INDIVIDUAL ASSIGNMENT 2

Big Data Framework and Solution

Student Name	:	AFIF RIFA'IE BIN MOHD GAZALI
Student ID	:	TP072295
Lecturer Name	:	ASSOC. PROF. DR. V. SIVAKUMAR
Module Name	:	BIG DATA ANALYTICS AND TECHNOLOGIES (032023-SIV)

Table of Contents

1. Introduction 1
2. Problem Statement – Addressing Challenges 1
    a. Data Sources Concern 2
    b. Data Storage Concern 2
    c. Data Serialization Concern 3
3. Adopting Big Data Solution – Justifications 4
4. Evaluation of Options 6
    a. Data Storage Layer 7
        i. The Hadoop Distributed File System (HDFS) 7
        ii. HBase 8
    b. Data Processing Layer 8
        i. Map Reduce 8
        ii. YARN 9
        iii. Apache Spark 9
    c. Data Querying Layer 10
        i. Apache Pig 10
        ii. Apache Hive 10
        iii. Apache SOLR 11
        iv. Apache Impala 11
    d. Data Ingestion Layer 12
        i. Apache Sqoop 12
        ii. Apache Flume 12
        iii. Chukwa 12
    e. Data Analytics 13
        i. Apache Mahout 13
        ii. R 13
    f. Management Layer 13
        i. Zookeeper 13
        ii. Oozie 14
    g. System Deployment Layer 14
        i. Ambari 14
    h. Data Cloud 14
        i. OpenStack 14

i. Data Security Layer 15
i. Apache Sentry 15
ii. Apache Knox 15
j. Data Streaming 16
i. Apache Storm 16
k. Visualization Layer 16
i. Tableau 16
ii. Power BI 16
5. Security, Social and Ethical Issue 17
a. User data privacy 17
b. The Creep Factor 17
c. Re-identification 18
6. Recommendations 18
7. Conclusion 21
8. References 22

1. Introduction

This individual assignment is designed to achieve two main learning outcomes in understanding the big-data analytics and technologies. First outcome is to have an evaluation of the framework of big-data eco-system for the given case study, and second, is to propose a big-data solution for the real-world scenario.

Based on the task description, the case study is centred on an experience of a cybersecurity tech company, SolidProtect Sdn. Bhd., that faced with various challenges in delivering the effective cybersecurity solutions for its clients, known as the Intrusion Prevention Systems (IPS). A good IPS requires a speedy identification of cybersecurity threats and inexpensive. Yet, the company faced with many challenges, in particular the enormous amount of data generated, can be as high as 10 TB per day, whereby in storing a such huge and growing amounts of data using traditional methods that is very costly and bottleneck in data storage due to the limited capacity. The current practices by SolidProtect Sdn. Bhd. in storing the data through a file-based system and extracting it by using a C++ program, and visualizing the data by Excel are not sustainable as they are time-consuming and costly.

2. Problem Statement – Addressing Challenges

With the above-mentioned background, the assignment is expected to propose a big-data solution for the company to address three data issues (storage, query, visualisation), speed up the time taken for network threat identification, and to have lower cost compared to the conventional approach. The proposed solution needs also to address the big-data eco-system, adoption of tools, data privacy and security, social and ethical issues, as well as to come up with appropriate assessment and suggestions.

An effective solution should be able to address challenges related to huge amount of network data, expensive conventional storage, and tedious data

extraction process due to heterogeneous format, so that the threat could be identified and rectified soonest possible. Therefore, there is an urgent need to find a better way to store, query, and visualise those huge network data in a more integrated form, that requires a big data solution that can speed up the threat identification with cheaper cost.

As indicated in the case study, currently the company are facing with several main challenges to serving better its clients, namely data source, storage, and data serialisation, as further elaborated in the following paragraphs.

2.a. Data Sources Concern

SolidProtect Sdn. Bhd.'s data sources include network devices such as firewalls, routers, switches, and other network equipment that generate a vast number of network activity data. The data generated by these devices can be complex and difficult to combine and analyse since they are in various log formats and from several sources. Furthermore, the volume of data collected might be overwhelming, necessitating substantial storage and processing capacity. Storing and managing such enormous amounts of data are costly, particularly if a traditional file-based system is utilised. Without effective data management strategies, the accuracy and integrity of the data may be jeopardised, resulting in decision-making errors, slow and inefficient network performance, and restricted capacity to support new applications and services.

2.b. Data Storage Concern

SolidProtect Sdn. Bhd. using file-based system to store the data. File-based storage systems have several limitations, including the cost of maintaining storage devices to cater a growing data size, data fragmentation, manual management and organization of data, and difficulty in handling

unstructured or semi-structured data. Compounding this problem is the file-based systems that are prone to data loss, and backups can take a lot of resources and time. These limitations make data processing less efficient, time-consuming, and prone to errors.

2.c. Data Serialization Concern

The SolidProtect Sdn Bhd process a log data that has several formats of data. Data serialization is the process of converting complex data structures, such as objects and arrays, into a format that can be easily transmitted over a network or stored in a file. However, this process can lead to various issues, including errors or lost data when data is serialized in one format and then deserialized in another format. Additionally, different serialization formats are available, each with its unique syntax, encoding, and data structures. If data is serialized in one format and then deserialized in another format, discrepancies in how the data is interpreted or processed may occur, leading to errors or lost data. Serializing and deserializing large volumes of data require significant processing power and memory, hence impacting system performance. Skilled professionals and advanced tools are also required to manage and analyse data, adding to the cost and complexity of the process.

To address the limitations of file-based systems and improve data processing efficiency, SolidProtect Sdn. Bhd. could consider implementing a big data analytics solution. A big data analytics solution can help integrate and analyse data from multiple sources, handle unstructured or semi-structured data, and provide real-time analysis of data to identify security threats promptly.

Those challenges in the current big-data eco-system can be summarised by the following diagram (Figure 1) as an illustration of the current framework of SolidProtect Sdn. Bhd. depicting problem areas and tools used in its current big-data eco-system.

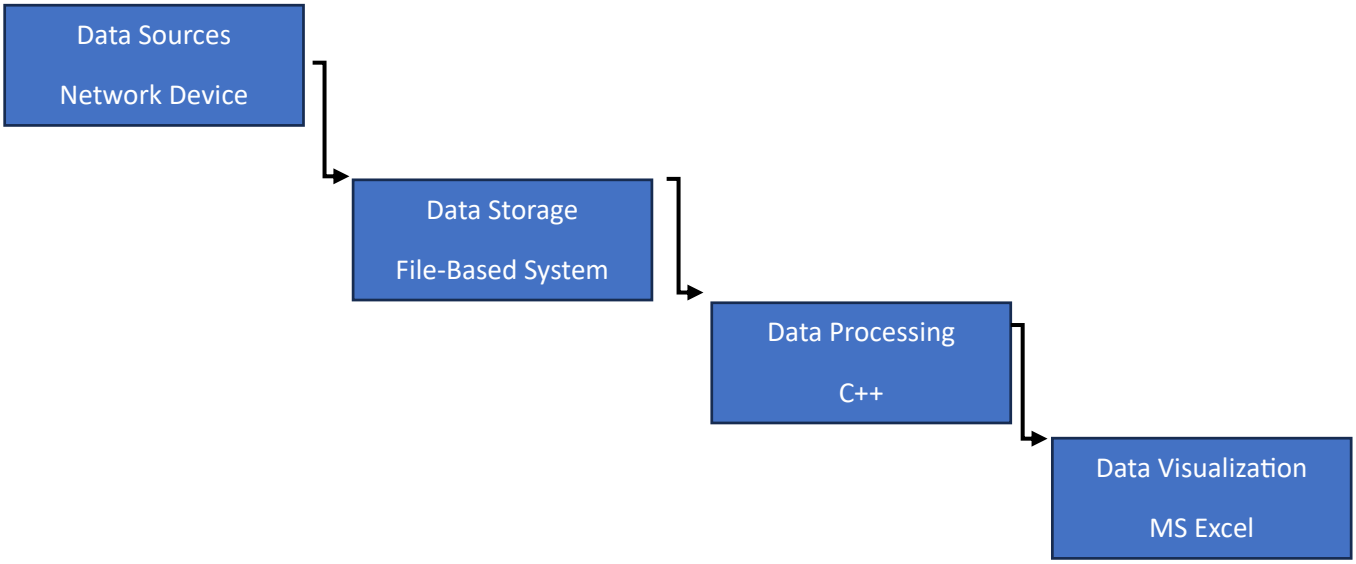

Figure 1: Current Framework of SolidProtect Sdn. Bhd.

3. Adopting Big Data Solution - Justifications

The limitations of file-based systems and the challenges associated with data serialization demonstrate the need for a more efficient and reliable data management approach for cybersecurity companies. Implementing a big data analytics solution could help SolidProtect Sdn. Bhd. to overcome the challenges, by improving data processing efficiency, and provide real-time analysis of data to strengthen their cybersecurity defences.

There is an urgency for the company to come up with new approaches and solutions to address the complexity of challenges posed by cybersecurity threat. It needs to find a better way to store, query and visualise the excessive network data in a more integrated platform. Fortunately, there are already available options to choose, which could be customised and mapped to the need or to address the identified challenges.

In this regard, Apache Hadoop is one of the examples that has emerged as a major player in the big data ecosystem. It is designed to process and

store large-scale data sets across distributed clusters in commodity hardware. Hadoop is a distributed computing framework that is cost-effective and scalable, suitable for managing structured, semi – structured and unstructured data. The framework can break down large files into smaller chunks and distribute them across multiple servers for faster processing using simple programming.

Hadoop is composed of two main components: Hadoop Distributed File System (HDFS), MapReduce (Zeebaree et al., 2020) providing a highly scalable and robust distributed storage system, programming model for parallel processing, resource management, and necessary tools and libraries for other components to function.

Hadoop's scalability, fault tolerance, and parallel processing capabilities make it an essential component in managing big data. Additionally, Hadoop features a deduplication capability that efficiently removes duplicate information from large datasets, enhancing its efficiency in data processing. Its ecosystem featured in Figure 2.

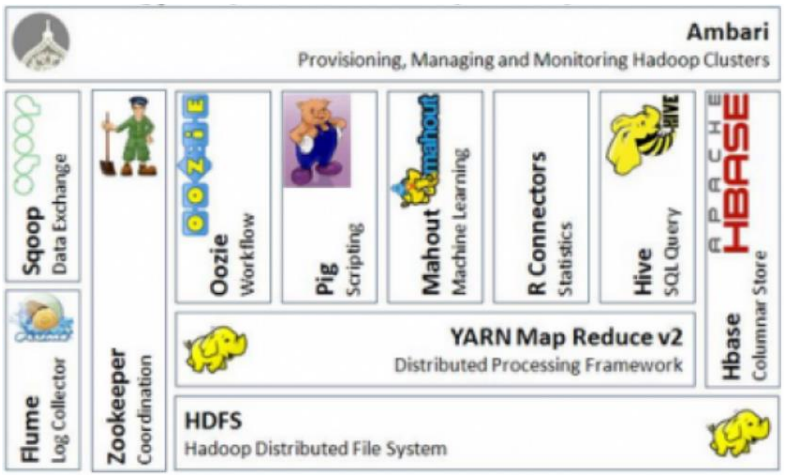

Figure 2: Hadoop Ecosystem

4. Evaluation of Options

Aside from the Hadoop ecosystem, alternative tools have been introduced to provide more options to users, which will be addressed in this section of this assignment. This section provides an overview of the main big data technologies and provides comparisons across different system layers, including Data Storage Layer, Data Processing Layer, Data Querying Layer, Data Ingestion Layer, and Data Analytics Layer, Management Layer, and System Deployment Layer (Oussous et al., 2017). Their respective features, advantages, and limitations will be categorized and discussed in this section. Additionally, this paper also includes Data Cloud Layer, Data Security Layer, Data Streaming Layer, and Visualization Layer that make a robust big data ecosystem of cybersecurity, as depicted in Figure 3.

BIG DATA ECOSYSTEM LAYERS	**TOOLS**
Visualization	Tableau & Power BI
Data steaming	Apache Storm & Spark
Data Security	Apache Sentry & Apache Knox
Data Cloud	OpenStack
System deployment	Apache Ambari
Management Layer	Zookeeper, Oozie
Data Analytics	Apache Mahout & R
Data Ingestion	Apache Sqoop, Apache Flume, & Apache Chukwa
Data Querying	Apache Pig, Apache Hive, Apache SOLR, & Apache Impala

Data Processing	Map Reduce, Yarn & Apache Spark,
Data Storage	HDFS & Hbase
Ecosystem	Apache Hadoop

Figure 3: Ecosystem of Hadoop and Alternatives Tools

4.a. Data Storage Layer

i. The Hadoop Distributed File System (HDFS)

HDFS (White, 2012) is a data storage system that is designed to support many nodes in a cluster with cost-effective and highly reliable storage capabilities. It could accommodate both structured and unstructured data in exponential volumes. The main features of HDFS are support heterogeneous hardware and software platform. In addition, HDFS mitigates network congestion and enhances performance by facilitating the storage of computation processes near data storage. The system exhibits fault tolerance by replicating data in the event of a fault. HDFS consist of Master-Slave architecture and distribute large data across the cluster. The cluster is comprised of a singular master, known as the Name Node, which is responsible for overseeing system operations. while many slave (Data Node) coordinate storage and individual node.

***Limitation*.** HDFS do not support general purpose system as it is designed with high latency in batch processing and do not provide fast lookup in files.

ii. HBase.

HBase (Prasad & Agarwal, 2016) is a free and open-source database. It is a distributed non-relational database with minimal latency developed on top of HDFS. Its column-oriented key/value structure allows for flexible structured hosting of large tables. HBase stores data in a logical pattern of rows and columns. This feature enables HBase to handle an exponential number of columns and rows by grouping the attributes into column families and storing them together. HBase is more adaptable and can handle changing application requirements better. It also supports real-time queries, linear and modular scalability, and consistent access to large data sets. By default, Zookeeper coordinates HBase. HBase, like HDFS, has a Master Node that administers the cluster and slaves that store sections of tables and conduct data operations.

Limitation. HBase does not support structured query languages such as SQL and has limitations in this regard.

4.b. Data Processing Layer

i. MapReduce

MapReduce (Lydia & Swarup, 2015) is a programming model framework that simplifies data quantities in an efficient and cost-effective manner. It allows for parallel processing using two functions. Reduce function and map function.

MapReduce features a Name Node that manages Job Tracker, which is used to schedule different jobs and assign tasks to slave nodes. To ensure reliability, the Job tracker monitors the slave nodes and reassigns the task if there is any malfunction. Hadoop clusters are made up of client servers that load data and send MapReduce tasks to Name nodes. It also includes Name Nodes (primary and secondary), with the primary master name node

coordinating and managing storage computation. While the secondary Master Node is in charge of data replication and availability.

ii. YARN

On top of HDFS, YARN provides more scalable, parallelizable, and resource-advanced operating system features than MapReduce. As a result, YARN can handle both batch processing and interactive real-time processing. To improve efficiency, YARN divided the Job Tracker into two daemons (White, 2012).

1. Resources Manager (RM) manages cluster resources.

2. Application Master (AM) is a task manager that allows you to schedule tasks, assign them to Task Tracker, and track their progress. This is done to ensure that task bookkeeping and job scheduling programmes run in a cluster.

iii. Apache Spark

Spark (Acharjya, 2016) is an open-source distributed processing network that, like Hadoop, however, it improves performance by utilising in memory system processing. With quick, easy-to-use, and flexible computation, Spark can handle very sophisticated analysis on big data sets. It is widely acknowledged that it is more efficient in data access than Hadoop. Spark may run independently or on top of Hadoop YARN, reading data straight from HDFS. Spark Streaming offers automated parallelization, a highly scalable streaming method, and fault tolerance. It is possible to combine batch jobs with interactive queries, as well as conduct streaming computations in a succession of batch jobs in memory, using data from the Resilient Distributed Dataset (RDD).

4.c. Data Querying Layer

i. Apache Pig

Apache Pig (Mazumder, 2016) is an open-source framework featuring a high-level scripting language known as Pig Latin. The primary function is to reduce the complexity of MapReduce by facilitating concurrent execution of MapReduce Jobs and workflow on Hadoop and processing massive amounts of data in parallel using HDFS. Pig Latin offers the advantage of having an intuitive syntax for simultaneous developing of MapReduce tasks in simple or nested flows. Pig can handle various data format and does not require schema like SQL. Pig supports a single local environment as well as a distributed environment on a Hadoop cluster, and it supports more data types than Hive.

ii. Apache Hive

Hive (Shaw et al., 2016) is a data warehousing tool designed to simplify the utilisation of Hadoop. It provides support for data within HDFS files. One advantage of utilising Hive is its ability to present data in a structured database format, which is more user-friendly compared to Hadoop that is primarily based on the JAVA programming language. The approach is predicated on the utilisation of tables, whereby the HDFS libraries are partitioned and subsequently subdivided into discrete buckets. The Hive platform uses a schema on read and provides support for multiple schemas.

Limitation. The Hive platform was designed for low latency, catering to the demands of large-scale processing. However, it may exhibit a longer processing time when handling smaller jobs. Row-level inserts, updates, and deletions are not available, and the queries are comparatively slow and do not support SQL.

iii. Apache SOLR

SOLR is a JAVA-based open-source search platform that can function as a standalone full-text search or utilise HDFS as a foundation (*Apache Solr Tutorial*, n.d.) .Its support system can manage and processing diverse data sets from multiple sources, including HDFS. The system employs the HDFS for the purpose of indexing and managing long-term storage of files. SOLR offers a range of advantages, including the provision of data in near real-time, support for full-text search, real-time indexing, integration with databases, NoSQL capabilities, and the ability to handle complex and diverse documents. It has been developed with a focus on scalability, fault tolerance, and an extensible plugin architecture. With Zookeeper, it has the capability to adjust its scale in either direction by utilising automated processes such as index replication, distribution, load-balancing, and failover and recovery mechanisms.

iv. Apache Impala

Impala is a parallel processing SQL query engine that is open source and designed to be compatible with Hadoop for storing data (Sakr, 2016) . The technology facilitates interactive and real-time analysis by employing its own memory processing engine, which enables swift queries over extensive data sets. This stands in contrast to Hive, which relies on the MapReduce framework. Make returns query results much more quickly than Hive. The software possesses a wide range of plugin capabilities that enable the direct utilisation of data from HDFS and HBase sources. The objective is to reduce the amount of data movement and decrease the duration of execution. Additionally, it conveys fault tolerance.

4.d Data Ingestion

i. Apache Sqoop

Scoop is an openly available software that facilitates the transfer of large amounts of data between Hadoop and structured data storage, such as relational databases, through a command-line interface (Vohra, 2016) . It offers fault tolerance, quick response times, and optimal system utilisation to lessen the processing demands placed on external systems. The importation of data is facilitated by MapReduce and other high level programming languages such as Pig, Hive, SOLR, and Impala. The software facilitates seamless incorporation with HBase, Hive, and Oozie.

ii. Apache Flume

Flume facilitates the transfer of data in between Hadoop Distributed File System (HDFS) through the process of collecting, aggregating, and transferring data from external machines (Hoffman, 2013) . Its flexible architecture, extendable data model, and ability to handle streaming data flows allow it to handle enormous distributed data sources. Flume exhibits a high level of integration, effectively interfaces with other platforms, and maintains robust connectivity with Hadoop. It has fault tolerance, failure recovery service. Flume can stream data from various data sources into HDFS and HBase for real time analysis.

iii. Chukwa

Chukwa (Oussous et al., 2017) is a data collection system that operates on the Hadoop platform. It keeps track of a distributed dataset, uses HDFS to gather information from several databases, and assigns the information to MapReduce for processing. It provides customizable data ingestion by structuring a pipeline of data collection, processing, and transitional stages.

4.e. Data Analytics

i. Apache Mahout

Mahout is an open-source learning software library that runs algorithms via MapReduce or another platform on top of Hadoop. Mahout has the advantage of being scalable and efficient in the implementation of large-scale machine learning and algorithms across huge datasets (Dinsmore, 2016) . Clustering, classification, collaborative filtering, frequent pattern mining, and text mining are all available. It provides a SQL-like interface for querying data in the Hadoop distributed file system.

ii. R

R is an open-source statistical computing, machine learning, and graphics programming language (Ihaka & Gentleman, 1996). Its input and output functionalities are well developed, simple, and effective. It includes tools such as ff, huge memory, snow package, and Teradata Aster R for discovery platform. In comparison to Mahout, R includes a set of classification models.

Limitation. R's capacity to handle extremely big datasets is limited due to one node memory, which causes memory overload.

4.f. Management Layer

i. Zookeeper

Zookeeper is an open-source tool for managing Hadoop applications and clusters. It is a client-server architecture-based distributed application that supports high speed and data availability in distributed programming (Lublinsky et al., 2013). It is built on in-memory data management, allowing for high-speed distributed data coordination. It also offers configuration management, replicated synchronisation to secure data and

nodes, and automatic system recovery from faults. It can also be used on platforms other than Hadoop.

ii. Oozie

Oozie is a workflow scheduler solution for running and managing Hadoop tasks. It is scalable, dependable, and extendable, and it can handle the efficient execution of enormous volumes of workflows (Islam & Srinivasan, 2015). There are two major components: a workflow engine that stores and executes any form of workflow job and a coordinator engine that executes a predetermined task schedule. The advantage of Oozie is that it allows both workflow and execution customisation. Oozie supports numerous database types, including MySQL and PostgreSQL.

4.g. System Deployment Layer

i. Ambari

Ambari (Wadkar & Siddalingaiah, 2014) simplifies Hadoop management by deploying, administering, and monitoring clusters via an easy Web User Interface. Ambari supports various Hadoop components, including HDFS, MapReduce, Hive, and others. It also adds security by allowing for role-based user authentication and authorization.

4.h. Data Cloud

i. OpenStack

The cloud operating system platform OpenStack is free and open source. It is made up of interconnected parts that work together to govern sizable computation, data storage, and networking pools through datacentres (*What Is OpenStack?*, n.d.). These pools are managed and provisioned using APIs using standard authentication techniques. It's built on broken-

up service-allow-plugin-components. Physical pools are used as the source for aggregating physical resources. OpenStack enables users to build their own cloud infrastructure, thereby mitigating the risk of exposing confidential data and proprietary information.

4.i Data Security Layer

i. Apache Sentry

Sentry is an open-source Hadoop authorization module with granular, role-based authorisation (Das, 2018). It allows authenticated users and applications on a Hadoop cluster to control data security levels. It works with Hive, SOLR, IMPALA, and HDFS. It's intended to function as a pluggable authentication engine for Hadoop components. Sentry is a modular system that can enable authorisation for a wide range of Hadoop data formats and construct authorization rules to check a user or application's access request for Hadoop resources.

ii. Apache Knox

Apache Knox is an open-source project that provides perimeter protection, allowing it to reliably extend Hadoop access to new users while adhering to enterprise security requirements (Cloudera et al., n.d.). Knox streamlines Hadoop security for users who access cluster data and run processes. It works with popular identity management and SSO solutions to safeguard Hadoop clusters. It gives a solitary point of access to the Hadoop access point.

4.j. Data Streaming

i. Apache Storm

Strom (Mazumder, 2016) is an open-source distributed system that is capable of processing real-time data, in contrast to Hadoop, which is designed for batch processing. It's has Stream, spouts, and bolts make up topology-based structures. A spout is utilised as a primary origin of a stream, while a bolt is employed to transform input streams into output streams. The Storm is distinguished by its user-friendly interface, scalability, and fault tolerance capabilities. It can be utilised for continuous computation, online machine learning, and real-time analytics. The storm is processed and analysed using Hadoop tools and to simplify programming model.

4.k. Visualization Layer

i. Tableau

Tableau is a big data visualisation application that allows to create charts, graphs, maps, and other visualisations (Caldarola & Rinaldi, 2017). It provides a server solution that allows for online visualisation of reports in a cloud service. It allows to connect to numerous data sources such as file formats, relational and non-relational data systems, and cloud systems. It features the distinct feature of data blending and the ability to cooperate in real time (Amer & El-hadi, 2019).

ii. Power BI

Power BI is a business intelligence-focused interactive data visualisation programme. Its a suite of software services, apps, and connectors for transforming disparate data sources into visually immersive, interactive insights (Amer & El-hadi, 2019). It accepts data from numerous sources

such as database, webpage, or structured file like excel. It offers a simple web-based interface with customisable visualisation and minimal control over data sources. It is linked to various business analytics tools and is well integrated with Microsoft products such as MS Excel, Azure Could Service, and SQL Server.

5. Security, Social and Ethical Issue.

a. User data privacy

Processing of personal data and information carries an inherent risk to individual rights. the potential for data loss, destruction, unauthorised alterations, and unauthorised disclosure to third parties. This could happen as a result of the large-scale processing of big data, which increases the danger of information and data leak because of human mistake or security flaws. Beside setting notifications and reducing the impact of the data breach, organisations must put up security measures and policies to prevent such an occurrence from occurring.

b. The Creep Factor

The term "creep factor" describes the uneasiness and discomfort that an entity feels when they realise that their personal information is being collected, watched, and used without their knowledge or permission. Unauthorised use of privacy, security, and the potential for data misuse for advertising, spying, or other objectives cause uneasy feelings. It is advised for the corporation to emphasise openness, adhere to ethical policies, and enforce regulations to protect clients' private rights as big data technology continues to advance at a rapid pace.

c. Re-identification

Re-identification after anonymization refers to the ability to locate an individual and connect them to previously anonymised data. Initially, anonymization is a technique for preserving privacy by removing sensitive data from a dataset. However, it is now possible to undo the anonymization process and link the data back to specific people due to the availability of new information.

Re-identification emphasises crucial to keep historical and present data secure and protected. Making it more difficult for someone to trace back to the unique information owner of the data is one way to reduce the risk. Other methods include introducing noise or generalising the data. To prevent unauthorised attempts to re-identify, stringent access controls and procedures should be prioritised and set a limit on who can access the data.

6. Recommendations.

The recommendation explored the choices that the Hadoop Ecosystem and its associated platform have to offer. Hadoop was selected due to their ability to collect, mitigate, transmit, store, and retrieve huge data linked to security in an inexpensive manner. The Hadoop ecosystem was chosen due to its open-source platform and services, support from the local experience community, and accessibility to public domain information, enhanced with cybersecurity model.

(Lněnička et al., 2017) The cybersecurity model is made up of three key parts. 1. Sources of Data. 2. gathering of data. 3. Transportation. For enhanced safety protection, a distributed security layer for authentication and permission is used. The architecture should be idealised such that data can be kept in a separate repository, connected, and analysed alongside external and historical data.

As a data collection and transit component, Apache Flume was chosen to provide a dependable, efficient service for gathering, aggregating, and moving massive amounts of log data.

Apache Sqoop was chosen due to its efficiency in transferring large amounts of data between structured datastores such as relational databases.

Apache Pig is used as a tool for analysing huge datasets that comprises of a high-level language for writing data analysis programmes and infrastructure for programme evaluations.

A distributed infrastructure with cloud computing proposes cost-effective storage. A cloud architecture, as a distributed system, allows for vast amounts of data storage and high computations to be performed by many computers in a data centre. To leverage the real-time monitored stream of data that can be fed and kept in the storage server while waiting for threat analysis and detection, the OpenStack solution is proposed.

For cyber threat information monitoring, gathering, and transmission to the cloud for security analysis, Apache Hadoop and Spark were chosen. Hadoop and Spark as data processing to improve data storage efficiency, speed up data access, improve threat monitoring and detection efficiency, and minimise operational delays. Hadoop is simply utilised for storage purpose, while Spark employs its own cluster management for computation.

Apache Hadoop and Apache Storm were utilised as a free and open source distributed real-time computing system to process data streams in real time and Hadoop for batch processing. The Apache Hadoop File System HDFS is used to store raw data. Apache HBase is utilised as a result data storage system, allowing the findings to be conveniently retrieved and used for detection and visualisation. Apache SOLR searches Hadoop data stored in HDFS. When these components are combined, large amounts of security data can now be stored in a specialised storage system that accept both internal or external data to aggregate and analyse real-time and historical data.

Apache Sentry and Apache Knox were introduced to address privacy and security concerns. Sentry allows authenticated users and applications on a Hadoop cluster to regulate and enforce precise levels of data privileges. Sentry now supports Apache Hive, Apache SOLR, Impala, and HDFS (with the exception of Hive table data). The Apache Knox Gateway is a system that serves as a centralised authentication and access point for Hadoop components. The Apache Sentry authorisation engine is then designed to be pluggable. It enables the creation of authorization rules to validate a user's or application's access requests to Hadoop resources. Sentry is very modular and can handle authorisation for a wide range of Hadoop data formats.

Zookeeper controls and coordinates the Hadoop application and cluster, whilst Oozie acts as a scheduler to run and manage jobs in clusters. Ambari is used for system deployment to simplify Hadoop management by allowing deploying, managing, and monitoring Hadoop clients via a Web User Interface. Tableau was chosen because of its ease of use and ability to perform advanced analytics at a lower cost than other visualisation tools in terms of setup, maintenance, and cost of ownership.

With the aforementioned big data ecosystem, SolidProtect Sdn Bhd is anticipated to be able to resolve three data-related issues (storage, query, and visualisation). This method employs high level query languages and real-time streaming processing to identify network threats more quickly and at a lesser cost because most of the tools are open source. Additionally, to develop Big Data Solutions for SolidProtect Sdn. Bhd. in the cybersecurity industry, the suggested ecosystem also looks into cloud infrastructure and security layer for robust big data solution. Figure 4 is the snipped of Hadoop ecosystem.

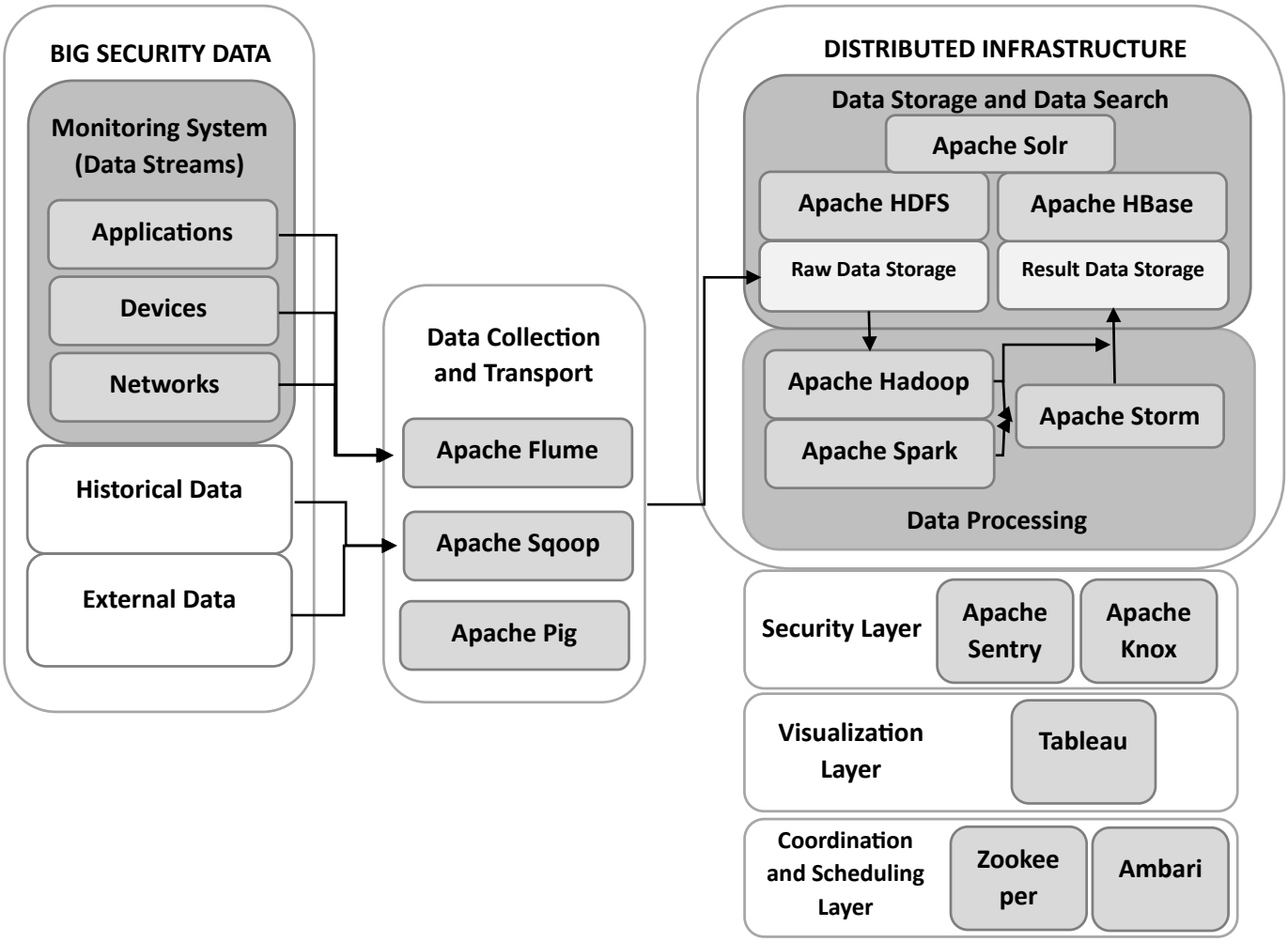

Figure 4: Cybersecurity Model Using Distributed Infrastructure With The Hadoop Ecosystem

7. Conclusion

This assignment examined the SolidProtect Sdn. Bhd's big data ecosystem framework, which has various problems in areas such as storage, query, and visualisation. Discussions done on the need for a big data solution to help SolidProtect Sdn. Bhd. solve its problems. In this context, big data tools were presented for replacing components in existing eco-systems with the Apache Hadoop framework eco-system. Investigation was made on the Hadoop ecosystem and tools at various layers, including the Data Storage Layer, Data Processing Layer, Data Querying Layer, Data Ingestion Layer,

and Data Analytics Layer, as well as the Data Cloud Layer, Data Security Layer, Data Streaming Layer, and Visualisation Layer. Critical assessments were made on the potential security issues, social, and ethical challenges, as well as potential solutions. Finally, this assignment assessed and recommended practical solutions and tools for SolidProtect Sdn. Bhd. addressing its identified challenges.

8. References

Acharjya, D. P. (2016). A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools. *International Journal of Advanced Computer Science and Applications*, *7*(2).

Amer, A., & El-hadi, M. (2019). *Tableau Big Data Visualization Tool in the Higher Education Institutions for Sustainable Development Goals*. https://www.semanticscholar.org/paper/Tableau-Big-Data-Visualization-Tool-in-the-Higher-Amer-El-hadi/f9df84a2d0f5a3c012b09b328b29443db81eed18

Apache Solr Tutorial: What Is, How It Works & What Is It Used For. (n.d.). Sematext. Retrieved May 19, 2023, from https://sematext.com/guides/solr/

Caldarola, E. G., & Rinaldi, A. M. (2017). Big Data Visualization Tools: A Survey - The New Paradigms, Methodologies and Tools for Large Data Sets Visualization: *Proceedings of the 6th International Conference on Data Science, Technology and Applications*, 296–305. https://doi.org/10.5220/0006484102960305

Cloudera, © 2023, Terms, I. A. rights reserved, Statement, C. | P., Hadoop, D. P. | U. / D. N. S. M. P. I. A., trademarks, associated open source project names are trademarks of the A. S. F. F. a complete list of, & Here, C. (n.d.). *Apache Knox*. Cloudera. Retrieved May 19, 2023, from https://www.cloudera.com/products/open-source/apache-hadoop/apache-knox.html

Das, D. (2018, February 6). Apache Sentry. *ThirdEye Data*. https://thirdeyedata.ai/apache-sentry/

Dinsmore, T. W. (2016). Streaming Analytics. In T. W. Dinsmore (Ed.), *Disruptive Analytics: Charting Your Strategy for Next-Generation Business Analytics* (pp. 117–144). Apress. https://doi.org/10.1007/978-1-4842-1311-7_6

Hoffman, S. (2013). *Apache Flume: Distributed Log Collection for Hadoop*. Packt Publishing Ltd.

Ihaka, R., & Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, *5*(3), 299–314. https://doi.org/10.1080/10618600.1996.10474713

Islam, M. K., & Srinivasan, A. (2015). *Apache Oozie: The Workflow Scheduler for Hadoop*. O'Reilly Media, Inc.

Lněnička, M., Capek, J., Komárková, J., Máchová, R., & Čermáková, I. (2017, November 8). *A Solution to Combat Cybersecurity Threats Involving Big Data Analytics in the Hadoop Ecosystem*.

Lublinsky, B., Smith, K. T., & Yakubovich, A. (2013). *Professional Hadoop Solutions*. John Wiley & Sons.

Lydia, E. L., & Swarup, D. M. B. (2015). *Big Data Analysis using Hadoop components like Flume, MapReduce, Pig and Hive*. *5*(11).

Mazumder, S. (2016). Big Data Tools and Platforms. In S. Yu & S. Guo (Eds.), *Big Data Concepts, Theories, and Applications* (pp. 29–128). Springer International Publishing. https://doi.org/10.1007/978-3-319-27763-9_2

Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., & Belfkih, S. (2017). Big Data Technologies: A Survey. *Journal of King Saud University - Computer and Information Sciences*, *30*. https://doi.org/10.1016/j.jksuci.2017.06.001

Prasad, B. R., & Agarwal, S. (2016). Comparative Study of Big Data Computing and Storage Tools: A Review. *International Journal of Database Theory and Application*, *9*(1), 45–66. https://doi.org/10.14257/ijdta.2016.9.1.05

Sakr, S. (2016). Introduction. In S. Sakr (Ed.), *Big Data 2.0 Processing Systems: A Survey* (pp. 1–13). Springer International Publishing. https://doi.org/10.1007/978-3-319-38776-5_1

Shaw, S., Vermeulen, A. F., Gupta, A., & Kjerrumgaard, D. (2016). Hive Architecture. In S. Shaw, A. F. Vermeulen, A. Gupta, & D. Kjerrumgaard (Eds.), *Practical Hive: A Guide to Hadoop's Data Warehouse System* (pp. 37–48). Apress. https://doi.org/10.1007/978-1-4842-0271-5_3

Vohra, D. (2016). Using Apache Sqoop. In D. Vohra (Ed.), *Pro Docker* (pp. 151–183). Apress. https://doi.org/10.1007/978-1-4842-1830-3_11

Wadkar, S., & Siddalingaiah, M. (2014). Apache Ambari. In S. Wadkar & M. Siddalingaiah (Eds.), *Pro Apache Hadoop* (pp. 399–401). Apress. https://doi.org/10.1007/978-1-4302-4864-4_20

What Is OpenStack? Benefits and Components. (n.d.). Fortinet. Retrieved May 19, 2023, from https://www.fortinet.com/resources/cyberglossary/openstack

White, T. (2012). *Hadoop: The Definitive Guide*. O'Reilly Media, Inc.

Zeebaree, S., Shukur, H., Haji, L., Zebari, R., Jacksi, K., & Abass, S. (2020). Characteristics and Analysis of Hadoop Distributed Systems. *Technology Reports of Kansai University*, *62*, 1555–1564.