these also undergo some security issues. These issues have been discussed thoroughly in the coming sections.

A. *Basic Big Data Framework*

Big Data Architecture is distributed in nature and can scale up to thousands of data and processing nodes. In big data architecture the data is partitioned, replicated and distributed among those thousands of nodes. Because of performance reasons, data is partitioned into two classes – hot data and cold data giving a nice feature to big data architecture called auto–tiering [6]. Easy move of code through disks rather than the data has also been another big paradigm shift from traditional to modern architecture. Modern architecture supports real time computation (real – time analytics), collection of data from variety of input sources [fig. 2] and transfer of that data to the big data solutions on a regular basis. Ad-hoc queries along with massive parallel and powerful programming layout provides flexibility. Many other frameworks like mapreduce where a program gets divided into multiple maps that get executed on respective data nodes and when finally they are reduced to a single result set, Storm Topology (Spouts & Bolts) ,

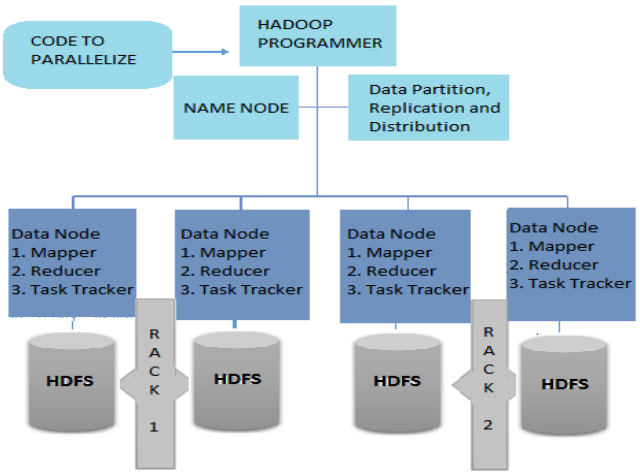

Fig. 1 DISTRIBUTED ARCHITECTURE

where Spouts are data sources and Bolts are data processing nodes following network topology for real time computation are being used. However there is no single silver bullet as Hadoop is already unsuitable for many big data problems like real time analytics, graph computation, low latency queries etc. and Storm topology.

B. *Auto-tiering*

Storage arrays have become good at managing huge amount of data that can dynamically move between different disk types and RAID levels in an array. Automated Storage Tiering, a storage software management feature, manages the need of maintaining space, performance and cost requirements for data processing. General policies are set-up by storage administrators that deal with partition of data into two major classes followed by assigning of memory addresses to data according to their respective classes.

For performance reasons, data is partitioned into two classes, hot data and cold data. Data that is used frequently, say for analysis or prediction analysis is classified as hot data. On the other hand, data that is used temporarily or less frequently falls under the category of cold data. Cold data is assigned to slower, less-expensive SATA storage, and hot data is moved to high-performing SAS or SSDs. However, data is automatically classified and migrated to the optimum tier of storage disk drives as its activity level rises or falls away. Auto tiering has security implications, there comes a certain amount of time when you do not know where the data resides.

C. *Real-Time, Streaming and Continuous Computation.*

Performing real time computation is the next big trend in big data. With applications like google analytics, real time monitoring of websites, webpages can be done. Huge amount of terabytes of data is collected from various sources, filtered, analyzed via multiple data mining, data classification and prediction algorithms and hence reports are maintained of all these analysis. These reports thereby help in decision making for better performance of organizations. Stream Processing Language is a real – time data processing language used to process data streams coming from multiple sources [6]. SPL comes with three basic types of operators – Utility, Relational and Arithmetic which take data through Input Source Operator and give output through Output Source Operators. These multiple operators present in between the source filter, aggregate, join multiple data streams accordance to the need of the user. Arrangements of the operators can be done manually by the users as per the requirements. This adds-up as a more efficient way of processing streaming data.

Another key feature of big data is that it supports ad-hoc queries. As an example from traditional database is that, end-users are able to create SQL queries at their end and are able to submit them to respective web applications directly. This will definitely give more flexibility, more power. But there are greater security concerns of having ad-hoc queries.

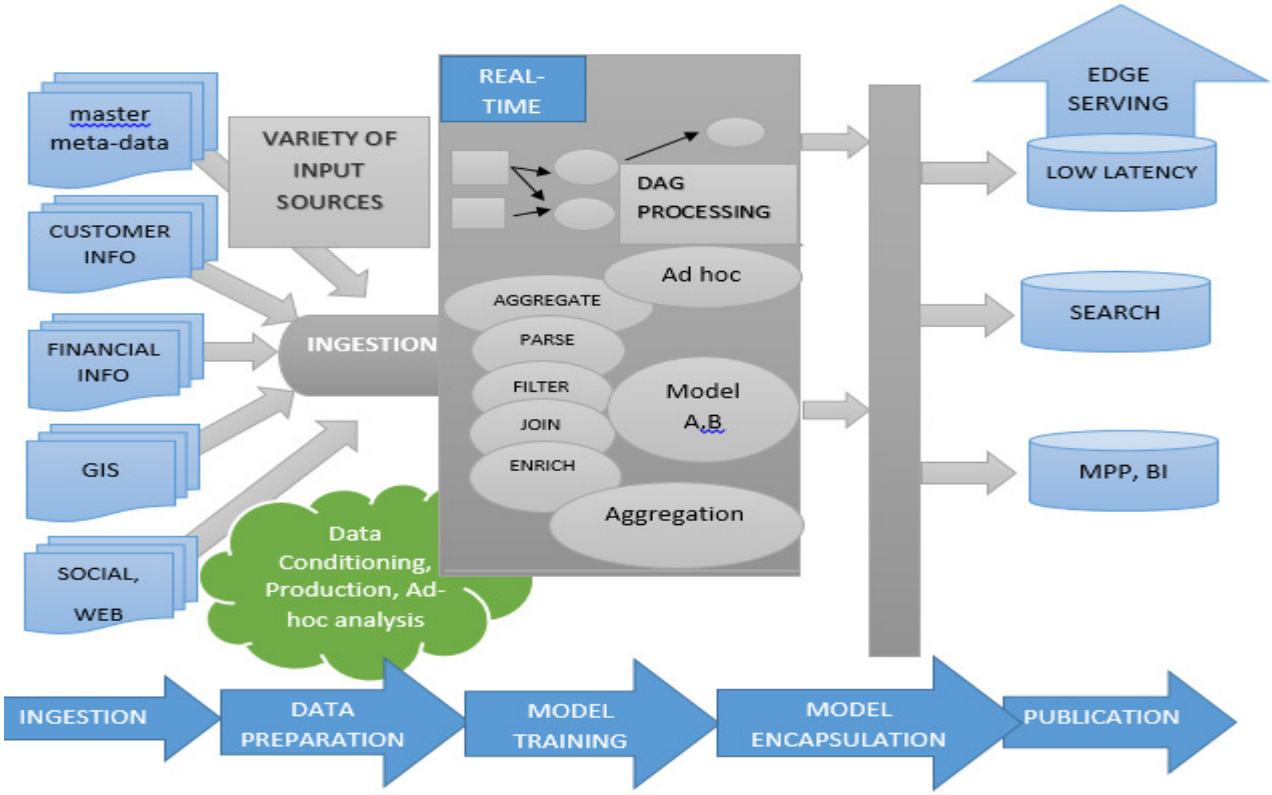

Fig. 2 REAL-TIME STREAMING, AND CONTINUOUS COMPUTATION

D. Parallel and Powerful Programming Framework

Big Data has massive parallel and powerful programming framework. Suppose if you have 16 TB of data, and the data is divided into 128 MB chunks, then your program will be divided into 82000 Maps or functions that will run concurrently on data processing nodes. Big Data Programming framework is very powerful. User can develop program in Java rather than SQL/PLSQL databases [6].

There are multiple frameworks available for Big data computation. MapReduce framework is used by Hadoop where a program gets divided into multiple map that get executed at multiple data nodes and then finally the results are merged together into single result set [3].

Topology based computation is yet another major framework associated with big data computation. This paradigm is used by a real time analysis big data solution called Storm. Storm utilizes a network topology of Spouts and Bolts. Here Spouts act as data sources, whereas Bolts act as data processing nodes. A lot of frameworks and topologies exist for data processing and management in the field of big data computation.

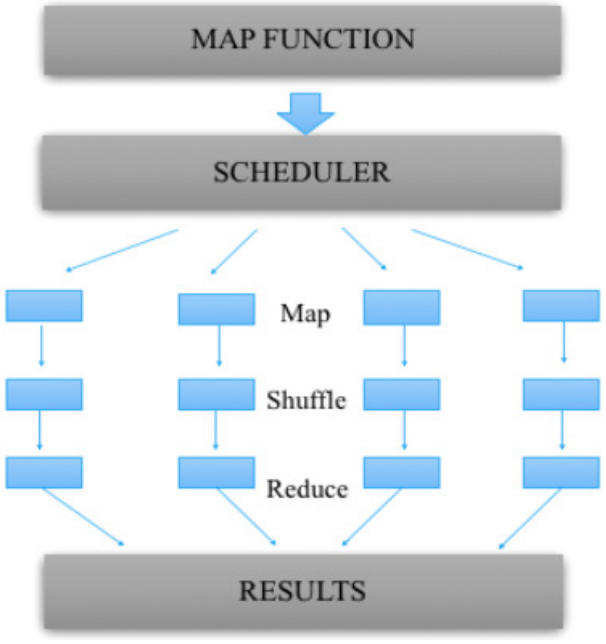

Fig. 3 MAPREDUCE FRAMEWORK

III. TOP 5 SECURITY RISKS

The security mechanisms in big data technology is generally weak. Hence implementing the above 3 features in big data architecture has been inevitable and of a big concern. Finding robust security mechanisms for the purpose of using features like auto – tiering, parallelism etc. has been a challenging problem. Issues like invasion of privacy, complexity of disk drive storage, invasive marketing etc. have led to challenges in implementing Big Data Analytics tools for Big data solutions and applications.

A. Insecure Computation

Untrusted computational programs are used by attackers in order to extract and turnout sensitive information from data sources. Insecure computation apart from causing information leak can also corrupt your data, leading to incorrect results in prediction or analysis. It can also result into Denial of Services (DoS) on your big data solution disabling the property of using massively parallel programming language.

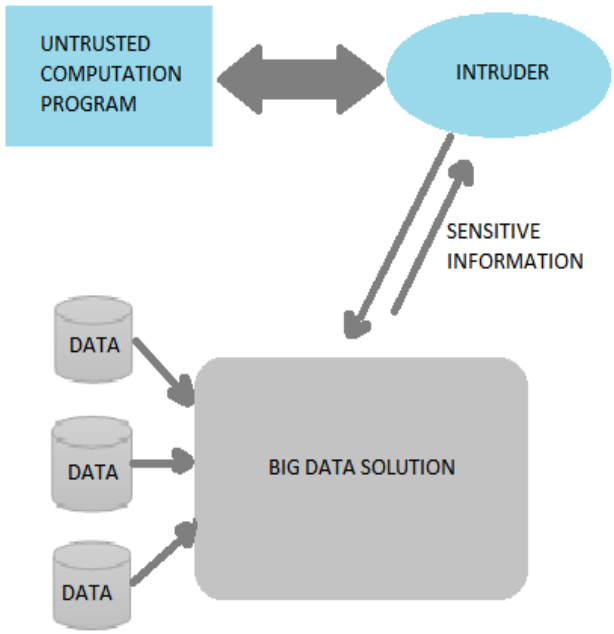

Fig. 4 INSECURE COMPUTATION

B. Input Validation and Filtering

Big Data needs to collect input from variety of sources, therefore it is quite important and mandatory to validate the input [4]. This involves making a decision of what kind of data is untrusted and what are the untrusted data sources. It also needs to filter rogue or malicious data from the good one. These two challenges are not new, i.e. these two challenges are also present in traditional databases, however in big data the huge amount of gigabytes and terabytes of continuous data flow makes it really very difficult to perform input validation or data filtering on the incoming batch of data [5]. Signature based data filtering also has limitations like it cannot filter rogue or malicious data having some behavioral aspect. When a large amount of malicious data is inserted into the dataset, its influence on the result produced is *massive*. Signature based data filtering is incapable of tracking down such attacks, thus individual custom algorithms need to be designed to deal with such cases.

C. Granular Access Controls

Big data was traditionally designed for performance and scalability with almost no security in mind. Traditional databases have very comprehensive table, row and cell level access control, and these have been really gone missing in big data solutions. Ad-hoc queries pose another additional challenge to big data solutions where user can retrieve sensitive information out of the data using ad-hoc queries. Even though being provided by a big data solution, access control is disabled by default. EX: NO-SQL databases depends upon access control provided by third party softwares, but it is actually disabled by default and you need to enable it explicitly.

D. INSECURE DATA STORAGE

As data is stored at thousands of nodes – 'authentication, authorization and encryption of data at those nodes becomes a challenging work'. Auto–tiering moves cold data to lesser secure medium – 'What if cold data is sensitive?' .If any solution provides encryption of real time data, it may not be useful, as encryption of real time data may have performance impacts. Secure communication amongst various nodes, middlewares, and end users is disabled by default, hence it needs to be enabled explicitly [4].

E. Privacy Concerns in Data Mining and Analytics

Monetization of Big Data involves Data Mining and Analytics and sharing of those analytical results involves multiple challenges like invasion of privacy, invasive marketing and unintentional disclosure of information. Quite a few examples of these include - AOL release of anonymized search logs where users could be identified easily, which is really concerning.

IV. TOP 5 BEST PRACTICES.

Traditional security mechanisms that focus on securing small scale data that is basically static (not streaming), are practically inadequate. Big data security issues are magnified with the velocity, volume and variety of big data. Big data processing requires ultra-fast response times for computation, and adding up multiple feature to the framework adds up the security implications. However,

these security issues can be handled by taking proper adequate steps. Five of the best recommendations for big data security have been discussed below.

A. *Secure Your Computation Code*

In order to prevent malicious data from entering your big data solution, implement access control, code signing and dynamic analysis of the computational code. Proper strategies need to be made having the capability of controlling the impact of untrusted code on the data, once it has been able to get into the big data solution. There are generally two ways of preventing attacks: securing the data when insecure mapper is present, and securing the mapper (related to MapReduce framework of Big Data Hadoop).

B. *Implement Comperihensive Input Validation and Filtering.*

For better processing and security practices, implementation of comprehensive input validation and filtering on almost all internal and external sources is mandatory. On the other hand, proper evaluation of key input validation and filtering features of respective big data solution is required to see if it can scale up the data requirement for respective big data solutions [5]. This may be implemented by building algorithms that will validate the input for large sets of data.

C. *Implement Granular Access Control.*

Reviewing and configuring the role and privilege matrix of different the kinds of users of big data which can be the admin, knowledge workers, end users, developers etc. is the core part for the implementation of granular access control. Ad-hoc queries are required for data computation, and the queries processed should be verified for what access has to be given to the respective ad-hoc queries [5]. By default the access control is disabled, it needs to get enabled explicitly for proper access to the data and its sources. This section of recommendation in simple words means preventing the data to be accessed by the people that should not have access to the data.

D. *Secure your data storage and computation.*

Protecting data storage and computation sections of big data analytics becomes a prime area to focus on as much part of sensitive data leakage portions are encountered in this phase. For this, the sensitive data should be segregated. Enabling Data Encryption for sensitive data and audit administrative access on Data Nodes marks to be a major step in this scenario [5]. And finally the verification of proper configuration of API security of all components of big marks to be the final step for secure data storage and computation.

E. *Review and Implement Privacy Preserving Data Mining and Analytics.*

For proper preservation of sensitive information, verification of analytical algorithms designed for data mining, pattern classification and recognition is necessary. This will reduce the rate of disclosure of sensitive information. Before any further actions, big data implementation should be pen tested. It is important to establish guidelines and recommendations for the prevention of inadvertent privacy disclosures.

CONCLUSION

Big data is trending. No new application can be imagined without it producing new forms of data, operating on data-driven algorithms, and consuming specified amount of data. With data storing and computing environments becoming more cheaper, cloud environments becoming more capable of storing and sharing system and analytics applications, software applications becoming more networked, data security, access control, compression – encryption and compliance have introduced several challenges that practically need to be handled and addressed in a very systematic manner.

In this paper, we have walked through the top five big data security challenges and have laid some recommendations for making big data processing and computation more reliable and in turn making its infrastructure more secure. Some common elements in this list of the top five security issues that are specific to big data arise from the multiple infrastructure tiers – (both computing and storage) used for big data processing, the new computation infrastructures like NoSQL databases used for fast throughput that are necessary for big volumes of data are not thoroughly secured from major security threats, the non – scalability of real – time monitoring techniques, the heterogeneous layout of devices producing data, confusion with diverse legal restrictions that somehow lead to ad-hoc approaches for security and privacy. There is a big ecosystem existing for specific big data problems. Topics in this paper serve to clarify specific aspects of the vulnerable areas in the entire big data processing infrastructure that need to be analyzed for certain threats. Major recommendations for dealing with the top five security risks have also been suggested in this paper.

Our hope is that this paper will collaboratively increase the focus of the research and development community towards the top five challenges, which will ultimately lead to greater security and privacy in respective big data platforms.

ACKNOWLEDGMENTS

We would like to acknowledge our fellow colleagues and researchers for their work and support.

REFERENCES

[1] Big data. In *Wikipedia, The Free Encyclopedia*. Retrieved 08:36, November10, **2015**
[2] Apache Hadoop. In *Wikipedia, The Free Encyclopedia*. Retrieved 10:28,November 20,**2015**
[3] MapReduce. In *Wikipedia, The Free Encyclopedia*. Retrieved 08:43, January 15, **2016**
[4] IBM Security Intelligence with Big Data, In IBM. Retrieved 09:38, November 22, **2015**
[5] Big Data Research, Security in big data research papers, Retrieved 08:10, December 10,**2015**
[6] Anuja Pandit, Amruta Deshpande and Prajakta Karmarkar, Log Mining Based on Hadoop's Map and Reduce Technique, Int. Journal of Computer Sciences and Engineering, Volume -05, Issue -04, Page No (1-4), April **2013**

AUTHORS PROFILE

Ms. Renu Bhandari
Renu Bhandari, an undergraduate scholar from University of Petroleum and Energy Studies is pursuing her degree in Computer Science with specialization in Business Analytics and Optimization. She has worked on many analytics based projects that involve innovation and R&D. Her research interest is in the area of Data Mining and Analytics and she is currently working as an intern in Indian Statistical Institute, Bangalore. Her work and experience strongly deal with Machine Learning, Big Data, Data Mining, and Python.

Mr. Vaibhav Hans
Vaibhav Hans is a student of University of Petroleum & Energy Studies. He's pursuing his degree in Computer Science Engineering with specialisation in Business Analytics. He's currently woking in upcoming field of Analytics and Internet of Things. His areas of expertise is object oriented programming, Machine Learning Data Analytics & Big Data.

Dr. Neelu Jyothi Ahuja
Neelu Jyothi Ahuja has received her PhD on development of a rule based expert system for seismic data interpretation, from University of Petroleum and Energy Studies, Dehradun. She has 17 years of experience in teaching, research and project proposal development and has published papers in journals and conferences at international and national level. She is currently executing 03 research projects and supervising doctoral work of 04 scholars. With keen interest in intra-disciplinary research, her research interests include Expert Systems, Knowledge-based tutoring, Artificial Intelligence, Object oriented development and Programming Languages. She is head computing research institute, a virtual R & D center and an Associate Professor in computer science and engineering department, at College of Engineering Studies at University of Petroleum and Energy Studies, Dehradun.