applied system innovation

MDPI

Article

Digital Forensics Readiness in Big Data Networks: A Novel Framework and Incident Response Script for Linux–Hadoop Environments

Cephas Mpungu *, Carlisle George and Glenford Mapp

Department of Computer Science, Middlesex University, The Burroughs, London NW4 4BT, UK; c.george@mdx.ac.uk (C.G.); g.mapp@mdx.ac.uk (G.M.)

* Correspondence: cm1677@live.mdx.ac.uk or mcephas@gmail.com; Tel.: +44-7487550360

Abstract: The surge in big data and analytics has catalysed the proliferation of cybercrime, largely driven by organisations' intensified focus on gathering and processing personal data for profit while often overlooking security considerations. Hadoop and its derivatives are prominent platforms for managing big data; however, investigating security incidents within Hadoop environments poses intricate challenges due to scale, distribution, data diversity, replication, component complexity, and dynamicity. This paper proposes a big data digital forensics readiness framework and an incident response script for Linux–Hadoop environments, streamlining preliminary investigations. The framework offers a novel approach to digital forensics in the domains of big data and Hadoop environments. A prototype of the incident response script for Linux–Hadoop environments was developed and evaluated through comprehensive functionality and usability testing. The results demonstrated robust performance and efficacy.

Keywords: digital forensics; digital forensics readiness; incident response; big data; Linux; Hadoop

check for updates

Citation: Mpungu, C.; George, C.; Mapp, G. Digital Forensics Readiness in Big Data Networks: A Novel Framework and Incident Response Script for Linux–Hadoop Environments. *Appl. Syst. Innov.* **2024**, *7*, 90. https://doi.org/10.3390/asi7050090

Academic Editor: Friedhelm Schwenker

Received: 17 July 2024
Revised: 26 August 2024
Accepted: 23 September 2024
Published: 25 September 2024

1. Introduction

In today's data-driven landscape, the Hadoop platform and its derivatives such as Apache Spark, Apache Hive, and Cloudera have emerged as the cornerstone of big data storage and analytics [1–3]. With their ability to efficiently manage vast amounts of data across distributed clusters, these platforms have become the preferred choice for most organisations seeking to harness the power of big data for insights and decision-making [4,5]. From e-commerce giants to healthcare providers, the adoption of Hadoop-based solutions has proliferated across various sectors, fuelling the expansion of business intelligence (BI) capabilities and transforming industries.

BI, facilitated by the analysis of large datasets, has revolutionised how organisations operate, enabling them to uncover valuable insights and optimise processes [6]. For instance, retail companies utilise customer data to personalise marketing campaigns, while financial institutions leverage data analytics to detect fraudulent activities. Such examples underscore the critical role of big data platforms like Hadoop in driving innovation and competitiveness across diverse domains.

Moreover, the exponential growth of big data has also led to an alarming increase in cybercrime [7,8] and data privacy issues [9,10]. As organisations amass significant volumes of sensitive data, they become prime targets for cybercriminals aiming to exploit weaknesses in data storage and processing systems [11]. This trend can exacerbate data privacy concerns, particularly because most organisations often prioritise data analytics for BI over transparency and comprehensive data protection efforts [9]. Furthermore, the intricate nature of Hadoop infrastructures may obscure these risks, making it challenging for digital forensics practitioners to effectively investigate security incidents within these environments [11].

To address these gaps, this research proposes a conceptual digital forensics readiness (DFR) framework tailored for both the cloud and on-premises environments. DFR is defined as the proactive steps and preparations taken by an organisation to effectively perform digital investigations when faced with an incident such as a cyberattack, security breach, or legal investigation [12,13]. The proposed framework's configuration consists of a standard Hadoop cluster running on a Linux operating system (OS), BI data nodes, and forensic nodes. Some of the forensic nodes are dedicated to evidential data collection whilst one is configured to capture all relevant security policy management and compliance information, guaranteeing an audit trail and simplifying digital investigations. Nevertheless, it is crucial to recognise that this research does not delve into the specifics of implementing security policy configurations. Additionally, the proposed framework is conceptual and serves as a foundational guide that organisations can tailor to their specific needs and requirements.

Furthermore, a novel incident response script (IR-script) was developed utilising the powerful scripting capabilities of the Linux Bash shell, standard Linux commands, and Hadoop commands relevant to DFR. Linux scripts help automate tasks, manage system operations, and facilitate various administrative activities. The IR-script provides a comprehensive solution for first responders conducting investigations within Linux–Hadoop cluster environments. By integrating automation to extract supporting evidential information, the script enables investigators to efficiently gather crucial preliminary artefacts necessary for thorough forensic analysis. Upon execution, the script systematically probes the Linux–Hadoop cluster, extracting supporting evidence from both the underlying Linux operating system and the Hadoop cluster. This information is particularly valuable for investigators at the initial and concluding stages of their investigations [5]. Its user-friendly design caters to both seasoned investigators and those with limited knowledge of Linux and Hadoop cluster environments, ensuring accessibility and usability in diverse investigative scenarios.

The primary objective of this research is to introduce a conceptual standard and customisable DFR framework designed for big data networks, alongside a customisable IR-script specifically designed for Linux–Hadoop systems. Due to resource and time constraints in setting up and evaluating the entire big data network, this research paper focuses solely on developing and evaluating the effectiveness of the IR-script. To evaluate its efficacy, functionality, and usability, tests were conducted on three separate Hadoop systems: Amazon Web Services (AWS), Oracle VirtualBox (VB), and Hortonworks Data Platform (HDP) Sandbox.

The rest of this paper is organised as follows: Section 2 summarises the focus of this research and introduces the research questions. Section 3 presents a thorough literature review, examining existing research and identifying gaps that this research aims to address. Section 4 details the research methodology employed in this study, providing a comprehensive overview of the approaches and techniques utilised to achieve the research objectives. Section 5 discusses the proposed system, including its design, architecture, and the theoretical underpinnings that inform its development. Section 6 elaborates on the prototype configuration and evaluation, describing the implementation process, testing environments, and the results obtained from the evaluations. Finally, Section 7 concludes the paper, summarising the key findings, implications, and potential areas for future research.

2. Research Focus

The focus of this research was to propose a comprehensive conceptual framework for enforcing DFR within big data networks. Additionally, a novel IR-script was developed for extracting supporting evidential information, which is particularly valuable at the start and end of investigations within Linux–Hadoop cluster environments.

The research addressed two primary research questions (RQs):

- RQ1: How can DFR be effectively implemented in an existing Linux–Hadoop big data network without disrupting its operations?

- RQ2: Can digital forensics investigations within a Linux–Hadoop cluster be simplified for both experienced and inexperienced investigators, considering the various configurations of different Hadoop environments?

3. Literature Review

This Section focuses on an extensive examination of existing research within the fields of digital forensics, digital forensics readiness, big data, business intelligence, Linux, and Hadoop environments. Guided by the research questions raised in Section 2, the objective is to gather insights and identify gaps in current knowledge and practises and to inform the identification of requirements for the proposed framework and the IR-script.

3.1. Background

In the rapidly evolving landscape of information technology, the advent of big data has revolutionised the way organisations manage and analyse vast volumes of data [2,4,6,14]. According to Phillip Russom [6], big data analytics denotes the use of advanced techniques on large datasets, combining the vast scale of big data with sophisticated analytics to drive a major trend in BI.

Big data technologies [4], particularly within wireless networks, present both opportunities and challenges that make wireless networks inherently more susceptible to attacks compared to their wired counterparts, due to factors such as signal interception and unauthorised access [15–17]. As organisations increasingly rely on big data for critical operations, ensuring the security and integrity of these networks becomes paramount.

Security incidents and data privacy issues within big data ecosystems have garnered significant attention in recent years [10]. The sheer volume, variety, and velocity [6] of data in big data environments make them attractive targets for cybercriminals seeking to exploit vulnerabilities and compromise sensitive information. Furthermore, the prioritisation of BI needs and profits over personal data privacy by some big tech organisations remains a cause for concern [9]. Examples of big tech organisations that have faced criticism in the past for prioritising BI needs and profits over personal data privacy include Facebook, Google, Amazon, Apple, and Microsoft [9]. Incidents such as unethical practises, data breaches, unauthorised access, and insider threats pose significant risks to organisations, highlighting the importance of implementing robust security measures and proactive monitoring mechanisms [8,10]

DFR plays a crucial role in mitigating the impact of security incidents and ensuring auditable accountability within organisations [18] as well as big data environments. DFR is derived from the term digital forensics. According to Sachowski [18] (p. 11), digital forensics is denoted as scientific methodologies, principles, and techniques applied to law to ensure the admissibility of digital evidence in a court of law. The researcher also discussed Locard's exchange principle [18] (pp. 11–12). The principle states that anything or anyone interacting with an environment (including digital environments) takes something or leaves behind something (evidence). A forensically ready digital environment (DFR) would thus be well suited to capture this evidence to aid in future investigations.

DFR involves setting up policies, procedures, and technical systems that allow for the swift and accurate collection, preservation, and analysis of digital evidence [16,18]. By proactively preparing for potential security breaches and establishing forensic capabilities, organisations can effectively investigate and respond to incidents, thereby minimising damage and preserving evidential data for legal proceedings.

Aside from security breaches, Sachowski [18] (pp. 65–66) discussed other "multiple business scenarios where digital forensics can be applied to manage business risk". These include cybercrime impact validation, solving disputes (internal and external), compliance adherence, electronic discovery (e-discovery) support, and support for contractual and commercial agreements. However, conducting digital forensics investigations within complex big data ecosystems, particularly those built on Apache Hadoop, still presents unique challenges as discussed by the researchers Asim et al. [2], Harshany et al. [11], Oo [19], and Thanekar et al. [20].

Apache Hadoop's distributed file system (HDFS) has emerged as the most preferred platform for processing and analysing big data [2,21] due to its scalability, fault tolerance, and cost-effectiveness. Its ecosystem of tools and frameworks, including Apache Spark, offers unparalleled capabilities for distributed computing and data processing. However, the distributed nature of Hadoop environments, coupled with the sheer volume of data stored across multiple nodes, complicates digital investigations [11,19]. Extracting, correlating, and analysing data from disparate sources within a Hadoop cluster requires specialised skills and tools, making forensic investigations a daunting task.

Additionally, the Linux OS is the most preferred choice for Hadoop deployments [1,3] due to its robust performance, security features, and compatibility with open-source software [22,23]. Linux offers a stable and customisable environment that aligns well with the needs of large-scale data processing. Its native support for Hadoop's underlying technologies, such as the Java Runtime Environment (JRE) and file system utilities, ensures seamless integration and optimal performance. Moreover, the open-source nature of Linux complements the Hadoop ecosystem, allowing organisations to tailor their infrastructure without the constraints of proprietary software. This synergy between Linux and Hadoop enhances the efficiency and reliability of big data operations, further solidifying their dominance in the field of data analytics and processing.

In this context, this research focuses on implementing DFR within a Linux–Hadoop environment. However, it is also important to acknowledge that Hadoop can run on Windows and other Unix-like systems, even if these platforms are less commonly used in production environments.

3.2. Review of Related Work

According to research conducted by Sremack [5] (p. 43), forensic evidence within a Hadoop big data environment can be categorised into three types:

- Supporting evidential information: This includes data that aid in identifying evidence or provide context regarding the operations and configurations of the Hadoop cluster.
- Record evidence: This encompasses any data that are processed within Hadoop, such as HBase data, sequence files, text files used in MapReduce jobs, or output generated by Pig scripts.
- User and application evidence: Comprises log and configuration files, analysis scripts, MapReduce logic, metadata, and other customisations or logic applied to the data. It is particularly valuable in investigations where there are questions about how the data were analysed or generated.

The challenge for forensic investigators in a Big Data environment like Hadoop lies in the sheer volume of data and its distribution across multiple nodes [11]. Unlike traditional systems where imaging a single hard drive might capture all necessary data, investigators in Hadoop must first identify supporting evidential information, e.g., logs and configuration files, the number of DataNodes, replication factor, etc. These files help pinpoint where evidence is stored and whether data archives exist, making them crucial during both the initial identification and later stages of an investigation as investigators work to understand the operations of the Hadoop cluster [5].

Given the complexity of conducting digital investigations within Linux–Hadoop clusters, Thanekar et al. [20] conducted a study titled "A Study on Digital Forensics in Hadoop". Their work highlighted the importance of understanding Hadoop's internal mechanisms for digital forensics. It emphasised the relevance of different files generated during Hadoop's processes for forensic analysis. The study focused on leveraging these files for digital forensics and described their roles. Using the open-source tool Autopsy, the study demonstrated efficient methods for identifying key files relevant to Hadoop digital forensics.

Additionally, Harshany et al. [11] focused on extracting forensic artefacts from HDFS metadata. Their study explored the effectiveness of using specific metadata from the HDFS data storage layer to reconstruct file system operations and link data to their physical storage locations. By mapping these data, evidence can be prioritised and targeted for preservation or further analysis.

The motivation for the current research paper builds upon the work of Thanekar et al. [20] and Harshany et al. [11] to propose an enhanced Linux–Hadoop big data DFR framework. It is important to note that the researchers focused on extracting digital forensics artefacts from the Hadoop environment with less focus on the underlying Linux OS. However, in Linux–Hadoop cluster setups, the confidentiality, integrity, and availability (CIA) of a Hadoop cluster big data environment may also be compromised through the underlying Linux OS without necessarily altering HDFS metadata. This can be carried out by exploiting various vulnerabilities and attack vectors specific to the OS, such as privilege escalation, direct disc access, and the manipulation of non-HDFS components like MapReduce and YARN. In that regard, it is important that the effective implementation of DFR within a Linux–Hadoop environment should always consider both the Linux OS and Hadoop environments.

3.3. Conclusion

The reviewed literature underscores the critical importance of ensuring security and DFR in big data environments, particularly within Linux–Hadoop clusters. The advent of big data has transformed organisational data management and analysis, presenting both significant opportunities and challenges. The vulnerability of big data networks to attacks necessitates robust security and DFR measures. The key findings of the literature review highlighted the increasing challenges posed by big data environments on forensic investigations, the prevalent use of Hadoop and its derivatives (like Apache Spark) for BI tasks, and the need for proactive DFR measures. The review also highlighted the predominant use of Linux-based environments for Hadoop installations due to their stability and security.

As discussed above, previous research studies by [11,20] have highlighted the role of HDFS metadata in forensic investigations and the complexities involved in such analyses. This research builds on these findings to propose a holistic DFR framework and IR-script for Linux–Hadoop clusters. The framework aims to enhance the collection and preservation of forensic artefacts, ensuring organisations can effectively respond to security incidents and maintain the CIA of their data.

4. Research Methodology

The research methodology adopted for this research paper comprises two phases:

1. Prototype development.
2. Prototype evaluation.

A detailed discussion of the two phases is as follows:

4.1. Prototype Development

According to [24], different types of prototyping include rapid prototyping, slow, probing (throwaway), exploratory (proof of concept), developmental (live), incremental, and evolutionary. Each of these types of prototyping serves different purposes and approaches to developing and testing prototypes. The exploratory (proof of concept) method was selected for the development of this research prototype, followed by a functional and usability evaluation. This method was chosen because it allows for the preliminary demonstration of the feasibility and viability of a proposed idea, concept or technology. Proof of concept (POC) prototyping denotes the preliminary demonstration of the feasibility and viability of a proposed idea, concept, or technology. It involves developing a prototype to test key functionalities and validate assumptions before committing to full-scale development [24–26].

For this research, the focus was on developing and testing the IR-script, a critical tool for extracting and summarising key locations of digital forensics artefacts within a standard Linux–Hadoop environment to support the initial and final stages of a digital forensics investigation. A comprehensive network diagram of the proposed big data DFR framework and detailed configuration steps were also discussed in Section 5.

4.2. Prototype Testing and Evaluation

To evaluate the robustness and compatibility of the prototype, particularly its IR-script, functionality and usability testing were chosen to ensure that the IR-script performs its intended functions accurately and is user-friendly and efficient in practical scenarios. Testing was conducted within three virtual environments: AWS using Ubuntu Linux, Oracle VB using CentOS Linux, and HDP Sandbox using CentOS. This approach mirrors the autonomous functional and usability evaluation methods used by Thanekar et al. [20]. Employing these three environments not only facilitates a thorough evaluation of the IR-script's performance across varied platforms but also validates the autonomous functional and usability testing approach. This methodology enhances the reliability and generalisability of the findings by providing unbiased insights into the user experience. Furthermore, the source code of the IR-script will be made open-source to encourage further improvement by other researchers.

5. Proposed Framework

5.1. Requirements Analysis

To ensure that the system effectively meets the needs of digital forensics investigations, a comprehensive requirements analysis was conducted, guided by previous work [2,5,7–11,16,18,20]. This analysis was crucial to identify and define the required functionalities and features that the proposed system must possess to address the specific challenges and requirements of DFR in big data environments. Additionally, the analysis also considered four key factors discussed below:

- Architecture: Research on DFR and big data frameworks informed the identification of necessary system components and their functionalities.
- Operational needs: The practical needs of digital forensics investigations, such as data collection, secure storage, efficient processing, and access control, were key drivers in defining the system requirements.
- Technological capabilities: Leveraging existing technologies like Hadoop and Apache Spark ensured that the system could meet the performance and efficiency demands of both BI and forensics analytics.
- Compliance and security: Ensuring compliance with legal standards and maintaining robust security measures were paramount in defining the system requirements.

For the proposed system, seven key DFR requirements were identified and are outlined as follows:

1. Data Collection and Storage: The system must be capable of capturing and securely storing network access logs, firewall logs, IDS logs, SIEM logs, and Hadoop cluster data. It should centralise the storage of all relevant forensic data to facilitate easy management and analysis.
2. Network Segmentation and Security: Network segmentation and security of both the big data network and its DFR segment must be implemented.
3. Efficient Data Processing: The system must leverage technologies that effectively enforce both BI and forensics analytics, ensuring efficient processing and analysis of large data volumes.

4. Metadata Management: The system should provide efficient metadata management through secure centralised nodes, crucial for precise and efficient forensic analysis.
5. Access Control and Compliance: Strict access control measures must be enforced to ensure that only authorised personnel can interact with the forensic data. Furthermore, the system should log all security policies and maintain an up-to-date terms and conditions big data collection policy.
6. Time Synchronisation: Effective synchronisation with the network's network time protocol (NTP) server is essential to adhere to digital forensics admissibility requirements.
7. Resource Allocation: The forensic nodes must have sufficient storage and processing capabilities to handle the volume of data collected.

5.2. Framework Overview

The proposed big data DFR framework, illustrated in Figure 1, comprises essential components of a standard big data network, including a router, access points, a firewall, Security Information and Event Management (SIEM) systems, and Intrusion Detection Systems (IDS). Central to this framework is the Hadoop big data, DFR storage, and the processing environment. To enhance network security and enforce security policies, the framework includes VLAN (Virtual Local Area Network) switches. These switches improve security by logically isolating traffic and enabling the logging of segmented DFR data directly into the Hadoop cluster's forensic DFR nodes. This integration facilitates the efficient capture, aggregation, and analysis of large volumes of forensic data, thereby strengthening the detection, investigation, and response capabilities within the big data network.

The framework is depicted in Figure 1 below and its configuration is further discussed in Section 5.3.

This research paper aims to address the RQs raised in Section 2 (Research Focus):

RQ 1: How can DFR be effectively implemented on an existing Hadoop big data network without disrupting its operations?

RQ 2: Can digital forensics investigations within a Hadoop cluster be simplified for both experienced and inexperienced investigators, considering the various configurations of different Hadoop environments?

The research questions are answered in the subsequent Sections.

5.3. Research Question 1 Insights

To address RQ 1, the proposed framework recommends integrating DFR on the same network where a company's BI operations are already running. Specifically, it recommends adding dedicated forensic nodes to the existing Hadoop cluster.

The forensic nodes comprise DFR nodes (depending on the organisation's network) and a BI compliance node. The DFR nodes are configured to collect and securely store both record evidence and user/application evidential data [5]. This includes network and security logs from various sources, Hadoop cluster data, system event logs, database transaction logs, access control logs, application logs, audit trails, network traffic data, firewall logs, IDS logs, and SIEM data. This consolidated storage simplifies management and analysis while safeguarding sensitive information for potential investigations. It is essential to allocate sufficient storage and processing resources to accommodate the data volume. A backup of the DFR nodes in the cloud is also implemented.

The BI compliance node handles logging related to security policy management and compliance within the big data environment. It tracks current and historical security policies, including those for data anonymisation, encryption, and password management. This node provides a thorough record of security measures for auditing and compliance purposes and supports forensic investigations. Additionally, it maintains an updated policy on big data collection and processing, ensuring adherence to legal and ethical standards and mitigating the risk of data breaches.

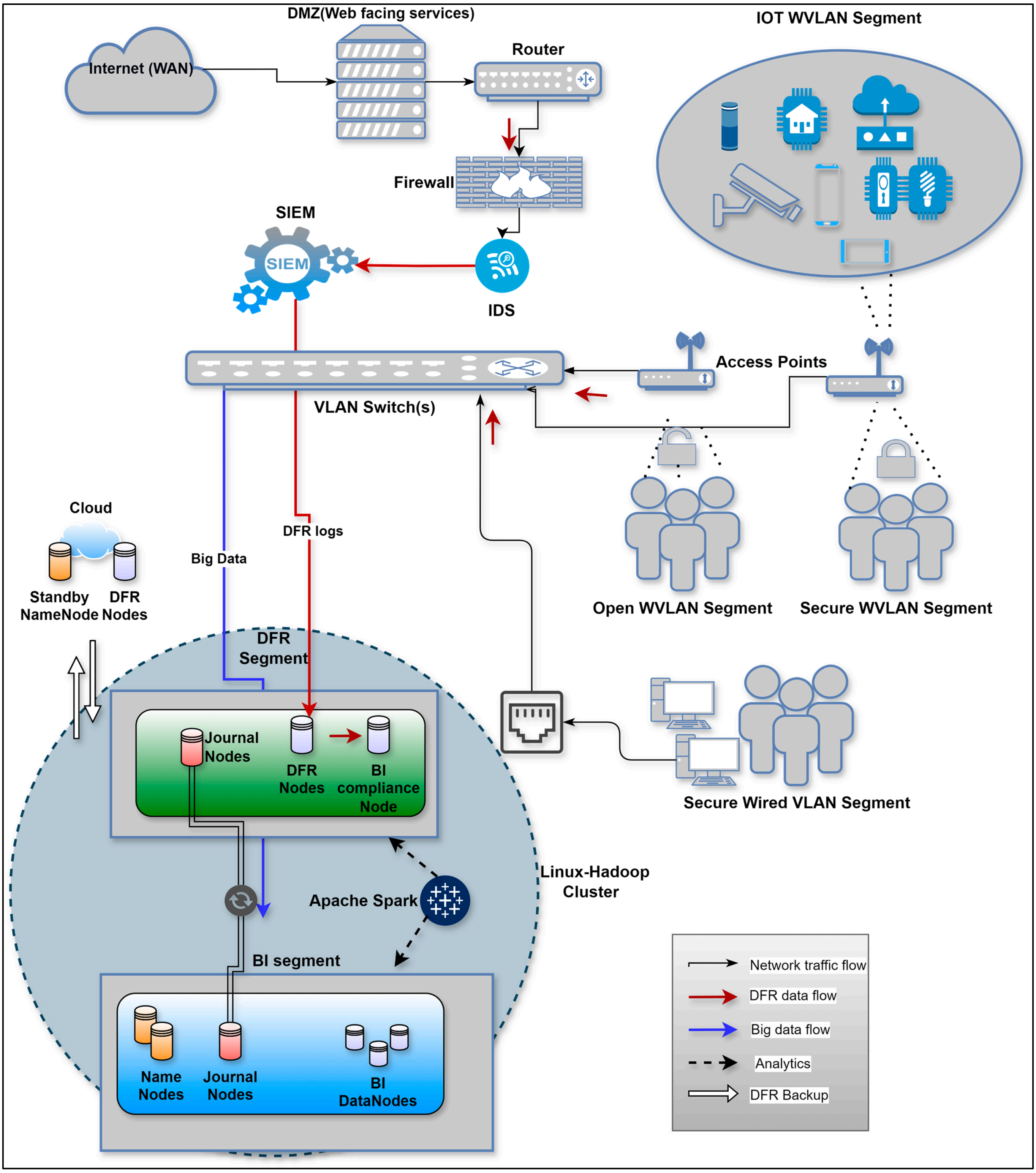


Figure 1. A Linux–Hadoop DFR framework for big data networks.

To add the new forensic nodes without disrupting the big data operations as depicted in Figure 1, the following steps are followed:

1. Begin by configuring a standby NameNode in the cloud to ensure redundancy and immediate backup capabilities.
2. Use VLANs or SDN (Software Defined Networking) to segment the network, creating separate segments for BI (BI segment) and DFR (DFR segment) operations. Ensure that

the forensic nodes are placed on their own segment, which can securely communicate with the BI segment's NameNode and the cloud-based standby NameNode.

3. Equip the forensic nodes with dedicated network interfaces that connect them directly to the BI segment. This ensures that forensic operations are prioritised within their dedicated segment and can communicate efficiently with the NameNode in the BI segment and the standby NameNode in the cloud.
4. Install Linux OS, Java, and Hadoop on the forensic nodes, set up SSH (Secure Shell) access, and update host resolution. Use automation tools like Ansible or Chef to standardise and automate the installation process, reducing configuration errors.
5. Synchronise all nodes (BI segment, DFR segment, and cloud-based standby NameNode) to a reliable NTP server to ensure consistent time settings across the cluster. This is crucial for accurate timestamping and coordination of distributed processes. Configure the NTP service on all nodes to point to the network's NTP server.
6. Implement an active–passive NameNode setup with the primary NameNode located in the BI segment and the standby NameNode in the cloud. Configure a quorum-based JournalNode cluster spanning both the BI and DFR segments. This setup ensures that all updates to the NameNode's metadata are immediately synchronised across all relevant nodes, maintaining consistency and reliability in both BI and forensic operations.
7. Set up a backup of the DFR nodes in the cloud to ensure that all forensic data are securely stored and can be recovered in case of a failure or disaster.
8. Start Hadoop services on the forensic nodes, such as DataNode and any additional services related to forensic readiness. Ensure these services are properly configured to interact with the primary NameNode in the BI segment and the cloud-based standby NameNode.
9. Conduct simulated failover tests to ensure that the forensic nodes maintain synchronisation with the BI segment's NameNode and the standby NameNode in the cloud during a failover. This verifies the resilience and reliability of the forensic setup.
10. Reconfigure Apache Spark to extend its capabilities from BI analytics to also support forensics analytics, ensuring that Spark can handle both BI and forensic data processing requirements effectively.
11. Implement end-to-end data integrity checks, including checksum validations and consistency checks across the HDFS. This step ensures that forensic data can be accurately retrieved and analysed without any discrepancies.
12. Implement MFA (multifactor authentication) for SSH access to the forensic nodes, adding an extra layer of security to prevent unauthorised access. This should be integrated into the network's existing security protocols.
13. Configure RBAC (Role-Based Access Control) to restrict access to forensic data and tools based on user roles, ensuring that only authorised personnel can access sensitive forensic data.
14. Set up a continuous monitoring solution using tools like Prometheus and Grafana to track the performance and health of the forensic nodes, as well as the BI segment. This ensures that any issues are quickly identified and resolved.
15. Finally, verify the integration by ensuring that the forensic nodes are recognised by the NameNode in the BI segment and can communicate efficiently with it and the cloud-based standby NameNode. Test data retrieval and forensic tool operations to confirm the setup functions as intended.

To optimise the implementation, a phased strategy is recommended for smooth integration. Firstly, the DFR node is set up to collect data from a subset of the network while monitoring and validating the overall configuration and performance. Gradual expansion to cover the entire network then ensues. Next, the establishment of the BI compliance node and subsequent migration of all relevant data and policies is conducted. Finally, the integration of BI and FI processes using Apache Spark to ensure seamless operation is performed.

Thorough testing and validation are essential. Functional testing ensures the new nodes operate correctly without negatively impacting the existing BI Hadoop cluster, while usability testing ensures the system meets the needs of both BI and FI teams. However, these tests are beyond the scope of this research. Lastly, adequate training programmes for IT and security personnel, alongside comprehensive documentation, will support the effective use and maintenance of the new nodes.

5.4. Research Question 2 Insights

To address RQ 2, an IR-script was developed. This script offers a comprehensive solution for streamlining digital forensics investigations when executed either on a Linux–Hadoop cluster's NameNode or an image of the NameNode. It is designed to be user-friendly for investigators of all experience levels, regardless of the diverse configurations of Linux–Hadoop environments. Developed using Bash shell scripting, standard Linux utilities, and Hadoop-specific commands, the script automates the retrieval and cataloguing of crucial supporting forensic information, making it easier to initiate investigations by abstracting the complexities of Hadoop environments. Additionally, the script is a valuable tool for network administrators and other IT personnel in their day-to-day tasks and troubleshooting, as will be demonstrated in its usability and functional evaluation.

The data artefacts extracted by the script include information about the Linux operating system, such as OS information, uptime, load averages, promiscuous mode status, potential rootkits, file system configurations, mount points, partitions, open ports, disc usage, and network settings. It also captures critical specifics of the Hadoop cluster, including the Hadoop version, configuration files, HDFS version, Hadoop log directories, environment variables, SSH configuration, DNS configuration, detailed cluster metrics, and active processes. Furthermore, the script is equipped with functionalities to detect installed IDS within the Linux–Hadoop environment and extract relevant log information. The script significantly enhances the efficiency and effectiveness of forensic investigations in complex Hadoop systems, proving invaluable in both the early and later stages of the investigation. By automating these tasks, the script enhances the investigation process, ensuring consistent and accurate data collection across diverse Hadoop configurations.

Moreover, the IR-script was developed in two versions: one that displays output in CSV format for further analysis in other tools, and another that produces well-titled documents and execution logs in text format. The text format improves readability and transparency by organising the retrieved information into accessible documents titled 'Hadoop_Info.txt', 'OS_Info.txt', and 'script_log.txt'. Alternatively, the CSV format produces 'Hadoop_Info.csv', 'OS_Info.csv', and 'script_log.csv' for further analysis. The choice between formats depends on the investigator's needs. These output formats provide investigators with well-documented supporting evidential information essential for the further analysis of the Linux–Hadoop environment. The configuration code for the IR-script is provided as part of the Supplementary Materials provided for this paper.

6. Prototype Configuration and Evaluation

6.1. Prototype Testbeds

As outlined in Section 1 (Introduction) and Section 4 (Research Methodology), due to resource and time constraints, setting up and evaluating a full-scale Hadoop big data environment was not feasible for this research. Therefore, the research focused on developing and assessing the effectiveness of a prototype of the IR-script. To evaluate its efficacy, functional and usability testing was conducted across three distinct Linux–Hadoop systems:

- AWS with Ubuntu Linux: This environment consisted of one NameNode, one secondary NameNode, and two DataNodes.
- Oracle VB with CentOS Linux: This setup included one NameNode and two DataNodes.

- Hortonworks Data Platform (HDP) Sandbox using CentOS: The testbed consisted of a single-node cluster running in a Docker container on a virtual machine. HDP comes with a suite of applications such as Apache Hive, Apache HBase, Apache Spark, and Apache Pig, providing a comprehensive platform for managing, processing, and analysing big data within a CentOS-based environment.

6.2. IR-Script Overview

The following discussion outlines the key features and functionalities of the IR-script.

- OS Information Collection: The script gathers detailed information about the operating system, including OS version, kernel details, users with sudo privileges, currently logged-in users, system uptime, load averages, disc usage, partition information, network configuration, and Linux log files. These comprehensive data are essential for understanding the system's state and configuration.
- Hadoop Cluster Information: This collects specific details about the Hadoop cluster, such as Hadoop configuration files, the HDFS version, Namenodes, secondary Namenodes, Hadoop log files, cluster environment variables, detailed Hadoop metrics, running Hadoop and Java processes, and information about cluster nodes. This ensures thorough coverage of critical Hadoop-specific information relevant to forensic analysis.
- Intrusion Detection Systems (IDS): The script checks the status and logs of various IDS tools, including Snort, Suricata, and OSSEC, if they are present on the system. This provides an overview of the security monitoring tools and their current operational status.
- Rootkit Detection: The script incorporates rootkit detection capabilities using tools such as chkrootkit, rkhunter, and unhide. These scans help in identifying potential rootkit infections, which is crucial for security investigations.

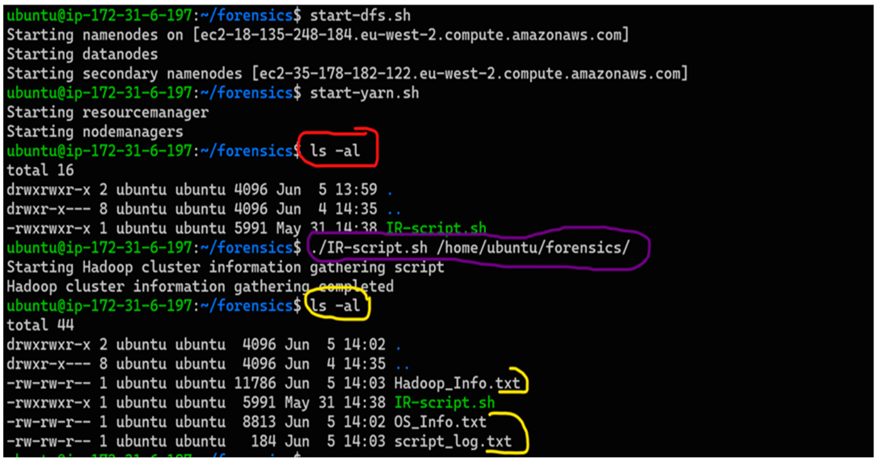
IR-script's Structure and Usage:
Functions:

- usage(): Displays usage instructions for the script.
- handle_error(): Manages errors by displaying an error message and exiting the script.
- command_exists(): Checks for the presence of required commands.
- get_os_info(): Retrieves and records OS-specific information.
- get_hadoop_info(): Gathers Hadoop cluster-related data.
- get_ids_info(): Provides information about installed IDS tools.
- get_rootkit_info(): Performs rootkit scans and records results.
- add_Section_header(), add_command_output(), add_csv_header(), and add_csv_row(): Functions to format and add data to output files, either as text or CSV.

The script produces output files in both text (OS_Info.txt, Hadoop_Info.txt,) and CSV (OS_Info.csv, Hadoop_Info.csv) formats. The choice of format can be selected based on user preference or requirements. All of the script outputs are logged in script_log.txt for comprehensive documentation.

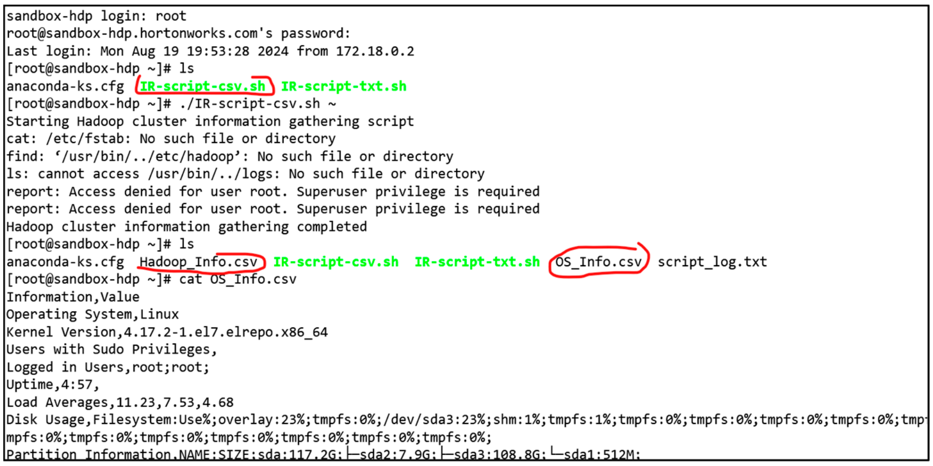
Users must specify the output directory as an argument when running the script. For example: ./hadoop-forensics.sh /path/to/output_directory. The script then performs a systematic collection of OS information, Hadoop cluster details, IDS statuses, and rootkit detection results, concluding with a completion message.

Figures 2 and 3 provide screenshots demonstrating the execution steps and results of the IR-script on an AWS Ubuntu Linux–Hadoop cluster for text document outputs and HDP sandbox for CSV document outputs.



```
ubuntu@ip-172-31-6-197:~/forensics$ start-dfs.sh
Starting namenodes on [ec2-18-135-248-184.eu-west-2.compute.amazonaws.com]
Starting datanodes
Starting secondary namenodes [ec2-35-178-182-122.eu-west-2.compute.amazonaws.com]
ubuntu@ip-172-31-6-197:~/forensics$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
ubuntu@ip-172-31-6-197:~/forensics$ ls -al
total 16
drwxrwxr-x 2 ubuntu ubuntu 4096 Jun  5 13:59 .
drwxr-x--- 8 ubuntu ubuntu 4096 Jun  4 14:35 ..
-rwxrwxr-x 1 ubuntu ubuntu 5991 May 31 14:38 IR-script.sh
ubuntu@ip-172-31-6-197:~/forensics$ ./IR-script.sh /home/ubuntu/forensics/
Starting Hadoop cluster information gathering script
Hadoop cluster information gathering completed
ubuntu@ip-172-31-6-197:~/forensics$ ls -al
total 44
drwxrwxr-x 2 ubuntu ubuntu  4096 Jun  5 14:02 .
drwxr-x--- 8 ubuntu ubuntu  4096 Jun  4 14:35 ..
-rw-rw-r-- 1 ubuntu ubuntu 11786 Jun  5 14:03 Hadoop_Info.txt
-rwxrwxr-x 1 ubuntu ubuntu  5991 May 31 14:38 IR-script.sh
-rw-rw-r-- 1 ubuntu ubuntu  8813 Jun  5 14:02 OS_Info.txt
-rw-rw-r-- 1 ubuntu ubuntu   184 Jun  5 14:03 script_log.txt
```

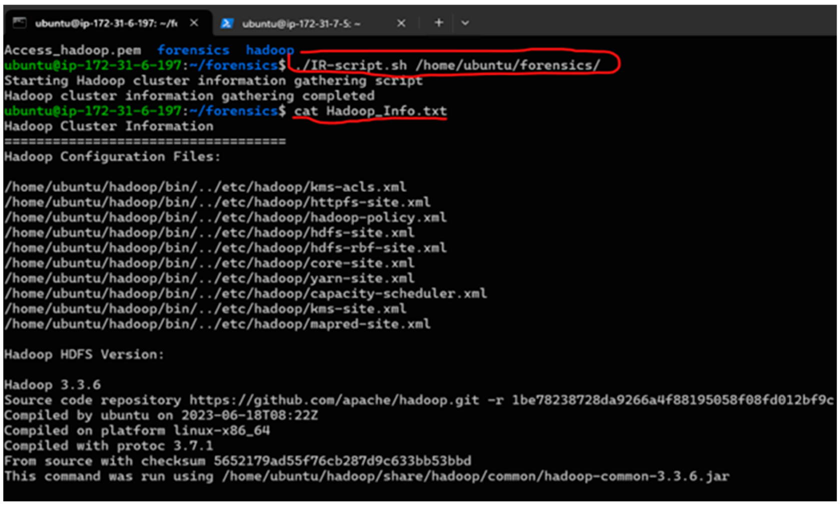
Figure 2. AWS IR-script execution and text files output.



```
sandbox-hdp login: root
root@sandbox-hdp.hortonworks.com's password:
Last login: Mon Aug 19 19:53:28 2024 from 172.18.0.2
[root@sandbox-hdp ~]# ls
anaconda-ks.cfg  IR-script-csv.sh  IR-script-txt.sh
[root@sandbox-hdp ~]# ./IR-script-csv.sh ~
Starting Hadoop cluster information gathering script
cat: /etc/fstab: No such file or directory
find: '/usr/bin/../etc/hadoop': No such file or directory
ls: cannot access /usr/bin/../logs: No such file or directory
report: Access denied for user root. Superuser privilege is required
report: Access denied for user root. Superuser privilege is required
Hadoop cluster information gathering completed
[root@sandbox-hdp ~]# ls
anaconda-ks.cfg  Hadoop_Info.csv  IR-script-csv.sh  IR-script-txt.sh  OS_Info.csv  script_log.txt
[root@sandbox-hdp ~]# cat OS_Info.csv
Information,Value
Operating System,Linux
Kernel Version,4.17.2-1.el7.elrepo.x86_64
Users with Sudo Privileges,
Logged in Users,root;root;
Uptime,4:57,
Load Averages,11.23,7.53,4.68
Disk Usage,Filesystem:Use%;overlay:23%;tmpfs:0%;/dev/sda3:23%;shm:1%;tmpfs:1%;tmpfs:0%;tmpfs:0%;tmpfs:0%;tmpfs:0%;tmp
mpfs:0%;tmpfs:0%;tmpfs:0%;tmpfs:0%;tmpfs:0%;tmpfs:0%;tmpfs:0%;
Partition Information,NAME:SIZE:sda:117.2G:├sda2:7.9G:├sda3:108.8G:└sda1:512M:
```

Figure 3. HDP sandbox IR-script execution and CSV files output.

Figure 4 below illustrates a partial output of the Hadoop_Info.txt file displayed on the AWS testbed using the cat command.



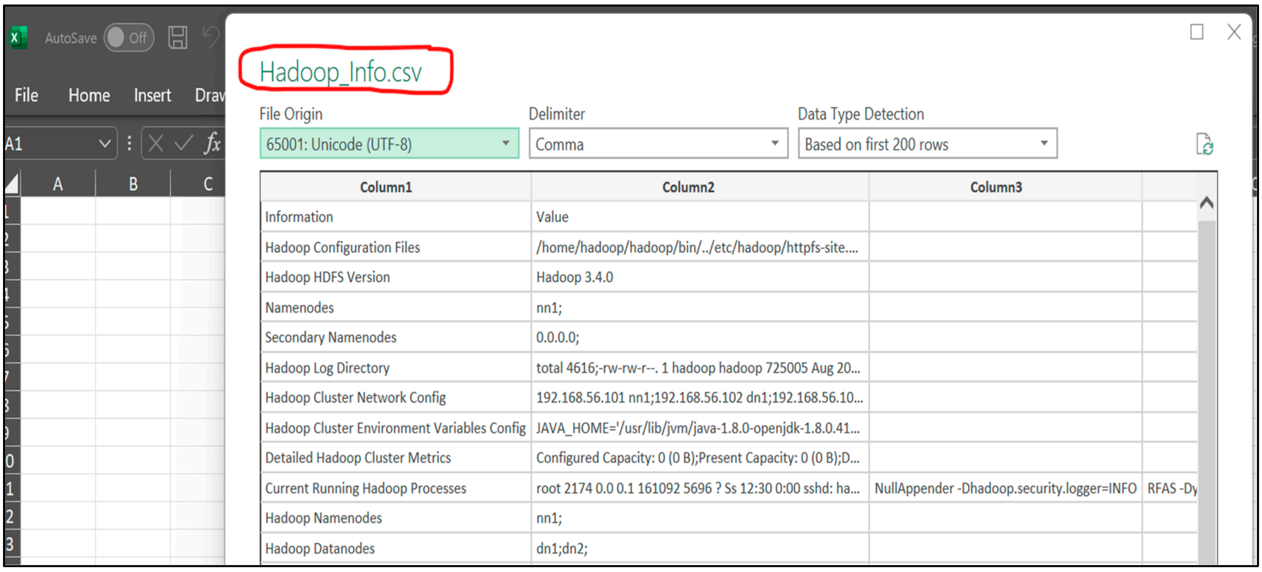
```
Access_hadoop.pem  forensics  hadoop
ubuntu@ip-172-31-6-197:~/forensics$ ./IR-script.sh /home/ubuntu/forensics/
Starting Hadoop cluster information gathering script
Hadoop cluster information gathering completed
ubuntu@ip-172-31-6-197:~/forensics$ cat Hadoop_Info.txt
Hadoop Cluster Information
===================================
Hadoop Configuration Files:

/home/ubuntu/hadoop/bin/../etc/hadoop/kms-acls.xml
/home/ubuntu/hadoop/bin/../etc/hadoop/httpfs-site.xml
/home/ubuntu/hadoop/bin/../etc/hadoop/hadoop-policy.xml
/home/ubuntu/hadoop/bin/../etc/hadoop/hdfs-site.xml
/home/ubuntu/hadoop/bin/../etc/hadoop/hdfs-rbf-site.xml
/home/ubuntu/hadoop/bin/../etc/hadoop/core-site.xml
/home/ubuntu/hadoop/bin/../etc/hadoop/yarn-site.xml
/home/ubuntu/hadoop/bin/../etc/hadoop/capacity-scheduler.xml
/home/ubuntu/hadoop/bin/../etc/hadoop/kms-site.xml
/home/ubuntu/hadoop/bin/../etc/hadoop/mapred-site.xml

Hadoop HDFS Version:

Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /home/ubuntu/hadoop/share/hadoop/common/hadoop-common-3.3.6.jar
```

Figure 4. Hadoop_Info.txt content from the AWS testbed.

6.3. IR-Script Evaluation

The evaluation of the IR-script was carried out using functional and usability approaches.

Functionality in this context refers to the ability of the script to perform its intended tasks accurately and efficiently. Functionality evaluation was conducted to ensure that the script successfully collected and organised the necessary supporting forensics artefacts. A key objective was to evaluate the script's effectiveness across various Linux–Hadoop configurations, focusing on its ability to collect detailed OS and Hadoop-specific information in both TXT and CSV formats. Figure 5 below shows a partial screenshot of the importation of the Hadoop_Info.csv file generated from the Oracle VB testbed into Microsoft Excel.



Hadoop_Info.csv

File Origin: 65001: Unicode (UTF-8)

Delimiter: Comma

Data Type Detection: Based on first 200 rows

Column1	Column2	Column3	
Information	Value		
Hadoop Configuration Files	/home/hadoop/hadoop/bin/../etc/hadoop/httpfs-site....		
Hadoop HDFS Version	Hadoop 3.4.0		
Namenodes	nn1;		
Secondary Namenodes	0.0.0.0;		
Hadoop Log Directory	total 4616;-rw-rw-r--. 1 hadoop hadoop 725005 Aug 20...		
Hadoop Cluster Network Config	192.168.56.101 nn1;192.168.56.102 dn1;192.168.56.10...		
Hadoop Cluster Environment Variables Config	JAVA_HOME='/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.41...		
Detailed Hadoop Cluster Metrics	Configured Capacity: 0 (0 B);Present Capacity: 0 (0 B);D...		
Current Running Hadoop Processes	root 2174 0.0 0.1 161092 5696 ? Ss 12:30 0:00 sshd: ha...	NullAppender -Dhadoop.security.logger=INFO	RFAS -Dy
Hadoop Namenodes	nn1;		
Hadoop Datanodes	dn1;dn2;		

Figure 5. Importing Hadoop_Info.csv data from Oracle VB testbed into Excel.

The functionality evaluation confirmed that the IR-script met 90.5% of the expected outcomes in both environments, demonstrating its robustness and reliability across different setups. During the prototype evaluation, a total of 63 artefacts were assessed for retrieval success across different testbed environments. The results are summarised as follows:

- Successful Retrieval: In total, 57 out of 63 artefacts were fully retrieved, indicating successful extraction and validation across the testbeds (marked as "Yes").
- Partial Retrieval: Three artefacts were only partially retrieved, reflecting incomplete or partial extraction (marked as "Partial").
- Failed Retrieval: Three artefacts could not be retrieved at all, denoting a failure in the extraction process (marked as "No").

The percentage of expected outcomes successfully met by the IR-script was calculated as follows:

$$\text{Percentage met} = (57/63) \times 100 = 90.5\%$$

Testbed Analysis:

1. HDP Sandbox Testbed: Out of the three artefacts that were not retrieved ("No"), all were associated with the HDP sandbox. Additionally, one artefact was only partially retrieved on this testbed.
2. Oracle VB Testbed: One artefact experienced partial retrieval in the Oracle VB environment.
3. AWS Testbed: Similarly, one artefact was partially retrieved in the AWS testbed environment.

These results highlight the relative reliability of the prototype across different test environments, with a particular focus on the HDP sandbox which exhibited the highest number of retrieval issues, as shown in Figure 3. Table 1 below summarises the results of the functional evaluation.

Table 1. Functional evaluation results.

IR-Script Generated Files	Forensics Artefacts	AWS	Oracle VB	HDP Sandbox	Results Summary
OS_Info.txt OS_Info.csv	1. Operating System Details 2. Users with Sudo Privileges 3. Logged-In Users 4. Uptime and Load Averages 5. Disc Usage and Partition Information 6. Mount Points and Filesystem Config 7. Network Configuration 8. Default Linux Log File List 9. Promiscuous Mode Check 10. Detect IDS and Check Logs 11. Check for Rootkits	1. Yes 2. Yes 3. Yes 4. Yes 5. Yes 6. Yes 7. Yes 8. Yes 9. Yes 10. Partial 11. Yes	1. Yes 2. Yes 3. Yes 4. Yes 5. Yes 6. Yes 7. Yes 8. Yes 9. Yes 10. Partial 11. Yes	1. Yes 2. Yes 3. Yes 4. Yes 5. Yes 6. Yes 7. Yes 8. Yes 9. Yes 10. No 11. Yes	In total, 10/11 instances of the supporting OS information in both the txt and CSV formats were retrieved for all three testbeds. More details in conclusion Section.
Hadoop_Info.txt Hadoop_Info.csv	1. Hadoop Configuration Files 2. HDFS Version 3. NameNodes and Secondary NameNodes 4. Default Hadoop Log File List 5. Hadoop Cluster Network Config 6. Cluster Environment Variables Config 7. Detailed Hadoop Cluster Info/Metrics 8. Current Running Hadoop Processes 9. Hadoop Cluster Nodes	1. Yes 2. Yes 3. Yes 4. Yes 5. Yes 6. Yes 7. Yes 8. Yes 9. Yes	1. Yes 2. Yes 3. Yes 4. Yes 5. Yes 6. Yes 7. Yes 8. Yes 9. Yes	1. No 2. Yes 3. Yes 4. No 5. Yes 6. Yes 7. Yes 8. Yes 9. Partial	All of the artefacts were retrieved for AWS and Oracle VB. However, 7/9 were retrieved for HDP sandbox.
Script_Log.txt	Time-Stamped Script Operations Account	Yes	Yes	Yes	Time-stamped log files for all testbeds were generated.

Usability pertains to the ease with which users (investigators) can effectively utilise the script to achieve their goals. The usability evaluation of the IR-script was performed to assess how user-friendly the script is for digital forensic investigators and network administrators. The evaluation revealed that while the script is thorough in its data collection, it can only be executed via the Command Line Interface (CLI). This limitation may pose a challenge for users who are not familiar with CLI operations or who prefer graphical interfaces. Despite this, the script performed fairly well, as it organised the collected data into clear and accessible text documents.

However, enhancing the script with a more user-friendly interface or additional guidance could improve its usability, making it more accessible to a broader range of users, including those with less technical expertise. Documents have been attached showing the usability steps and functional results of IR-script execution within the three testbeds.

7. Conclusions

In this study, the proposed framework and IR-Script effectively address the challenge of gathering essential supporting evidential information for digital forensics investigations in Linux–Hadoop environments. Traditionally, this process is manual, time-consuming, and prone to errors, requiring significant expertise in both Linux and Hadoop systems.

The IR-Script was evaluated across three different Linux–Hadoop environments: AWS with Ubuntu Linux, Oracle VB with CentOS, and HDP sandbox with CentOS. It successfully collected and organised a comprehensive system and Hadoop-specific information in the

AWS and Oracle VB environments. However, in the HDP sandbox, the script encountered difficulties retrieving Hadoop configuration files, Hadoop log directories, and Hadoop data nodes. These issues were primarily related to error messages about superuser privileges, suggesting that the test environment's configuration or privilege management may have contributed to these retrieval failures.

Despite these limitations, the script's thorough documentation and well-organised output, combined with its CLI-based design, significantly enhance its usability. The IDS check results were inconclusive, as no IDS were installed in the test environments, which was outside the scope of this research.

Future work will focus on the following:

- Extended Testing: A prototype and evaluation of the proposed conceptual DFR framework for big data networks. Additionally, the further evaluation of IR-script on additional Linux distributions to ensure its robustness and adaptability across a wider range of environments.
- AI Integration: Incorporating AI to enhance the script's capabilities, potentially improving its adaptability and accuracy in diverse scenarios.
- User-Friendly Interface: The development of a Python-based graphical user interface to make the script more accessible and user-friendly for a broader range of investigators and administrators.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/asi7050090/s1, IR_script_code: IR-script-csv-code.docx and IR-script-txt-code.docx: IR-script source code.

Author Contributions: Conceptualisation, C.M., C.G. and G.M.; methodology, C.M. and C.G.; software, C.M.; validation, C.M.; investigation, C.M.; resources, C.M.; data curation, C.M.; writing—original draft preparation, C.M.; writing—review and editing, C.M., C.G. and G.M.; visualisation, C.M., C.G. and G.M.; supervision, C.G. and G.M.; project administration, C.G. and G.M.; funding acquisition, C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding and was self-funded.

Data Availability Statement: Data are available upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ahmed, H.; Ismail, M.A.; Hyder, M.F. Performance optimization of hadoop cluster using linux services. In Proceedings of the 17th IEEE International Multi Topic Conference 2014, Karachi, Pakistan, 8–10 December 2014; IEEE: New York City, NY, USA, 2014; pp. 167–172.
2. Asim, M.; McKinnel, D.R.; Dehghantanha, A.; Parizi, R.M.; Hammoudeh, M.; Epiphaniou, G. Big data forensics: Hadoop distributed file systems as a case study. In *Handbook of Big Data and IoT Security*; Springer: Cham, Switzerland, 2019; pp. 179–210.
3. Taylor, R.C. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. *BMC Bioinform.* **2010**, *11*, S1. [CrossRef] [PubMed]
4. Singh, D.; Reddy, C.K. A survey on platforms for big data analytics. *J. Big Data* **2015**, *2*, 8. [CrossRef] [PubMed]
5. Sremack, J. *Big Data Forensics—Learning Hadoop Investigations: Perform Forensic Investigations on Hadoop Clusters with Cutting-Edge Tools and Techniques*; Mumbai Packt Publishing: Birmingham, UK, 2015.
6. Russom, P. Big data analytics. *TDWI Best Pract. Rep. Fourth Quart.* **2011**, *19*, 1–34.
7. Thakur, M. Cyber security threats and countermeasures in digital age. *J. Appl. Sci. Educ. (JASE)* **2024**, *4*, 1–20.
8. Sarker, I.H. *AI-Driven Cybersecurity and Threat Intelligence: Cyber Automation, Intelligent Decision-Making and Explainability*; Springer Nature: Berlin/Heidelberg, Germany, 2024.
9. Beloume, A. The Problems of Internet Privacy and Big Tech Companies. The Science Survey. 2023. Available online: https://thesciencesurvey.com/news/2023/02/28/the-problems-of-internet-privacy-and-big-tech-companies/ (accessed on 12 March 2024).
10. Olabanji, s.o.; Oladoyinbo, O.B.; Asonze, C.U.; Oladoyinbo, T.O.; Ajayi, S.A.; Olaniyi, O.O. Effect of Adopting AI to Explore Big Data on Personally Identifiable Information (PII) for Financial and Economic Data Transformation. 2024. Available online: https://ssrn.com/abstract=4739227 (accessed on 3 March 2024).
11. Harshany, E.; Benton, R.; Bourrie, D.; Glisson, W. Big Data Forensics: Hadoop 3.2.0 Reconstruction. *Forensic Sci. Int. Digit. Investig.* **2020**, *32*, 300909. [CrossRef]

12. Akinbi, A.O. Digital forensics challenges and readiness for 6G Internet of Things (IoT) networks. *Wiley Interdiscip. Rev. Forensic Sci.* **2023**, *5*, e1496. [CrossRef]
13. Shoderu, G.; Baror, S.; Venter, H. A Privacy-Compliant Process for Digital Forensics Readiness. *Int. Conf. Cyber Warf. Secur.* **2024**, *19*, 337–347. [CrossRef]
14. Elgendy, N.; Elragal, A. Big data analytics: A literature review paper. In *Advances in Data Mining. Applications and Theoretical Aspects, Proceedings of the 14th Industrial Conference, ICDM 2014, St. Petersburg, Russia, 16–20 July 2014*; Proceedings 14; Springer International Publishing: Berlin/Heidelberg, Germany, 2014; pp. 214–227.
15. Kumar, Y.; Kumar, V. A Systematic Review on Intrusion Detection System in Wireless Networks: Variants, Attacks, and Applications. *Wirel. Pers. Commun.* **2023**, *133*, 395–452. [CrossRef]
16. Mpungu, C.; George, C.; Mapp, G. Developing a novel digital forensics readiness framework for wireless medical networks using specialised logging. In *Cybersecurity in the Age of Smart Societies, Proceedings of the 14th International Conference on Global Security, Safety and Sustainability, London, 7-8 September 2022*; Springer International Publishing: Cham, Switzerland, 2023; pp. 203–226.
17. Yaman, O.; Ayav, T.; Erten, Y.M. A Lightweight Self-Organized Friendly Jamming. *Int. J. Inf. Secur. Sci.* **2023**, *12*, 13–20. [CrossRef]
18. Sachowski, J. *Implementing Digital Forensic Readiness: From Reactive to Proactive Process*, 2nd ed.; CRC Press: Boca Raton, FL, USA; Taylor & Francis Group: Abingdon On Thames, UK, 2019.
19. Oo, M.N. Forensic Investigation on Hadoop Big Data Platform. Ph.D. Thesis, University of Computer Studies, Yangon, Myanmar, 2019.
20. Thanekar, S.A.; Subrahmanyam, K.; Bagwan, A.B. A study on digital forensics in Hadoop. *Indones. J. Electr. Eng. Comput. Sci.* **2016**, *4*, 473–478. [CrossRef]
21. Joshi, P. Analyzing big data tools and deployment platforms. *Int. J. Multidiscip. Approach Stud.* **2015**, *2*, 45–56.
22. Messier, R.; Jang, M. *Security Strategies in Linux Platforms and Applications*; Jones & Bartlett Learning: Burlington, MA, USA, 2022.
23. Nazeer, S.; Bahadur, F.; Iqbal, A.; Ashraf, G.; Hussain, S. A Comparison of Window 8 and Linux Operating System (Android) Security for Mobile Computing. *Int. J. Comput. (IJC)* **2015**, *17*, 21–29.
24. Evaluating Prototypes. Available online: https://www.tamarackcommunity.ca/hubfs/Resources/Tools/Aid4Action%20Evaluating%20Prototypes%20Mark%20Cabaj.pdf (accessed on 20 May 2024).
25. Häggman, A.; Honda, T.; Yang, M.C. The influence of timing in exploratory prototyping and other activities in design projects. In Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, OR, USA, 4–7 August 2013; American Society of Mechanical Engineers: New York, NY, USA, 2013; Volume 55928, p. V005T06A023.
26. Sadia, H. 10 Prototype Testing Questions a Well-Experienced Designer Need to Ask. Webful Creations. 2022. Available online: https://www.webfulcreations.com/10-prototype-testing-questions-a-well-experienced-designer-need-to-ask/ (accessed on 18 February 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.