Scientific Research Publishing

Journal of Information Security, 2022, 13, 23-42
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

Hadoop Distributed File System Security Challenges and Examination of Unauthorized Access Issue

Wahid Rajeh

Department of Information Technology, University of Tabuk, Tabuk, Saudi Arabia
Email: Wahid.ra@ut.edu.sa

How to cite this paper: Rajeh, W. (2022) Hadoop Distributed File System Security Challenges and Examination of Unauthorized Access Issue. *Journal of Information Security*, **13**, 23-42.
https://doi.org/10.4236/jis.2022.132002

Received: January 14, 2022
Accepted: February 13, 2022
Published: February 16, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

Hadoop technology is followed by some security issues. At its beginnings, developers paid attention to the development of basic functionalities mostly, and proposal of security components was not of prime interest. Because of that, the technology remained vulnerable to malicious activities of unauthorized users whose purpose is to endanger system functionalities or to compromise private user data. Researchers and developers are continuously trying to solve these issues by upgrading Hadoop's security mechanisms and preventing undesirable malicious activities. In this paper, the most common HDFS security problems and a review of unauthorized access issues are presented. First, Hadoop mechanism and its main components are described as the introduction part of the leading research problem. Then, HDFS architecture is given, and all including components and functionalities are introduced. Further, all possible types of users are listed with an accent on unauthorized users, which are of great importance for the paper. One part of the research is dedicated to the consideration of Hadoop security levels, environment and user assessments. The review also includes an explanation of Log Monitoring and Audit features, and detail consideration of authorization and authentication issues. Possible consequences of unauthorized access to a system are covered, and a few recommendations for solving problems of unauthorized access are offered. Honeypot nodes, security mechanisms for collecting valuable information about malicious parties, are presented in the last part of the paper. Finally, the idea for developing a new type of Intrusion Detector, which will be based on using an artificial neural network, is presented. The detector will be an integral part of a new kind of virtual honeypot mechanism and represents the initial base for future scientific work of authors.

Keywords

Hadoop Security Issue, Unauthorized Access, Honeypot Node, Intrusion

Detector

1. Introduction

The Big Data concept is based on storing, processing and transferring of vast amounts of unstructured, semi-structured and structured data [1]. It could be a collection of large data sets with petabytes of raw data (user and enterprise data, sensor information, medical and transaction data). Generally, it is a real challenge to store and adequately processes enormous data quantities by using traditional processing tools. Because of that, Big Data technology is gaining global importance that will have exponential growth in the future [2]. This technology provides new opportunities for all industry sectors, companies, and institutions that depend on quality processing of large amounts of raw data. It can be described by three main properties ("3V" properties): volume, velocity, and variety [3]. The volume represents the quantity of data that could be transferred from a source of information to a system of interest. The variety feature can be determined by existing data types within a data set, while velocity represents the speed of storing and processing the data. Besides these three essential characteristics, Big Data can be described with variability (inconsistency with periodic peaks during the flow of data) and complexity (various types of data that come from multiple sources) [4]. However, along with all the benefits of using Big Data technology, many challenges and potential issues occur [5] [6]. The challenges are the consequence of Big Data complexity and difficulties with performing data operations like storing, sharing, searching, analyzing and transferring large amounts of information. On the other side, one of the leading Big Data issues is a potential system vulnerability by malicious parties. Large quantities of valuable and private information can be easily exposed to malicious clients who want to steal or use data without required permissions. That way, the privacy and integrity of data can be strongly jeopardized. With the purpose to improve effectiveness and increase the robustness of existing Big Data systems, the Hadoop mechanism is proposed.

1.1. The New Era of Distributed File System

The Hadoop is a master-slave open source platform for storing, managing and distributing data across a more significant number of servers [7]. It is a Java-based solution to the majority of Big Data issues that is distributed under the Apache License. It is a highly accessible technology that operates with the large volumes of data and can be used for high-speed distribution and processing of information. Hadoop efficiently resolves the "3V" challenge by providing next features to a system: a framework for horizontal scaling of large data sets, for the handling of furious transfer velocity rates and efficient framework for processing a variety of unstructured data. Also, it can handle the failure of a single machine

by re-executing all of its tasks. However, in the large-scale system as Hadoop, the occurrence of failures is unavoidable.

On a basic level, Hadoop is built from two main components [8]: MapReduce and Hadoop Distributed File System (HDFS). MapReduce component is used for the computational implementation of Hadoop in the form of distributed processing of data. It organizes multiple processors in a cluster to perform required calculations. MapReduce distributes the computation assignments between computers and puts together final computation results in one place. Additionally, this component takes care of network failures in the way they do not disturb or disable active computation processes. On the other side, HDFS is used for information management and distributed storage of data. It is the file system component that provides reliable and scalable storage features and global file access option. HDFS component is of the main interest in this paper so that it will be additionally explained in the next two subsections.

1.2. HDFS Architecture

The main goals of HDFS are storing large amounts of data in clusters and providing a high throughput of information within a system. Data stores in the form of the same sized blocks, where the typical size of each block is 64 MB or 128 MB. Depending on the size, each file is stored in one or a few blocks. Size of a block is configurable, and each file can have one writer at the moment. Within the HDFS component, a client can create new directories, create, save or delete files, rename and change a path of a file, and etc.

HDFS architecture is based on the master-slave principle, and it is built from a single NameNode and group of DataNodes [9]. The NameNode (the master node), as the core part of the system, manages the HDFS directory tree and stores all metadata of the file system. Clients communicate directly with the NameNode with the purpose to perform standard file operations. Further, the NameNode performs a mapping between files stored at DataNodes with proper file names. Another function is monitoring the possible failure of a DataNode and resolving this issue by creating a block replica [10]. The NameNode can have two other roles in the system: it can act as a CheckpointNode and a BackupNode. A periodical checkpoint is an excellent way to protect the system metadata. On the other hand, BackupNode maintains file image that is synchronized with the NameNode state. It handles potential failures and rolls back or restarts using the last good checkpoint. Additionally, in enterprise versions of Hadoop, there is a practice to introduce Secondary NameNode. It is a useful system addition in a case the original NameNode crashes. In that case, Secondary NameNode uses saved HDFS checkpoint and restarts crashed NameNode. DataNodes are proposed to store all file blocks and to perform the tasks that are delegated by the NameNode. Each file of a DataNode can be split into a few blocks and labelled with an identification timestamp. These nodes are used to provide service of writing and reading of desired files. By default, each data block is replicated

three times, of which, two copies are stored in two different DataNodes in a single rack, and a third copy is saved on a DataNode which belongs to another rack.

2. Key Security Challenges

Many security challenges characterize Hadoop technology [11] [12]. That comes from the fact that it operates by using a variety of different technologies such as databases, operating systems, networks, communication protocols, memory resources, processors, etc. The occurrence of a security problem in one of the mentioned components can endanger the work of the entire system. That's why it is necessary to seriously consider the security challenges and operational issues of Hadoop from all perspectives. It is a complex system that could be significantly affected by network security problems, authentication and authorization issues, data availability and integrity and by additional security requirements. Parallel computation capability of Hadoop results with a complex environment that is at high risk of attacks. Users share physical resources between each other, and therefore, a user does not have complete control over data. It is a consequence of parallelism which establishes the data storage across many machines. In that case, a client and a malicious party can easily share the same physical devices. If an adequate security system is not implemented, a malicious party can get full access to data and compromise honest clients. Compromised clients propagate malicious data through a network by what a whole system can be affected. However, three challenges are discussed in the following section.

2.1. Utilizing Remote Procedure Call Protocol

The Hadoop technology is based on Remote Procedure Call over TCP/IP for the transfer of data between different nodes [13]. Default communication is not secure, and malicious parties can easily modify internode communication for hacking the system. Computations can be performed anywhere within clusters, so it is a complex task to find the precise location of a single computation. Because of that, it is also challenging to ensure the security of each computation location. Apart from that, insecure communication can influence data leakage during the transfer of data between a DataNode and a client. Also, it is not a rare case that undesirable nodes are added to a system with a task to steal data or compromise computations.

2.2. Replication Storage Model

Hadoop technology is based on storing large amounts of information in multiple clusters in a distributed way. Every cluster could be built from thousands of DataNodes, which makes the entire system structure complex. A simple example of HDFS storage working principles is presented in **Figure 1**. A large file is first split into three data blocks. Each of them is replicated three times and saved into different DataNotes from security reasons. Distributed File System (DFS) organizes

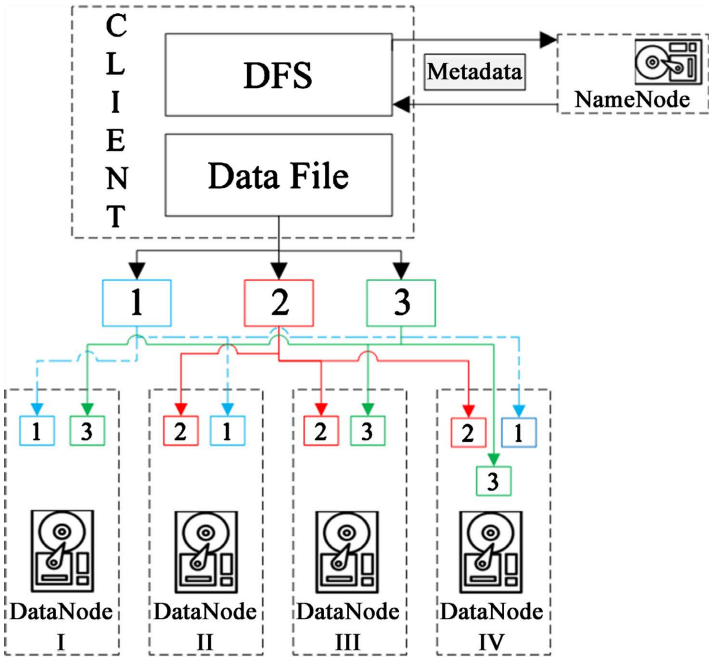

Figure 1. Replication storage model.

the transfer of metadata between the NameNode and a client.

This operational algorithm is designed for all honest users within a system. However, a malicious party can access HDFS as any user, which can result in a variety of severe security issues. From that reason, implementation of access control algorithm and denial of access of unauthorized clients are of the most prominent importance for reliable and secure distributed systems. An entry point for a malicious client can be every DataNode itself. When unauthorized party access to a DataNode, private user data is easily accessible. This severe and unacceptable risk for all potential Hadoop users who want to use this framework. Additionally, a malicious intruder can run malicious executive codes to Hadoop services and interrupt the operational mode of HDFS, NameNode, DataNodes, and all other network components.

Generally speaking, there are four types of Hadoop users. The first one is a regular user who exploits Hadoop capacities to process, transfer, and store data. The second type of user is a business user, for which one the Hadoop technology is as an ideal solution for improving existing business solutions. Scientific researchers and developers are the third types of users. The final, fourth group, are malicious users who try to steal or misuse the data by accessing it on the unauthorized way. These users cannot be stopped from attacking the system in advance. But learning about the potential threats and possibilities for unauthorized attack can result in improving security performances of Hadoop. The platform is continuously evolving, getting

2.3. Cluster Security Levels

Default Hadoop setup includes no security feature within clusters. On Level 0 of

the security, the system assumes that including parties are honest ones and that a trust level exists. However, authorization features may exist inside Level 0 in which case they are assigned to objects. An attacker could quickly overcome this security level by performing a bypass authentication attack. Level 1 is the first stronger defense line from unauthorized attacks. It includes EdgeNode which limits and restrict access to a cluster. It is a type of intermediator on which users are logging into first, and then EdgeNode connects directly with the cluster and performs transmission of information. Direct connectivity between users and clusters can be disabled by using EdgeNode. On the other side, Level 2 provides authentication and access control, which are based on an authentication mechanism that guarantees that all active services and participating users are authenticated. One of the most popular authentication mechanisms is Kerberos [14]. Network encryption feature represents the Level 3 of Hadoop security mechanism. Final security line, Level 4, is HDFS encryption. If a malicious party somehow overcomes previous security levels, there is a good chance that it will get access to the compromised node from which it can exploit data blocks and get access to files at the level of the operating system. HDFS encryption is one successful way to prevent this threat.

Distribute data processing and parallelism of Hadoop mostly affect that mentioned issues have not been entirely resolved until now. By other words, no perfect security algorithm in Hadoop Environment is proposed. As can be concluded from the previous analysis, one of the biggest problems is the potential attack of malicious clients and compromising of user data. In the next section, the HDFS issue of unauthorized access to a system will be analyzed in detail.

2.4. User Access Monitoring

First, it is important to explain concretely what an unauthorized client is. Such a client does not have permission to access data or perform any operation within a cluster. A malicious user can attack Hadoop by accessing a file via HTTP protocols or via the remote procedural call (RPC). Further, it can execute malicious codes to a system and read or write arbitrary data block of a file by using a pipeline streaming data-transfer protocol. Also, it can get privileges which could provide him the capability to change the priority of assigned jobs inside the Hadoop, to delete jobs or to submit its malicious tasks. When an unauthorized user performs an operation on a data block, he bypasses a mechanism of access control. Another way of a malicious party to get unauthorized access is to intercept communications to consoles of Hadoop. It could be any communication process between a NameNode and DataNodes. When communication is intercepted, credentials or data could be stolen.

In order to make the Hadoop system secure from unauthorized users, it is from vital importance to check all system changes. These changes could be any addition or deletion of data, modification of information, node management, etc. With the purpose to provide a suitable security mechanism, a log monitor-

ing system must be deployed, and the complete Hadoop system must be audited all the time [15].

Log monitoring became essential Hadoop component for acquiring frequent information of the entire system. The problem is that the Hadoop framework does not have built-in monitoring features for detecting malicious queries or misuse of data. Further, ongoing researches still cannot precisely theoretically formulate a malicious query and its characteristics. Therefore, the universal monitoring solution is not proposed yet, and every individual usage case of HDFS and Hadoop has its monitoring functions.

The purpose of the other feature, an audit [16], is to comply with all security requirements, and it is used by MapReduce and HDFS components. An audit can be used to detect when a party accesses the system on an unauthorized way by exploiting an event log and checking an activity record. Event logs are proposed to register all user actions, from correct ones to the activities which are wrongly or maliciously performed. But it is not enough to have records of user IDs and IP addresses when logs are created. It is also strongly recommended to snapshot information about issued queries. However, another problem can occur here because there is a possibility that an attacker can delete or modify even log entries in the system. That threat is a tremendous challenge for already proposed monitoring features which continuously need to be upgraded.

Usage of log monitoring feature and a proper audit mechanism can significantly lower a possibility for security violations. Still, to build an adequate security mechanism, it is also essential to know the environment in which a system operates and to perform detail user assessment.

2.5. User and Environment Assessment

Direct access to the HDFS is required by two types of clients: developers/analysts and indirect access users. If a developer wants access permissions, it is expected that he will need access to different nodes, developer tools, log files, etc. If a data analyst wants to use HDFS resources, it is logical that he will not need the same tools as the developer, but he will require analytical tools instead. On the contrary, an indirect access user does not have reasons to use developer tools, exploit and analyze data or to use computation resources of a system. Indirect access users do not require exclusive access, and they should be involved as a part of the general security model.

It is not often enough to understand the types of users that utilize the system. In order to fully assess the risk to the security of the system, it is necessary to know the environment in which the system operates [17]. First, it is essential to determine if a system is available on a global network and connected to the internet. If that is the case, the system is open to many different threats, viruses, and potential attacks of unauthorized clients, which tries to exploit its vulnerabilities. HDFS components with internet access must have an appropriate mechanism for monitoring and continuous alert platform in the case of occurring

of unauthorized parties. A lot of time and programming effort is required to continually investigate possible threats, to develop new security patches, upgrade existing ones, and to research the latest state of development of techniques for endangering system security.

The environment should also be evaluated by its physical characteristics. The physical location of machines that are a part of a distributed system could be essential from the perspective of determining who has direct access to individual computers. Generally, machines can be stored inside a company data center, a third-party data center, or in a cloud. Based on these expectations, the security of the system must be accustomed to defend the complete structure sufficiently. A big problem can occur if the servers are hosted in a public cloud because it is challenging to tell for sure who has access to the system. In that case, the security mechanism will be much more complex and demanding, in comparison with requirements if the servers are located in private property and absolute control of a few people. Communication problems and security issues are significantly lower if cloud services are entirely avoided.

The overall security of Hadoop is improved significantly in the last few years. Most of the issues which occur today are examined and researched by experts from the field. However, absolute protection from unauthorized access does not exist, and it is an everyday struggle of developers to improve performances of distributed systems and HDFS component. Enterprise Hadoop distributions have high power in dealing with access and identity management tasks. Majority of organizations which utilize Hadoop distributed storing system have two different types of administrators: Hadoop and platform administrators. Both groups have authorized access to the files of a cluster. Regular issue is the requirement to separate duties and access restrictions of two groups. Sometimes it is necessary that an administrator does not see private information and sensitive data. Then, the feature with the capability of segregation of administrative roles and limitation of access to the desired level is required. Newer versions of Hadoop protection include authorization of different roles, file permissions, management of access lists, etc. However, these features cannot stop malicious users from unauthorized access, acquiring private information, and snapshotting content of files. Better security performances are possible if features such as different key management services are implemented within a system. One popular and highly used key management service is HDFS encryption, which provides a unique key for every application [18].

2.6. Authorization Labeling

Developers of Hadoop technology, in its beginning, mostly devoted their attention to preventing data loss accidents, while unauthorized access to HDFS and data was not correctly considered. HDFS system possesses the file permission feature that prevents a user from accidentally deleting a file system. But there was a lack of protection from an unauthorized party which wants to assume

root's identity with the intention to disclose or delete some cluster data. Also, there is a possibility that an honest party tries to access data on some inappropriate way or by mistake, which can also be labeled as illustrated in Figure 2 where an attempt of unauthorized access is detected. HDFS file permission feature is based on an authorization mechanism [19], which is not enough for complete security and protection from unauthorized users. The Hadoop is still

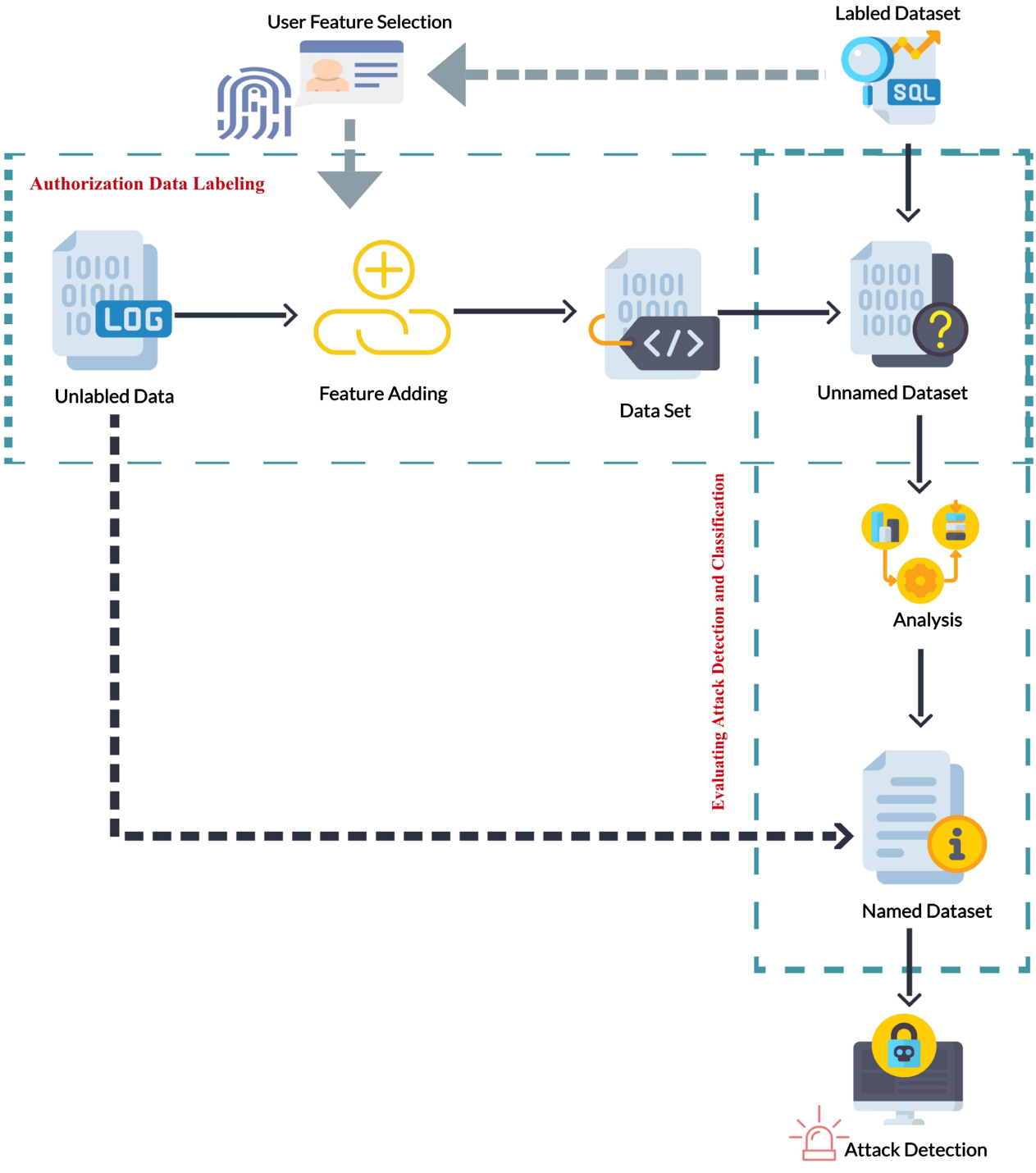

Figure 2. Detection of Untheorized Named Dataset.

vulnerable from attacks, and therefore, integrity and confidentiality of data are endangered. The system is open to abuse via manipulations of data by a malicious party which can access a network cluster. File permissions manage what operations are allowed/non-allowed for each user and what specific user can do with a single file. For instance, a file can be writable only by one user and readable by a group of users. Additionally, all other users have no rights on the file, and it is completely locked to them.

It is essential to provide the system with a mechanism which will adequately implement all data access policies within HDFS. Every authorization decision that is made on the NameNode must also be consistently executed on DataNodes. That way, unauthorized access can be stopped more easily. However, the mechanism of authorization is one of many security steps which are required for full protection of data.

2.7. Node Based Authentication Issues

The process of authentication indicates activity of proving the identity of a party to someone else. Authentication is an essential step for secure and reliable work of distributed data systems [20]. Before the communication between two parties starts, an authentication protocol runs. The task of the protocol is to establish identities between two parties, after what they can cooperate securely. Existence of such a protocol is an essential feature for building secured and protected environment.

A lack of authentication feature characterizes the elemental Hadoop framework. The first problem here is that the Hadoop does not authenticate a user or a service. However, it is vital to provide Hadoop with a mechanism that allows a user to perform a file operation only if it is verified that the user is who its claim to be. In that case, the user becomes an honest or trusted party of the system. Further, personal information of users like names, IP addresses, credit card numbers, etc., should be protected by the framework and access of a random user to this data should be restricted. Generally, the authentication mechanism must include all regulatory requirements which are essential for the protection of the data. The typical case is building such a mechanism which will allow different security levels for different sets of data. For example, data with a low level of security can be some public or anonymized data which can be shared between all network parties without any restriction. On the other hand, the highest level of protection should be provided for personal and sensitive data, allowing only chosen parties to have access rights. If a proper authentication mechanism is built, Hadoop will effectively determine if a user has required permissions to perform an action. Every action of a user is followed by the mechanism which requires user credentials whenever the user tries to access the data.

Three main components are required for the authentication process. The first one is a string in the form of a username or a service name. Additionally, a password could be added to each user. If the password is added, the protection

of it can be achieved by applying the salted password hashing algorithm [21]. The second, optional component, is the instance which defines the specific role of a user or a name of a host on which a service is running. The third one is the realm itself, and it can be described as a type of DNS domain. By the default set up, DataNodes do not execute any access checkups on input points of data blocks. If an unauthorized client wants to perform an operation to some data block, it is enough for him only to know the block ID. Also, it is possible for all users (honest and malicious) to write a new data block to DataNodes. In that case, any malicious party can overload the system with a large amount of garbage data, and computation power of Hadoop will be used for processing spam files. This activity is known as a denial of resources. A malicious party is submitting multiple jobs to the system which tries to perform all required tasks and spends the majority of available cluster resources on these tasks. The system is occupied with undesirable activities, and other users are disabled to process real jobs optimally.

Lack of authentication problem can result also with allowing a random user to start random service on any machine. When NameNode registers a malicious user, it will automatically begin to receive data blocks from a cluster. It is the consequence of the replication feature of Hadoop, where data is replicated three times within the system from the safety reasons. One way of solving this problem is the implementation of a feature which can restrict machine registration as DataNodes. It is shown in practice that the main element of this feature is dfs.hosts property that contains names of hosts which are allowed to connect and register with a NameNode. The property is physically stored inside hdfssite.xml. Initially, this feature is turned off and needs to be activated individually. While it is off, a malicious client can communicate with any DataNode, read existing data blocks, or add new malicious data blocks.

Secure communication of new Hadoop versions is provided by using the Transport Layer Security (TLS) protocol [22]. This protocol ensures communication privacy between all nodes and name servers. That way, HDFS can protect the transfer of data through a system and guarantee the safety of all honest parties.

2.8. Threats and Possible Attacks

In the previous subsection, unauthorized access was described from the perspective of Hadoop. The most important part of developing a secure and reliable system is to have good insight at possible threats of unauthorized access [23] and what an unauthorized party can do to a network. If an intruder is known, the defense mechanism and security feature could be developed easier.

Port Scanning Attack is a malicious method for collecting data that disclose information about openness of services and ports. Also, the technique records ways in which services are responding to individual queries. On the other hand, a Dictionary Attack is continuously entering word by word as a password until it hit the right one and defeated a mechanism of authentication. Another working

mode of Dictionary Attack is to determine the decryption key of a file or a document. The third type is Remote to User Attack, which begins its activity when a malicious party sends data packets to a target node through the network. That way, an unauthorized party is looking for vulnerabilities of the node and exploits them. If it does not find a defect on the attacked node, it repeats the procedure from the beginning with another node. In this way, a malicious party is searching for weak components which are suitable for endangering. Computer Exploit Attack is based on an intruder which attacks a system and concentrate on a node with some specific vulnerability, trying to take advantage of it.

An unauthorized user can also attack the NameNode, trying to put it out of service and to stop operational state of a cluster. That kind of attack is called Attack on NameNode Availability. The next one is the Man in The Middle Attack. An intruder first accomplishes unauthorized access to the system, and he is positioning himself between two honest nodes on such a way that all the commutation between two parties goes only through the malicious party. When he becomes intermediator between two parties, data can be easily copied, deleted, or modified. In the best scenario for Hadoop, the Man in The Middle Attack will be only used for monitoring purposes without any modification of transmitted information. In the end, the Bypass Authentication Attack principles of work are based on disabling a system to perform its access policies and to restrict unauthorized access to the network.

3. Honey-Based Intrusion Detection

There are plenty of published research papers which cover recommendation and suggestion topics for effective and optimal usage of Hadoop technology [24] [25]. The first suggestion, which is essential for the process of restricting unauthorized access, is to use Kerberos. Kerberos is easy to deploy feature, which is a foundation of security. It is an effective solution for node validation and user authentication. Another recommendation is to perform encryption of file layer with the purpose to protect data storage. Encryption prevents malicious users, as well as administrators, to directly inspect data by accessing data nodes. Still, encryption cannot be used for protection from credentialed user access with multi-key support. Encryption is also not usable if a malicious party gains the keys of encryption. Security of the keys can be achieved by using a key management service. This service distributes certificates and keys, and it is an effective solution when it is used together with HDFS encryption zones. Next recommendation is to use Apache Ranger, which ensures the usage of security policies for protecting cluster data and establishes a variety of configurations that deal with a deployment validation issue.

Investigation of suspicious system activities, unauthorized access attempts, and failure diagnostics can be performed if an activity record exists. Hadoop has built-in functions for creating and managing logs of events, and additionally, even clusters can be used for storing logs. The recommendation is to use LogS-

tash or Kafka as additional features for management of recorded logs and streaming of applications. The final suggestion will be the implementation of secure communication protocols between an application and nodes, and between the nodes themselves. That can be performed by introducing a Secure Sockets Layer (SSL) and Transport Layer Security (TLS) to the system. These two security layers protect the entire communication network of the system. The drawback of using security protocols can be reduced system performances during the transfer of extensive data.

To conclude, efficient management features, secure authentication, and encryption tools provide better security of Hadoop and prevent successfully unauthorized malicious users from accessing the system, manipulating data, and stealing sensitive information. In the next section, honeypot nodes, as one type of efficient security mechanism for monitoring malicious activities and collecting valuable data about attempts of unauthorized access, will be presented.

4. Deploying Honeypot Nodes

A honeypot [26] is a server whose purpose is to detect unauthorized user by simulating a real system and represents a virtual trap for malicious parties. There are no restrictions about operating systems in which it can operate, and it can be used as a universal security solution. This feature looks like any other real server within a system, and it is a tool that can be precious for early warnings against intruders. However, a Honeypot node only includes false data and simulates fake transactions which are not part of verified network transactions. It can be located in the firewall or out of it (Figure 3), and it is an excellent mechanism to familiarize with attack techniques of malicious parties. Also, it can be used to find possible vulnerabilities of a system and to warn a system about them. Honeypots can take a form of a file, data record, space of unused IP address, etc. Actually, they can run a random number of services, depending on specific

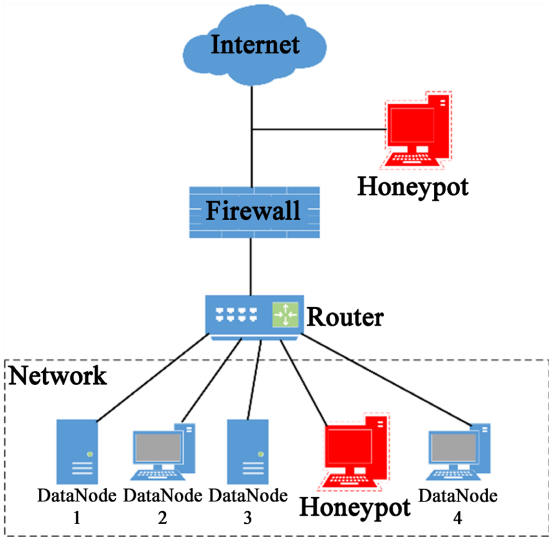

Figure 3. Honeypot deployment possibilities.

requirements. In order to be attractive for the attacks, honeypots are designed to look completely unprotected and vulnerable. By other words, they pretend to be the weakest input points of the system, which are the most suitable for potential attacks.

During the operational time, honeypots register each activity they have with users. However, since they do not provide any legitimate service, all their activities can be considered as unauthorized. Because of that, they do not have any significance or contribution value to network performances or data management. When a honeypot is attacked, all records about the attack are examined, and information about intruder can be extracted. Honeypot importance relies on information which the system can acquire by using it [27]. Acquired information can include data about unauthorized users, types of attack which they perform and possible consequences which they can cause. Honeypots can detect even attacks which are brand new and unfamiliar to a system until now. Based on acquired insights, new security mechanisms can be proposed more easily.

Generally, honeypot technology can provide advantages in comparison to other security solutions from a few reasons. First, honeypots do not process legitimate traffic, a significantly smaller quantity of data is used that way, and that implies fewer possible mistakes in the detection of malicious activities. Further, they collect only high-value information and just logs of unauthorized activities, so their databases are small and easy for maintaining. Finally, they can be used in encrypted environments, and do not need signatures of attacks to be effectively used.

The categorization of a single honeypot node depends on its interaction level with unauthorized malicious users and on possible services which it can provide during the work. Based on these conditions, two types of honeypots can be distinguished: low-interaction and high-interaction honeypots.

Low-interaction honeypots are mostly daemons, which emulate services and pretend to be real applications. Their purpose is to scan and detect unauthorized access and sources of undesirable activities [28]. Because they do not emulate all services, but just chosen ones, they require only a limited number of functionalities from a server. Emulation of services implies that these honeypots are a safe solution if possible attacks of malicious parties succeed as can be seen in **Figure 4**. If they become infected and hacked, no harm to a real system will be made. The disadvantage is they do not provide a right level of realism because they emulate only some parts of the operating system such as specific network applications and core services.

High-interaction honeypots, in contrary with low-interaction units, are regular physical machines connected to a network in a traditional way, and with a unique IP address for each of them. Because of the physical property, they are expensive to maintain and install. They do not emulate applications and do not propose fake services, but traditionally run processes. Applications operate in their real environments, but with the additional possibility of being infected.

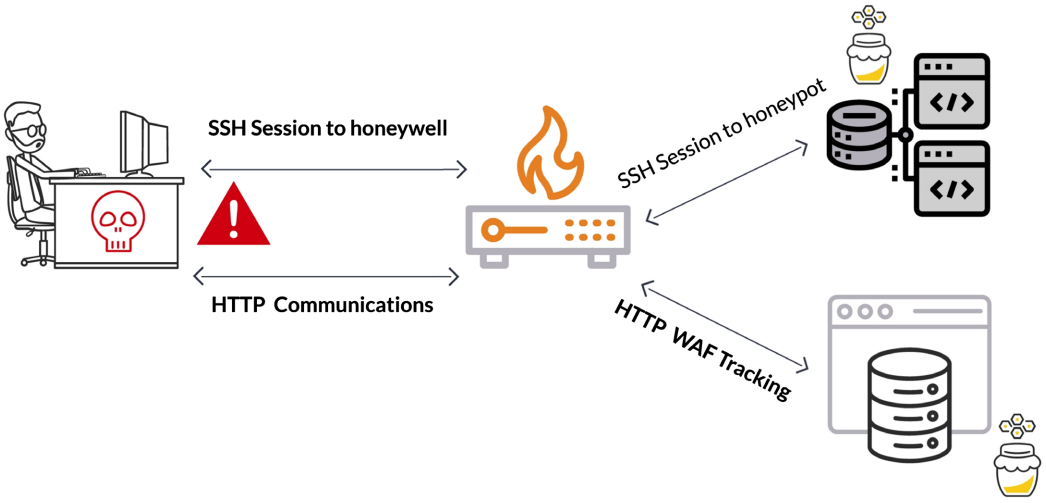

Figure 4. Honeypot service emulation.

Malicious users can attack these honeypots as any real operating system. They can be jeopardized completely, and unauthorized and malicious clients can get full access to a network. Intruders can even use infected honeypots as a tool for strengthening the attack on the system. If an attacker comprises the system, there is no operational restriction what he can do within the system. Beside from increased possibility of being infected, deficiency of high-interaction honeypots is that there could be a problem to monitor activities of a real system adequately.

It is always beneficial to have multiple operating honeypots inside a system because the number of them directly influences the accuracy level of information and the amount of valuable data which is collected. Rising needs for obtaining such a system have inspired proposal of another way of using high-interaction mechanisms: their installation and usage of virtual machines. Virtual honeypots [29] are a good alternative for real machines in term of reducing maintaining costs, which is the consequence of the requirement for fewer computers inside a system. Also, they are simpler to setup in comparison to high-interaction honeypots and provide a system with the option to use multiple honeypots on one server. That can be beneficial from reason that instances which represent multiple honeypots can be multiplexed on a single machine. This characteristic can be beneficial when there is a requirement to work with large address spaces, in which case, it is even impossible to provide a physical honeypot for each IP address. Further, if the rebooting process is required, a virtual machine environment will be ready significantly faster in comparison with a physical machine environment. Another reason for using virtual machines is improved monitoring performances of operating systems. A real system does not have proper monitoring capabilities of an operating system as a virtual machine has. A virtual machine can provide notably better monitoring options with a good chance for collecting more massive amounts of useful data. Additionally, system calls, activities on a network, and management of resources can be observed more efficiently on a virtual machine. Finally, virtual honeypots can be used in parallel

with high-interaction nodes. A load of high-interaction honeypots can be reduced if traffic is pre-processed by exploiting virtual honeypots.

5. Future Work

In the previous section, Honeypot technology was explained. Future work of the authors of this paper will be based on further research of virtual honeypots and proposing a new solution which will eventually improve overall HDFS security and efficiently restrict unauthorized access of malicious parties. Concretely, special attention will be devoted to the development of new on-demand honeypot structures. As an explanation, it should be mentioned that there are two possible approaches to implement all honeypots: their generation in advance within a system, and generation on demand only after the system needs them. The second way provides additional system efficiency since it will not require the additional consumption of resources for maintaining honeypots during the time instances when the system is not attacked, and there is no presence of malicious parties.

An algorithm we want to research in future work is to propose a solution of scalable distributed file sharing system which proposed on [30] and then creating virtual honeypots on demand, where the unauthorized party will be detected first, and then redirected to a generated honeypot. The redirection will be performed when an attacker tries to compromise any system node. The feature we will try to develop is a new type of Intrusion Detector mechanism whose primary goal is to detect an intrusion quickly by using an artificial neural network. Neural networks are successful function approximators that can be used for different purposes. In our case, the idea is to empirically find an adequate neural network, to train it properly and build a capable threat detector. Each detection of malicious activity should be followed by the registration of a victim node IP address. The idea is to use a neural network on a proper log file, which is generated by the Windows Firewall system. Examination of the log file by the network will provide detection of undesirable logs and activities.

The new system will also require the development of honeypot component for deploying a virtual node which represents a replica of the attacked node. The component will be active when the Neural Network Intrusion Detector registers a request which is identified as malicious. When a virtual system is deployed, the malicious party will be redirected to a virtual honeypot. Additionally, the proposal should include a module for generating log files periodically and for storing these logs in a database. Finally, another component of the proposed system should perform the searching procedure for log files of a system which is under attack. This component should analyze a client IP address and determine if a party should be directed to a network or redirected to a honeypot.

Mentioned considerations in the last two paragraphs will be the initial base for further scientific researches in this field, and starting point for the development of new Hadoop security mechanisms.

6. Conclusions

This research was motivated by the importance of Hadoop technology on the development of the IT sector in the 21st century and by great opportunities which the technology provides to distributed networks. The research task was to determine the current state of the art of the technology, to identify its strengths and to perform a review of observed deficiencies, with the accent on examination of unauthorized access issue. Following that, in the first part of the paper, a brief Hadoop summary and fundamental challenges are shown, as well as security challenges of Hadoop Distributed File System (HDFS). Then, HDFS architecture is explained briefly, required HDFS features are introduced and functionalities of NameNodes and DataNodes are described. The last part of the Section I was dedicated to different security challenges of HDFS, authorization and authentication issues, and some other security problems. Majority of described issues are a direct consequence of parallelism and distributed data processing of Hadoop technology and represent a real challenge for scientists and developers to overcome them effectively.

The most crucial section of this review paper, Section 2, represents a detail examination of the issue of unauthorized access to a Hadoop system. The starting points for this section are explanations of HDFS storage operating principles and activities of storing data inside DataNodes. Also, four possible types of Hadoop users are presented, their properties and operating modes during work with distributed systems. At the beginning of the explanation of security issues, different security levels of Hadoop technology are explained. Four different security layers and mechanisms which can be used to enable these layers are presented. Next, unauthorized users are described, as well as malicious activities which they can perform within a system. Audit and Log Monitoring features as precious components for complying with all security requirements and acquiring information of a system are also described. Part C of Section 2 explains the importance of knowing the system environment, physical characteristics, and all accompanying features. It is required to determine an openness level of the system to a global network and to propose security solutions on behalf of that. Additionally, this subsection describes the differences between direct and indirect access users.

Authorization and authentication issues are examined in detail in subsections 2.5 and 2.6. The importance of two components for overall Hadoop security is highlighted, and basic requirements for their development within a system are analyzed. It is explained what consequences can occur if proper mechanisms are not implemented within a system. In the last part of Section 2, some concrete recommendations for resolving issues of unauthorized access are presented. Additionally, encryption and key management services are explained, as well as reasons for usages of software packages such as Apache Ranger, LogStash, and Kafka.

Section 3 is dedicated to the review of Honeypot nodes. These features are in-

troduced to the paper from the reason they will represent the starting point for further research attempts of the authors. It is explained how honeypots operate, their purpose, and possible benefits of using them. Further, the categorization of the feature is shown. Differences between low-interaction and high-interaction honeypots are presented as well as possible deficiencies of both types of nodes. The main focus of the section is then put on virtual honeypots, the virtual machine features which are very popular nowadays. They can replace physical honeypots easily, their monitoring power is significantly higher in comparison with traditional units, and multiple virtual honeypots can be installed on a single machine.

Benefits of using virtual honeypots motivated the authors of the paper to begin their new researches in this domain. In Section 4 of the paper, the future work and initial considerations are presented. The initial idea is to propose a new type of on-demand honeypot mechanism, which will be based on a novel Intrusion Detector. This detector should be developed by using an artificial neural network and be capable of efficient detection of unauthorized activities and undesirable logs to a system. All main components which should be included in the new structure are briefly described, and desired performances are mentioned. The overall goal of this section was to make an introduction for further scientific work in the field of Hadoop and HDFS security.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Saraladevi, B., Pazhaniraja, N., Paul, P.V., Basha, M.S.S. and Dhavachelvan, P. (2018) Big Data Security Challenges: Hadoop Perspective. *International Journal of Pure and Applied Mathematics*, **120**, 11767-11784.

[2] Kumar, V. and Chaturvedi, A. (2017) Challenges and Security Issues in Implementation of Hadoop Technology in Current Digital Era. *International Journal of Scientific & Engineering Research*, **8**, 984-990.

[3] Abouelmehdi, K., Beni-Hssane, A., Khaloufi, H. and Saadi, M. (2016) Big Data Emerging Issues: Hadoop Security and Privacy. 2016 5*th International Conference on Multimedia Computing and Systems* (*ICMCS*), Marrakech, 29 September-1 October 2016, 731-736. https://doi.org/10.1109/ICMCS.2016.7905621

[4] Inukollu, V.N., Arsi, S. and Ravuri, S.R. (2014) Security Issues Associated with Big Data in Cloud Computing. *International Journal of Network Security & Its Applications* (*IJNSA*), **6**, 45-56. https://doi.org/10.5121/ijnsa.2014.6304

[5] Kumar, M.P. and Pattem, S. (2017) Security Issues in Hadoop Associated with Big Data. *IOSR Journal of Computer Engineering* (*IOSR-JCE*), **19**, 80-85.

[6] Singh, K. and Kaur, R. (2014) Hadoop: Addressing Challenges of Big Data. 2014 *IEEE International Advance Computing Conference* (*IACC*), Gurgaon, 21-22 February 2014, 686-689. https://doi.org/10.1109/IAdCC.2014.6779407

[7] Honnutagi, P.S. (2014) The Hadoop Distributed File System. *International Journal*

of Computer Science and Information Technologies, **5**, 6238-6243.

[8] Shvachko, K., Kuang, H., Radia, S. and Chansler, R. (2010) The Hadoop Distributed File System. 2010 *IEEE* 26*th Symposium on Mass Storage Systems and Technologies* (*MSST*), Incline Village, NV, 3-7 May 2010, 1-10. https://doi.org/10.1109/MSST.2010.5496972

[9] Shafer, J., Rixner, S. and Cox, A.L. (2010) The Hadoop Distributed File System: Balancing Portability and Performance. 2010 *IEEE International Symposium on Performance Analysis of Systems & Software* (*ISPASS*), White Plains, NY, 28-30 March 2010, 122-133. https://doi.org/10.1109/ISPASS.2010.5452045

[10] Borthakur, D. (2005) The Hadoop Distributed File System: Architecture and Design. The Apache Software Foundation, Wakefield, MA.

[11] Rani, N.S. and Lakshmi, N.V.M. (2018) Major Challenges with Hadoop Distributed Framework: An Overview. *IADS International Conference on Computing, Communications & Data Engineering*, Tirupati, 7-8 February 2018.

[12] Sajwan, V., Yadav, V. and Haider, M. (2015) The Hadoop Distributed File System: Architecture and Internals. *International Journal of Combined Research & Development* (*IJCRD*), **4**, 541-544.

[13] Sharma, P.P. and Navdeti, C.P. (2014) Securing Big Data Hadoop: A Review of Security Issues, Threats and Solution. *International Journal of Computer Science and Information Technologies* (*IJCSIT*), **5**, 2126-2131.

[14] Neuman, B.C. and Ts'o, T. (1994) Kerberos: An Authentication Service for Computer Networks. *IEEE Communications Magazine*, **32**, 33-38. https://doi.org/10.1109/35.312841

[15] Kadam, S.R. and Patil, V. (2017) Review on Big Data Security in Hadoop. *International Research Journal of Engineering and Technology* (*IRJET*), **4**, 1362-1365.

[16] Kumar, A., Nikitha, S. and Sundarajan, P. (2015) A Framework of Privacy Scribed Thesis on Cloud Environment and Hadoop Based Big Data. *International Journal of Technology Enhancements and Emerging Engineering Research*, **3**, 114-118.

[17] Reddy, Y.B. (2015) Access Control Mechanisms in Big Data Processing. *Proceeding* (829) *Software Engineering and Applications*/831: *Advances in Power and Energy Systems*, Marina del Rey, 26-27 October 2015. https://doi.org/10.2316/P.2015.829-006

[18] Parmar, R.R., Roy, S., Bhattacharyya, D., Bandyopadhyay, S.K. and Kim, T. (2017) Large-Scale Encryption in the Hadoop Environment: Challenges and Solutions. *IEEE Access*, **5**, 7156-7163. https://doi.org/10.1109/ACCESS.2017.2700228

[19] Gupta, M., Patwa, F., Benson, J. and Sandhu, R. (2017) Multi-Layer Authorization Framework for a Representative Hadoop Ecosystem Deployment. *Proceedings of the* 22*nd ACM Symposium on Access Control Models and Technologies* (*SACMAT*), Indianapolis, 21-23 June 2017, 183-190. https://doi.org/10.1145/3078861.3084173

[20] Kanyeba, M. and Yu, L. (2016) Securing Authentication within Hadoop. 2016 *International Conference on Electrical, Mechanical and Industrial Engineering* (*ICEMIE*), Phuket, 24-25 April 2016, 100-103. https://doi.org/10.2991/icemie-16.2016.25

[21] Blocki, J. and Datta, A. (2016) CASH: A Cost Asymmetric Secure Hash Algorithm for Optimal Password Protection. 2016 *IEEE* 29*th Computer Security Foundations Symposium* (*CSF*), Lisbon, 27 June-1 July 2016, 371-386. https://doi.org/10.1109/CSF.2016.33

[22] Turner, S. (2014) Transport Layer Security. *IEEE Internet Computing*, **18**, 60-63. https://doi.org/10.1109/MIC.2014.126

[23] Shahane, R., Shruthi, P., Viswanadh, B. and Abhilash, D. (2019) A Comparative Study of Various Security Threats and Solutions for the Security of Hadoop Framework in Terms of Authentication and Authorization. *International Journal of Innovative Technology and Exploring Engineering* (*IJITEE*), **8**, 1599-1602.

[24] Tripathi, S., Gupta, B., Almomani, A., Mishra, A. and Veluru, S. (2013) Hadoop Based Defense Solution to Handle Distributed Denial of Service (DDoS) Attacks. *Journal of Information Security*, **4**, 150-164. https://doi.org/10.4236/jis.2013.43018

[25] Daoudhiri, K., Abouchabaka, J. and Rafalia, N. (2018) Attacks and Countermeasures in a Hadoop Cluster. *International Journal of Scientific & Engineering Research*, **9**, 66-70.

[26] Provos, N. (2004) A Virtual Honeypot Framework. *Proceedings of the* 13*th USENIX Security Symposium*, San Diego, CA, 9-13 August 2004, 1-14.

[27] Samu, F. (2016) Design and Implementation of a Real-Time Honeypot System for the Detection and Prevention of Systems Attacks. Master's Thesis, St. Cloud State University, St Cloud, MN.

[28] Holz, T. and Raynal, F. (2005) Detecting Honeypots and Other Suspicious Environments. *Proceedings of the* 2005 *IEEE Workshop on Information Assurance and Security*, West Point, NY, 15-17 June 2005, 29-36.

[29] Vârlan, C., Rughiniş, R. and Purdilă, O. (2010) A Practical Analysis of Virtual Honeypot Mechanisms. *Proceedings of the* 9*th RoEduNet IEEE International Conference*, Sibiu, 24-26 June 2010, 25-30.

[30] Ekwonwune, E. and Ezeoha, B. (2019) Scalable Distributed File Sharing System: A Robust Strategy for a Reliable Networked Environment in Tertiary Institutions. *International Journal of Communications*, *Network and System Sciences*, **12**, 49-58. https://doi.org/10.4236/ijcns.2019.124005