IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.6, Special Issue.11, Dec 2018 E-ISSN: 2347-2693

Anchoring Your Big Data Environment

S. Regha[1*], M. Manimekalai[2]

[1,2]Dept. of Computer Science, Shrimati Indira Gandhi College, Tiruchirappalli, India

Available online at: www.ijcseonline.org

Abstract— Security and protection issues are amplified by the volume, assortment, and speed of Big Data. The decent variety of information sources, arrangements, and information streams, joinExternalize data security when possible and d with the gushing idea of information procurement and high volume make one of kind security dangers. This paper points of interest the security challenges when associations begin moving touchy information to a Big Data store like Hadoop. It distinguishes the diverse danger models and the security control structure to address and alleviate security hazards because of the recognized risk conditions and use models. The system laid out in this paper is likewise intended to be circulation skeptic.

Keywords— Hadoop, Big Data, enterprise, defense, risk, Big Data Reference Framework, Security and Privacy, threat model

I. INTRODUCTION

The term "Big Data" refers to the massive measures of digital data that companies collect. Industry estimates on the growth rate of information is roughly double every two years, from 2500 Exabytes in 2012 to 40,000 Exabytes in 2020 [1]. Big information is definitely not a specific technology. It is a collection of attributes and capabilities.

NIST defines Big Data as the following [2]: Big Data comprises of extensive datasets, principally in the characteristics of volume, velocity, or potentially variety that require a scalable architecture for efficient storage, control, and investigation. Securosis research [3] includes extra characteristics for a specific environment to qualify as 'Big Data'.

1. It handles a petabyte of information or more
2. It has distributed redundant information storage
3. Can leverage parallel undertaking processing
4. Can provide information processing (MapReduce or equivalent) capabilities
5. Has extremely quick information insertion
6. Has central management and orchestration
7. Is hardware agnostic

Is extensible where its fundamental capabilities can be augmented and altered Security and protection issues are magnified by the volume, variety, and velocity of Big Data.

The diversity of information sources, arrangements, and information streams, combined with the streaming nature of information obtaining and high volume create unique security dangers.

It isn't merely the existence of large measures of information that is creating new security challenges for organizations. Big Data has been collected and utilized by enterprises for several decades. Software infrastructures, for example, Hadoop enable developers and experts to easily leverage hundreds of computing nodes to perform information parallel computing which was not there before. As a result, new security challenges have arisen from the coupling of Big Data with heterogeneous organizations of ware hardware with ware operating systems, and item software infrastructures for storing and computing on information. As Big Data expands at the different enterprises, customary security mechanisms tailored to securing little scale, static information and information streams on firewalled and semi isolated networks are inadequate. So also, it is unclear how to retrofit provenance in an enterprise's existing infrastructure. Throughout this document, unless explicitly called out, Big Data will refer to the Hadoop framework and its normal NoSQL variations (e.g. Cassandra, MongoDB, Couch, Riak, etc.). This paper details the security challenges when organizations begin moving sensitive information to a Big Data repository like Hadoop. It provides the different threat models and the security control framework to address and mitigate the hazard due to the identified security threats. In the following sections, the paper describes in the detail the architecture of the modern Hadoop ecosystem and identify the different security weaknesses of such systems. We then identify the different threat conditions associated with them and their threat models. This paper concludes the examination by providing a reference security framework for an enterprise Big Data environment.

Hadoop Security Weakness

Conventional Relational Database Management Systems (RDBMS) security has evolved over the years and with many 'eyeballs' assessing the security through different security

evaluations. Unlike such arrangements, Hadoop security has not undergone the same level of rigor or evaluation for that matter and subsequently can assert little assurance of the implemented security.

Another big challenge is that today, there is no institutionalization or convey ability of security controls between the different Open-Source Software (OSS) projects and the different Hadoop or Big Data vendors. Hadoop security is completely fragmented. This is true even when the above parties implement the same security feature for the same Hadoop component. Vendors and OSS parties' force-fit security into the Apache Hadoop framework.

Top 10 Security & Privacy Challenges

The Cloud Security Alliance Big Data Security Working Group has compiled the following as the Top 10 security and privacy challenges to overcome in Big Data [4].

1. Secure computations in distributed programming frameworks
2. Security best practices for non-relational data stores
3. Secure data storage and transactions logs
4. End-point input validation/filtering
5. Real-time security monitoring
6. Scalable privacy-preserving data mining and analytics
7. Cryptographically enforced data centric security
8. Granular access control
9. Granular audits
10. Data provenance

The above challenges were grouped into four broad components by the Cloud Security Alliance. They were:

- Infrastructure Security
- Secure computations in distributed programming frameworks
- Security best practices for non-relational data stores Data Privacy
- Scalable privacy-preserving data mining and analytics
- Cryptographically enforced data centric security
- Granular access control Data Management
- Secure data storage and transactions logs
- Granular audits
- Data provenance Integrity & Reactive Security
- End-point input validation/filtering
- Real-time security monitoring

Fig. 1

II. ADDITIONAL SECURITY WEAKNESSES

The earlier section regarding Cloud Security Alliance list is an excellent begin and this research and paper significantly adds to it. Where possible, effort has been made to outline to the categories identified in the CSA work. This section records some extra security weaknesses associated with Open Source Software (OSS) like Apache Hadoop. It is meant to give the reader an idea of the possible assault surface. However it's not meant to be exhaustive which subsequent sections will provide and add to.

Infrastructure Security and Integrity The Common Vulnerabilities and Exposures (CVE) database just shows four reporting and fixed Hadoop vulnerabilities over the previous three years. Software, even Hadoop, is a long way from perfect. This could either reflect that the security network isn't active or that the vast majority of vulnerability remediation happens internally inside the vendor environments themselves with no open reporting. Hadoop security configuration files are not selfcontained with no legitimacy checks preceding such policies being deployed. This as a rule results in information integrity and accessibility issues.

Identity and Access Management

- Role Based Access Control (RBAC) approach files and Access Control Lists (ACLs) for components like MapReduce and HBase are generally configured through clear-text files. These files are editable by privileged records on the system like root and other application accounts.

Information Privacy and Security

- All issues associated with SQL injection type of assaults don't go away. They move with Hadoop components

like Hive and Impala. SQL prepare capacities are currently not available which would have enabled separation of the query and information

- Lack of native cryptographic controls for sensitive information protection. Frequently, such security is provided outside the information or application stack. Clear-text information might be sent when communicating between Data Node to DataNode since information region can't be entirely enforced and the scheduler might not be able to discover resources next to the information and force it to read information over the network.

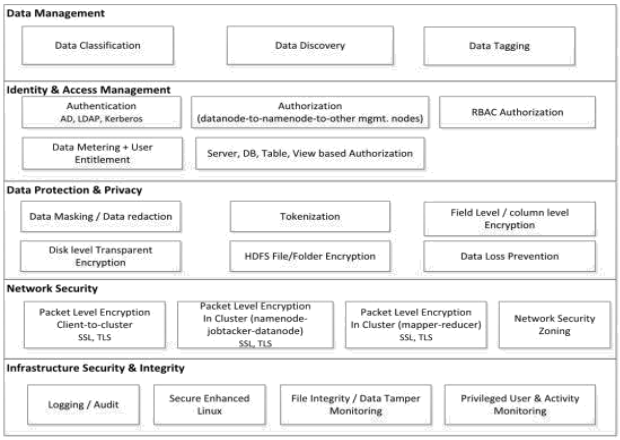

Figure 2: Big Data Security Framework

Big Data Security Framework The following section provides the target security architecture framework for Big Data stage security. The core components of the proposed Big Data Security Framework are the following:
1. Information Management
2. Identity and Access Management
3. Information Protection and Privacy
4. Network Security 5. Infrastructure Security and Integrity

The above '5 mainstays' of Big Data Security Framework are further decomposed into 21 sub-components, each of which are basic to ensuring the security and mitigating the security hazard and threat vectors to the Big Data stack. The overall security framework is demonstrated as follows.

III. DATA MANAGEMENT

Data Management component is decomposed into three core sub-components. They are Data Classification, Data Discovery, and Data Tagging.

3.1.1 Data Classification Effective information grouping is presumably one of the most critical activities that can thus lead to effective security control implementation in a Big Data stage. When organizations deal with an extremely large measure of information, otherwise known as Big Data, by clearly being able to identify what information matters, what needs cryptographic protection among others, and what fields need to be prioritized first for protection, more often than not determine the success of a security initiative on this stage.

The following are the core items that have been developed over time and can lead to a successful information arrangement lattice of your environment.
1. Work with your legal, protection office, Intellectual Property, Finance, and Information Security to determine every single unmistakable datum fields. An open bucket like health information isn't sufficient. This exercise encourages the reader to go beyond the representative arrangement level exercise.

2. Perform a security control assessment exercise. a. Determine area of information (e.g. exposed to internet, secure information zone) b. Determine number of users and systems with access c. Determine security controls (e.g. would it be able to be protected cryptographically)

3. Determine value of the information to the attacker a. Is the information easy to resell on the bootleg market? b. Do you have valuable Intellectual Property (e.g. a country state looking for nuclear reactor blueprints)

4. Determine Compliance and Revenue Impact
a. Determine breach reporting requirements for all the unmistakable fields b. Does loss of a specific information field prevent you from doing business (e.g. card holder information) c. Estimate re-architecting cost for current systems (e.g. buying new security items) d. Other costs like more frequent auditing, fines and judgements and legal expenses related to compliance.

5. Determine effect to the owner of the PII information (e.g. a customer) a. Does the field cause phishing assaults (e.g. email) versus simply replace it (e.g. loss of a credit card)

3.1.2 Data Discovery The absence of situational awareness with respect to sensitive information could leave an organization exposed to significant dangers. Identifying whether sensitive information is present in Hadoop, where it is located and subsequently triggering the appropriate information protection measures, for example, information masking, information redaction, tokenization or encryption is key.
- For structured information going into Hadoop, for example, relational information from databases, or, for example, comma-separated values (CSV) or JavaScript Object Notation (JSON)- formatted files, the area and grouping of sensitive information may already be known. In this case, the protection of those segments or fields can happen programmatically, with, for example, a labeling

engine that assigns perceivability labels/cell level security to those fields.

- With unstructured information, the area, check and order of sensitive information becomes significantly more troublesome.

Information discovery, where sensitive information can be identified and located, becomes a critical initial phase in information protection.

The following items are urgent for an effective information discovery exercise of your Big Data environment:

1. Define and validate the information structure and schema. This is all useful prep work for information protection activities later

2. Collect metrics (e.g. volume tallies, unique checks etc.). For example, if a file has 1M records however it is duplicate of a single person, it is a single record versus 1M records. This is very useful for compliance however more imperatively chance management.

Share this insight with your Data Science teams for them to fabricate threat models, profiles which will be useful in information exfiltration prevention scenarios.

4. On the off chance that you discover sequence files, work with your application teams to move far from this information structure. Instead leverage columnar storage arrangements, for example, Apache Parquet where possible regardless of the information processing framework, information mode, or programming language.

5. Fabricate contingent search routines (e.g. just report on date of birth if a person's name is found or Credit Card # + CVV or CC +zip)

6. Record for usecases where once sensitive information has been cryptographically protected (e.g. encrypted or tokenized), what is the usecase for the discovery arrangement.

3.1.3 Data Tagging Understand the end-to-end information streams in your Big Data environment, especially the ingress and egress methods.

1. Identify every one of the information ingress methods in your Big Data cluster. These would include all manual (e.g. Hadoop administrators) or automated methods (e.g. ETL employments) or those that go through some meta-layer (e.g. duplicate files or create + write).

2. Knowing whether information is coming in leveraging Command Line Interface or though some Java API or through Flume or Sqoop import of in the event that it is being SSH'd in is essential.

3. Likewise, take after the information out and identify all the egress components out of your Big Data environment. 4. This includes whether reporting occupations are being gone through Hive queries (e.g. through ODBC/JDBC), through Pig occupations (e.g. reading files or Hive tables or HCatalog), or exporting it out by means of Sqoop or copying through REST API, Hue etc. will determine your control boundaries and trust zones. 5. The majority of the above will likewise help in information discovery movement and other information access management exercises (e.g. to implement RBAC, ABAC, etc.)

3.2 Identity and Access Management POSIX-style permissions in secure HDFS are the reason for some, access controls over the Hadoop stack.

3.2.1 User Entitlement + Data Metering Provide users access to information by centrally managing access policies. It is imperative to tie strategy to information and not to the access method Leverage Attribute based access control and protect information based on tags that move with the information through lineage; permissions decisions can leverage the user, environment (e.g. area), and information attributes. Perform information metering by restricting access to information once a typical threshold (as determined by access models + machine learning algorithms) is passed for a specific user/application.

3.2.2 RBAC Authorization Deliver fine-grained approval through Role Based Access Control (RBAC)

Manage information access by role (and not user) Determine relationships between users and roles through groups. Leverage AD/LDAP group membership and enforce rules over all information access ways

3.3 Data Protection and Privacy most of the Hadoop conveyances and vendor additional items package either information at-rest encryption at a square or (whole) file level. Application level cryptographic protection (like field-level/section level encryption, information tokenization, and information redaction/masking provide the next level of security needed.

3.3.1 Application Level Cryptography (Tokenization, field-level encryption) While encryption at the field/element level can offer security granularity and review tracking capabilities, it comes at the expense of requiring manual intervention to determine the fields that require encryption and where and how to enable authorized decryption.

3.3.2 Transparent Encryption (circle/HDFS layer) Full Disk Encryption (FDE) prevents access by means of the storage medium. File encryption can likewise guard against (privileged) access at the node's operating-system level. In case you need to store and process sensitive or regulated information in Hadoop, information at-rest encryption protects your organization's 6 sensitive information and keeps in any event the circles out of review scope. In larger

Hadoop clusters, plates often need to be removed from the cluster and replaced. Plate Level transparent encryption ensures that no intelligible residual information remains when information is removed or when circles are decommissioned. Full-circle encryption (FDE) can likewise be OSnative plate encryption, for example, dm-tomb

3.3.3 Data Masking/Data Redaction Data masking or information redaction before stack in the normal ETL process de-identifies personally identifiable data (PII) information before stack. Therefore, no sensitive information is stored in Hadoop, keeping the Hadoop Cluster potentially out of (review) scope. This might be performed in group or real time and can be achieved with a variety of designs, including the use of static and dynamic information masking instruments, and in addition through information services.

3.4 Network Security The Network Security layer is decomposed into four sub-components. They are information protection in-travel and network zoning + approval components.

3.4.1 Data Protection In-Transit Secure correspondences are required for HDFS to protect information in-travel. There are multiple threat scenarios that thus mandate the necessity for https and prevent data disclosure or elevation of privilege threat categories. Using the TLS convention (which is currently available in all Hadoop conveyances) to authenticate and ensure security of correspondences between nodes, name servers, and applications. An attacker can gain unauthorized access to information by intercepting correspondences to Hadoop consoles. This could include correspondence between Name Nodes and Data Nodes that are in the clear back to the Hadoop clients and thus can result in credentials/information to be sniffed. Tokens that are granted to the user postKerberos authentication can likewise be sniffed and can be used to impersonate users on the NameNode.

Following are the controls that when implemented in a Big Data cluster can ensure properties of information confidentiality.
1. Packet level encryption using TLS from the client to Hadoop cluster 2. Packet level encryption using TLS inside the cluster itself. This includes using https between NameMode to Job Tracker to Data Node. 3. Packet level encryption using TLS in the cluster (e.g. mapper-reducer) 4. Use LDAP over SSL (LDAPS) rather than LDAP when communicating with the corporate enterprise directories to prevent sniffing assaults. 5. Permit your administrators to configure and enable encrypted shuffle and TLS/https for HDFS, MapReduce, YARN, HBase UIs etc.

3.4.2 Network Security Zoning The Hadoop clusters must be segmented into purposes of delivery (PODs) with chokepoints, for example, Top of Rack (ToR) switches where network Access Control Lists (ACLs) confine the allowed activity to approved levels.
End users must not be able to connect to the individual information nodes, however to the name nodes as it were.
The Apache Knox gateway for example, provides the capacity to control movement all through Hadoop at the per-service-level granularity. An essential firewall that ought to permit access just to the Hadoop NameNode, or, where sufficient, to an Apache Knox gateway. Clients will never need to communicate directly with, for example, a DataNode.

3.5 Infrastructure Security and Integrity The Infrastructure Security and Integrity layer is decomposed into four core sub - components. They are Logging/Audit, Secure Enhanced Linux, File Integrity + Data Tamper Monitoring, and Privileged User and Activity Monitoring.

3.5.1 Logging/Audit All system/ecosystem changes unique to Hadoop clusters need to be audited with the review logs being protected. Examples include: Addition/deletion of information and management nodes Changes in management node states including work tracker nodes, name nodes Pre-shared secrets or certificates that are rolled out when the underlying package of the Hadoopdispersion or of the security arrangement is pushed to the node prevent the expansion of unauthorized cluster nodes. When information isn't limited to one of the core Hadoop components, Hadoop information security ends up having numerous moving parts and high percentage of fragmentation. Consequently, there results a sprawl of metadata and review logs over all fragments. In a regular enterprise, the DBAs are ordinarily leveraged to place the security responsibility at the table, line, segment, or cell level and keeping in mind that the configuration of file systems and system overseers, and the Security Access Control team is generally accountable for the more granular file level permissions.

Yet, in Hadoop, POSIX-style HDFS permissions are frequently vital for information security or are at times the main means to enforce information security by any means. This leads to questions concerning the manageability of Hadoop security.

Technologies recommendations to address information fragmentation: Apache Falcon is an incubating Apache OSS project that focuses on information management. It provides graphical information lineage and actively controls the information life cycle. Metadata is retrieved and mashed up from wherever the Hadoop application stores it. Cloudera Navigator is a proprietary instrument and GUI that is a piece of Cloudera's Distribution Including Apache Hadoop (CDH) dispersion. CDH Navigator is a device to address log sprawl, lineage and some aspects of information discovery. Metadata

is retrieved and mashed up from wherever the Hadoop application stores it. Zettaset Orchestrator is an item to harness the overall fragmentation ofHadoop security with a proprietary combined GUI and work process. Zettaset has its own metadata repository where metadata from all Hadoop components is collected and stored.

3.5.2 Secure Enhanced Linux (SELinux) SELinux was created by the United States National Security Agency (NSA) as a set of patches to the Linux Kernel using Linux Security Modules (LSM). It was eventually released by the NSA under the GPL license and has been adopted by the upstream Linux kernel. SELinux is an example of a Mandatory Access Control (MAC) for Linux. Truly Hadoop and other Big Data stages based over Linux and UNIX systems have had discretionary access control. What this means for example is that a privileged user like root is omnipotent. By enforcing and configuring SELinux on your Big Data environment, through MAC, there is arrangement which is administratively set and fixed. Even if a user changes any settings on their home directory, the arrangement prevents another user or process from accessing it. A sample strategy for example that can be implemented is to make library files executable however not writable or vice-versa. Employments can write to/tmp area however not be able to execute anything in there. This is a great method to prevent summon injection assaults among others. With policies configured, even in the event that someone who is a sysadmin or a noxious user can gain access to root using SSH or some other assault vector, they might be able to read and write a considerable measure of stuff. However, they won't be able to execute anything incl. potentially any information exfiltration methods. The general recommendation is to run SELinux is permissive mode with regular workloads on your cluster, reflecting commonplace usage, including using any apparatuses. The warnings generated would then be able to be used to define the SELinux arrangement which after tuning can be deployed in a 'targeted enforcement' mode.

IV. FINAL RECOMMENDATIONS

The following are some key recommendations in helping mitigate the security dangers and threats identified in the Big Data ecosystem.

1. Select items and vendors that have proven experience in comparative scale deployments. Request vendor references for large deployments (that is, comparable in size to your organization) that have been running the security controls under consideration for your project for no less than one year 2. Key columns are: Accountability, balancing network centric, access-control centric, and information centric security is absolutely basic in achieving a good overall reliable security posture.

2. Information centric security, for example, label security or cell-level security for sensitive information is preferred. Label security and cell-level security are integrated into the information or into the application code rather than adding information security after the reality 4. Externalize information security when possible and use information redaction, information masking or tokenization at the time of ingestion, or use information services with granular controls to access Hadoop 5. Harness the log and review sprawl with information management instruments, for example, OSS Apache Falcon, Cloudera Navigator or the Zettaset Orchestrator. This helps achieve information provenance over the long haul

V. RELATED WORK

A considerable measure of distributions have been released in the recent past around Hadoop. However, there are very few to none around Big Data and Hadoop security. This is getting remediated with the book Hadoop in real life, Second Edition from Manning Publications [5] set to be published towards the end of 2015. This book will integrate security as a major aspect of the overall Hadoop ecosystem including greater depth and verbalization of the concepts presented in this paper.

VI. CONCLUSION

Hadoop and big information are never again trendy expressions in large enterprises. Whether for the correct reasons or not, enterprise information warehouses are moving to Hadoop and along with it come petabytes of information.

In this paper we have laid the groundwork for conducting future security assessments on the Big Data ecosystem and securing it. This is to ensure that Big Data in Hadoop does not become a big problem or a big target. Vendors pitch their technologies as the magical silver bullet. However, there are numerous challenges when it comes to deploying security controls in your Big Data environment.

This paper likewise provides the Big Data threat model which the reader can further expand and customize to their organizational environment. It moreover provides a target reference architecture around Big Data security and covers the entire control stack. Hadoop and big information represent a green field open door for security practitioners. It provides a chance to get ahead of the curve, test and deploy your devices, processes, patterns, and techniques before big information becomes a big problem.

REFERENCES

[1] EMC *Big Data 2020* Projects http:// www.emc.com /leadership/digitaluniverse/iview /big-data-2020.html

[2] NIST Special Publication 1500-1 NIST Big Data Interoperability Framework: Volume 1, Definitions http:// bigdatawg.nist.gov/_uploadfiles/M0392_v1 _3022325181.pdf

[3] *Securosis – Securing Big Data Security issues with Hadoop environments* https://securosis.com/blog /securing-big-datasecurity-issues-with-hadoop-environments

[4] *Top 10 Big Data Security and Privacy Challenges, Cloud Security Alliance,2012* https://downloads .cloudsecurityalliance.org/inititives/bdwg/ Big_Data_Top_Ten_v1.pdf

[5] *Hadoop in Action*, Second Edition by Manning Publications.ISBN:9781617291227http://www.manning. com/ lam2/

