Forensic Investigation in Robots

T. Janarthanan and S.Zargari,
“Forensic Investigation in Robots”,
Latin-American Journal of Computing (LAJC), vol. 11, no. 2, 2024.

ARTICLE HISTORY
Received 08 March 2024
Accepted 05 May 2024
Published 08 July 2024

Tharmini Janarthanan
Sheffield Hallam University
Sheffield, United Kingdom
tharmini.janarthanan@shu.ac.uk
ORCID: 0009-0008-9047-8556

Shahrzad Zargari
Sheffield Hallam University
Sheffield, United Kingdom
s.zargari@shu.ac.uk
ORCID: 0000-0001-6511-7646

Forensic Investigation in Robots

Tharmini Janarthanan
Sheffield Hallam University
Department of Computing
Sheffield, United Kingdom
tharmini.janarthanan@shu.ac.uk

Shahrzad Zargari
Sheffield Hallam University
Department of Computing
Sheffield, United Kingdom
s.zargari@shu.ac.uk

Abstract**—**Integrating robots into industrial automation has led to a revolutionary transformation in executing complex tasks, harnessing precision and efficiency. The Robot Operating System (ROS) has played a significant role in driving this advancement. ROS Bag files in robots are crucial for preserving data, as they provide a format for recording and playing back ROS message data. These files serve as a comprehensive log of a robot's sensory inputs and operational activities, enabling detailed analysis and reconstruction of the robot's interactions and performance over time. However, there have been instances where security considerations were overlooked, giving rise to concerns about unauthorized access, data theft, and malicious actions. This research investigates the forensic potential of data generated by robots, with a particular focus on ROS Bag data. By analyzing ROS Bag data, we aim to uncover how such information can be used in forensic investigations to reconstruct events, diagnose system failures, and verify compliance with operational protocols. The components of the ROS ecosystem were examined, identifying the challenges in parsing ROS Bag files and underscoring the need for specialized tools. This analysis highlights the security risks associated with plain text communication within legacy ROS systems, emphasizing the importance of encryption. While providing valuable insights, this research calls for further exploration, tool development, and enhanced security practices in robotics and digital forensics, aiming to lay the foundation for effective crime resolution involving robots.

Keywords**—***Robot forensics, forensics, ROS, Cybersecurity*

I. Introduction

In recent years, researchers have been increasingly focused on enhancing industrial automation processes by integrating advanced robotic technologies. A key aspect to this progress is the Robot Operating System (ROS), which significantly boosts the speed and capabilities of robots, positioning them as indispensable assets to the industry. Particularly, robots play an indispensable role in critical sectors such as healthcare as they significantly contribute to patient care, medication administration, and surgical procedures. In fact, the data generated by robots holds valuable insights that assist medical professionals in making informed decisions, ultimately improving patient outcomes [1]. However, amidst the pursuit of these advancements, security considerations have frequently been overlooked, which can expose significant vulnerabilities that malicious actors might exploit [2]. For instance, unauthorized access to robot control systems, sensors, and data poses a substantial threat, potentially allowing adversaries to seize control, manipulate actions, or steal sensitive information. Moreover, inadequate security protocols may lead to data breaches, not only endangering the privacy of collected information, but also the intellectual property of the robotic system's critical functionality. Also, insecure communication channels could enable eavesdropping due to the lack of trust mechanisms which might allow unauthorized modifications to ROS software/firmware. Such insecure practices can lead to physical tampering, and unpredictable robot behavior with severe legal and reputational repercussions [3].

Since the robots' interaction with machinery can generate important digital traces, valuable insights can be retrieved and analyzed as potential evidence in forensic investigations. Particularly, analyzing *ROS bag data* can uncover significant artefacts, retrace robot movements, and reconstruct events [4], using traces produced by diverse sensor inputs like visual, audio, and environmental data [5]. In this study, we propose two primary objectives to enhance the investigation of cybercrimes involving robots: Firstly, developing an understanding of the Robot Operating System (ROS) and its underlying structure. Secondly, exploring the potential forensic artefacts that can be extracted from a ROS via scenario-based simulations.

The rest of the paper is organized as follows: Section II presents a literature review on ROS communication process, addressing its inherent security challenges while exploring the emerging field of ROS forensics. In Section III, an overview of the research methodology is provided. In Section IV the experimental setup for ROS-based forensic evidence retrieval is described. Later, Section V discusses the findings and the results, highlighting the significance of the artefacts discovered. Finally, conclusions and directions for future work are outlined in Section VI.

II. Literature Review

In this section, we discuss the characteristics of the Robot Operating System, its communication process as well as its security challenges, and the emerging field of ROS Forensics.

A. Robot Operating System (ROS)

ROS is a framework for developing robotic software, offering an extensive suite of tools and libraries. Its powerful features enable developers to facilitate message passing, perform distributed computations, reuse code, and implement algorithms for various robotic applications. A key objective of ROS is to create a standardized programming approach for robots, providing off-the-shelf software components that can be seamlessly integrated into custom robotic projects. Today, ROS has become the preferred platform for many leading robotics companies. This shift is also evident in industrial robotics, where companies increasingly transition from proprietary robotic applications to ROS [5].

ROS manages multiple distributed functional entities known as *nodes*, each representing an autonomous process with its own lifecycle within an application context. Central to a ROS's architecture is a dedicated entity operating on a specific host within the ROS network also known as a *master* which is responsible for overseeing and mediating operations. The master maintains a directory of all existing nodes and their corresponding data [6]. At the architecture's core, there

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: https://doi.org/10.33333/lajc.vol11n2.03

is the *publish-subscribe communication model*, which main purpose is to effectively simplify complex components and establish precise interfaces for their connections. This model uses a *topic-based approach*, creating virtual channels (or topics) for individual instances. Thus, subscribers can use such topics to access the transmitted information. For example, in the ROS environment, a sensor node capturing images from a camera would publish this visual data on a specific topic, allowing any node requiring this information to subscribe to the relevant topic. Within the publish-subscribe framework, the specific identities of the publisher and the subscriber are relatively unimportant, facilitating seamless swapping within a ROS network. This feature also streamlines the addition of existing nodes or their adaptation for new applications.

B. ROS Communication

In a ROS environment, the entire communication process adheres to the publish-subscribe paradigm for each action-related topic [5] [7]. The master keeps a comprehensive catalogue of all available services. ROS also supports client-server communication through services, enabling a service client to request connection details for a specific service [5] [6]. Since ROS communication can flexibly use both TCP (ROSTCP) and UDP (ROSUDP) protocols, a service, identified by a unique name, can be accessed interactively and synchronously by a client, serving various purposes, such as obtaining one-time information. During its initialization, a publisher node contacts the master to declare the topics it plans to publish [4]. Then, a subscriber communicates its topic requirements to the master. When the master finds a compatible match between a publisher and a subscriber, it informs the subscriber about potential publishers for the designated topic. Subsequent communication between these nodes occurs directly, bypassing the ROS master.

For complex tasks, such as directing a mobile robot, ROS uses a communication pattern utilizing five topics. In this case, the process begins with the client sending a goal to the server, which in turn provides continuous updates and feedback through dedicated topics, including the robot's location. The outcome of the task is communicated via a result topic, while a cancel topic allows for the termination of the task.

C. Security Challenges in Robots

Industry experts have already noted that although manufacturers initially prioritized the physical safety of human operators, and their interaction with robotic systems, robot cybersecurity is currently critical due to their exposure to a broader range of vulnerabilities [8]. Likewise, according to ABI Research [9], the number of connected industrial robots will reach 4.3 million units by 2025, highlighting their expanding attack surface, making them increasingly prone to cyberattacks, physical tampering, and ethical issues. While becoming more interconnected, autonomous, and capable of managing critical tasks, it is clearer that robot widespread adoption has significantly increased the complexity of managing cybersecurity-related challenges, affecting both industries and individuals. Particularly, individuals without formal training may unintentionally introduce security risks [3] [8] while developing robot platforms, applications, hardware, and sensors. In fact, the landscape of robot cybersecurity is defined by many attack surfaces, including the physical robot, operating system, software or firmware, remote control technologies, vendor Internet services, cloud services, and networks [10].

Conversely, hackers may target robots for various impactful reasons, such as manipulating them to introduce defects in manufactured parts or assemblies; thereby sabotaging production processes. They may also use ransomware tactics to coerce manufacturers into paying substantial ransoms to prevent the exposure of compromised production lots. In some cases, hackers cause physical damage to the robots or robotic cells themselves, posing a direct threat to human workers. The stakes are further heightened by the theft of critical information, intellectual property, and data manipulation, leading to erroneous decision-making [3]. Moreover, robots are significantly vulnerable to cybersecurity concerns due to limitations such as the absence of proper authorization or authentication, encryption deficiencies, and insufficient physical protection measures [11].

In contrast, ROS, serving as the foundational infrastructure for numerous robots, could become a target in operating system attacks aimed at exploiting vulnerabilities. An in-depth analysis involving 176 threats from the robot vulnerability database revealed that 92.6 per cent are predominantly linked to software-related issues. This highlights the elevated threat level posed by software in robotics systems compared to hardware components [12]. Besides, in [13], researchers illustrate how genuine attacker profiles targeting ROS-based robots can be discerned, providing valuable insights for experts in selecting appropriate ROS security solutions for mitigating man-in-the-middle (MITM) attacks. Furthermore, vulnerabilities extend to multi-robot active surveillance systems, where intruders can manipulate or disable any ROS node within the network using shutdown commands. Another study highlights the potential for attackers to misguide robots by tampering with velocity commands [14]. ROS environments also face significant risks due to unsecured communication ports. In [15], ROS-based attacks involving plain-text communication over unsecured ports are illustrated, potentially leading to unauthorized access. Also, while [16] showcased the exploitation of APIs by attackers to undermine ROS applications, [17] emphasized the interception, manipulation and disruption of communication between two ROS nodes which may cause not only poor performance and disruption in operations, but also a potential disclosure of sensitive information.

Therefore, as robots become more appealing targets for hackers, the need to understand the ROS file structure and the locations of data artefacts is essential for conducting forensic investigations in this domain in order to reduce the impact of security breaches and prevent future incidents.

D. ROS Forensic Investigation

ROS Forensics is an emerging field that rapidly explores digital traces to extract valuable insights about security vulnerabilities present in robots. These insights serve various purposes, including investigating cybercrimes, uncovering malicious activities, and providing evidence for legal proceedings. Despite the increasing importance of ROS Forensics in the context of robotics and the Internet of Things, research in this area remains relatively undeveloped due to the lack of understanding of the ROS file structure for identifying artefacts within these systems for effective incident response and thorough forensics examinations. Compared to studies on

ROS security, research on ROS forensics is sparse. An early notable contribution to this domain is the work of [18], which was focused on assessing ROS cyber-physical security. The authors identified vulnerabilities and potential threats to ROS-based systems through various attacks and scenarios, laying the groundwork for digital investigations in this field.

Conversely, researchers in [19] introduced a framework aimed at aiding investigators in retrieving digital evidence from such systems. The authors noted that ROS presents investigative challenges, particularly in real-time scenarios, due to the complexities of communication between robot components and the ROS framework. These complexities introduce supplementary data during the investigation process, thereby complicating the gathering of accurate evidence from ROS. The complex communication between robot components and the ROS framework in real-time scenarios introduces additional data during investigations. A deep understanding of the ROS file structure enables investigators to identify and extract relevant data streams efficiently. This capability aids in extracting relevant information while eliminating unnecessary data noise.

On the other hand, in [4], researchers conducted a forensic examination of ROS, describing the tools required for ROS memory acquisition. The authors used three memory forensics tools: the *DD command*, *LiME*, and the *Volatility memory framework*. Their Kali Linux experiment involved executing fundamental breaches to acquire live memory data. The memory images collected with DD and LiME were verified using the Bdsum5 command. Subsequently, the analysis of memory images was compared with live response analysis. According to the report, live response activities disrupted ROS operational processes. Although the impact was less pronounced in the case of image forensics, it contributed to enhanced digital evidence integrity. Likewise, research carried out in [20], employed memory forensics in ROS using the Volatility tool. The authors utilized the *Linux_rosnode plugin*, facilitating memory investigation within ROS. Their study focused on collecting digital evidence from robot memory, specifically focusing on the cognition device, a part of ROS, to assess the possibility of tampering within the system. Unlike the study conducted in [20], this work presents a well-defined method and definitive outcomes.

As opposed to the previous approaches, [21] provided a comprehensive overview of ROS forensics, focusing on ROS 2 security features, security challenges, and vulnerabilities. This study revealed the difficulties distinguishing between a software bug and a deliberate attack within ROS systems. Such challenges arise from the goal of forensics investigations to identify potential attacks and their subsequent outcomes. The authors highlighted the research gap and the limited availability of in-depth ROS forensics investigation studies.

III. Methodology

To explore the challenges of forensics investigation in robotics domains, a simulated ROS environment was utilized to replicate real-life scenarios. Data was generated by engaging the robot in various movement activities. Although ROS 2 is the latest version of ROS and surpasses its predecessor, ROS 1, the latter has been used for over a decade and many robots still operate on ROS 1. Actually, ROS 1 is still supported by a substantial user community and a comprehensive repository of libraries, tools, and resources. Thus, in this study, ROS 1 was chosen due to the higher probability of having ROS 1 robots being targeted by potential attackers compared to their ROS 2 counterparts. After configuring the environment and recording robot activities to generate data, Autopsy and FTK Imager were employed to extract and analyze forensic artefacts from different locations. This aids in gaining a clearer understanding of the type of information and the location of valuable artefacts.

IV. Experimental Work

In this section, our proposed ROS experimental setup is explained.

A. ROS Environmental Setup

In this research, ROS was deployed on a Linux-based virtual machine to establish the laboratory environment. The machine operated on Windows 11, using VMware for virtualization. Within VMware, Ubuntu 20.04.06 (Focal Fossa) was installed as the guest operating system. The configuration included the installation of *ROS Noetic Ninjemys (ROS 1)* distribution systems on Ubuntu. Figure 1 illustrates the proposed experimental setup visually.

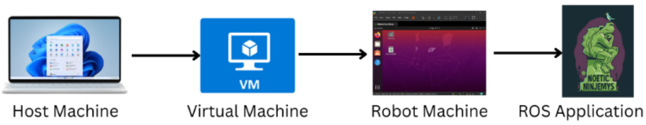
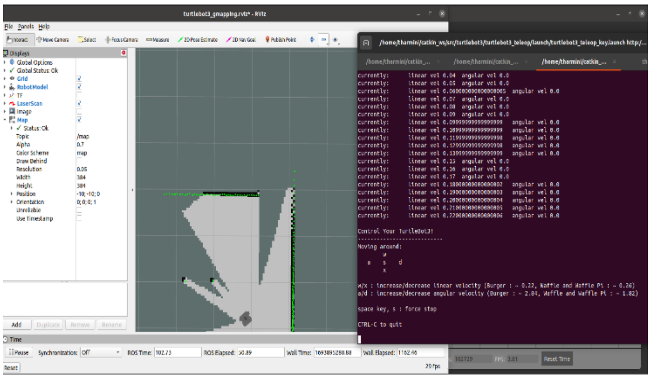


Fig. 1. Experimental setup of the ROS environment.

Additionally, the date and time settings on the Ubuntu virtual machine were adjusted to Pacific Daylight Time (Los Angeles, United States) to align with the default settings.

B. TurtleBot3 Experimentation and Data Generation

The next step involved setting up a simulated environment using the *TurtleBot3 model Waffle_Pi*. This simulation was designed to closely replicate the robot's behavior within a controlled virtual setting called *House World*. Additionally, the *teleoperation package* was employed to manually control the robot's movements within this environment.


turtlebot3_gmapping.rviz* - RViz
/home/tharmini/catkin_ws/src/turtlebot3/turtlebot3_teleop/launch/turtlebot3_teleop_key.launch http:/...
Reset Time

```
currently:	linear vel 0.04	 angular vel 0.0
currently:	linear vel 0.05	 angular vel 0.0
currently:	linear vel 0.060000000000000005	 angular vel 0.0
currently:	linear vel 0.07	 angular vel 0.0
currently:	linear vel 0.08	 angular vel 0.0
currently:	linear vel 0.09	 angular vel 0.0
currently:	linear vel 0.09999999999999999	 angular vel 0.0
currently:	linear vel 0.10999999999999999	 angular vel 0.0
currently:	linear vel 0.11999999999999998	 angular vel 0.0
currently:	linear vel 0.12999999999999998	 angular vel 0.0
currently:	linear vel 0.13999999999999999	 angular vel 0.0
currently:	linear vel 0.15	 angular vel 0.0
currently:	linear vel 0.16	 angular vel 0.0
currently:	linear vel 0.17	 angular vel 0.0
currently:	linear vel 0.18000000000000002	 angular vel 0.0
currently:	linear vel 0.19000000000000003	 angular vel 0.0
currently:	linear vel 0.20000000000000004	 angular vel 0.0
currently:	linear vel 0.21000000000000005	 angular vel 0.0
currently:	linear vel 0.22000000000000006	 angular vel 0.0

Control Your TurtleBot3!
---------------------------
Moving around:
        w
   a    s    d
        x

w/x : increase/decrease linear velocity (Burger : ~ 0.22, Waffle and Waffle Pi : ~ 0.26)
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle Pi : ~ 1.82)

space key, s : force stop

CTRL-C to quit
```

Fig. 2. Creating a SLAM map of the environment.

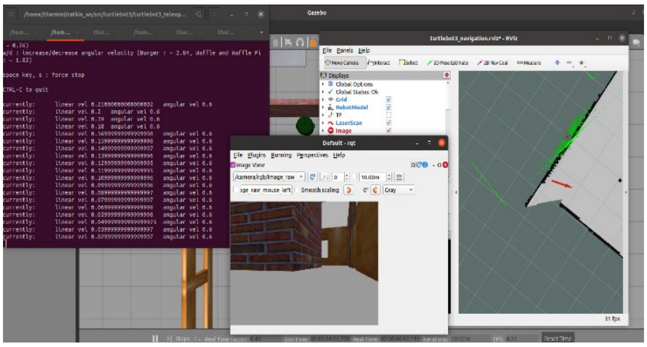
A key aspect of this experiment was the use of *Simultaneous Localization and Mapping (SLAM) algorithms*, which were instrumental in generating detailed maps of the environment while simultaneously tracking the robot's precise position (see Figure 2). This was mainly used to enable the robot to explore partially known environments using SLAM technology autonomously.

LATIN-AMERICAN JOURNAL OF COMPUTING (LAJC), Vol XI, Issue 2, July 2024

An interesting observation during the experiment was the robot's ability to detect and adapt to new obstacles, including other robots or human entities within the evolving map. This ability allowed the robot to autonomously and dynamically adjust its navigation. In the mapping process, walls or obstacles were depicted in black, unobstructed areas in white, and uncharted or ambiguous regions in varying shades of grey or transparency.

Rviz, the ROS visualization tool, was employed to acquire real-time insights into sensor data, mapping progress, and the robot's precise position. This tool proved invaluable in monitoring and assessing robot interactions with their environment. However, we encountered performance-related issues, leading to frustrating instances of lag and reduced responsiveness. These issues had tangible consequences, notably slowing down the robot's movements and introducing complexities into the mapping process.

By default, the ROS tool saves the map into the home directory in two formats (unless specified otherwise): (i) *the map.pgm file* and (ii) *the map.yaml file*. The file (i) visualized the environment, featuring distinct white, grey, and black regions. These regions denoted various aspects of the mapped space, with black indicating walls or obstacles, white representing unobstructed areas, and grey or transparent sections signifying uncertainty regions. On the other hand, the file (ii) held essential configuration data for the *map.pgm image* [22]. This configuration data provided critical information about the map's scale, resolution, and other parameters for effectively interpreting and utilizing the map. This usually benefits the developers or researchers for future navigation undertakings and analysis. To end the SLAM process, the terminal operating the SLAM node was shut down. This completed the mapping and location process. With this capability, we could make the most of the TurtleBot3 for various SLAM-connected purposes, like self-governing navigation, exploration, and automation. Also, cameras and sensors were incorporated into the simulation environment to enhance the dataset and introduce a visual dimension to data collection, as shown in Figure 3.



```
 - 0.26)
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle Pi
: ~ 1.82)

space key, s : force stop

CTRL-C to quit

currently:      linear vel 0.21000000000000002   angular vel 0.6
currently:      linear vel 0.2   angular vel 0.6
currently:      linear vel 0.19  angular vel 0.6
currently:      linear vel 0.18  angular vel 0.6
currently:      linear vel 0.16999999999999998   angular vel 0.6
currently:      linear vel 0.15999999999999998   angular vel 0.6
currently:      linear vel 0.14999999999999997   angular vel 0.6
currently:      linear vel 0.13999999999999996   angular vel 0.6
currently:      linear vel 0.12999999999999995   angular vel 0.6
currently:      linear vel 0.11999999999999995   angular vel 0.6
currently:      linear vel 0.10999999999999996   angular vel 0.6
currently:      linear vel 0.09999999999999996   angular vel 0.6
currently:      linear vel 0.08999999999999997   angular vel 0.6
currently:      linear vel 0.07999999999999997   angular vel 0.6
currently:      linear vel 0.06999999999999998   angular vel 0.6
currently:      linear vel 0.05999999999999998   angular vel 0.6
currently:      linear vel 0.049999999999999975  angular vel 0.6
currently:      linear vel 0.03999999999999997   angular vel 0.6
currently:      linear vel 0.02999999999999997   angular vel 0.6
```

/home/tharmini/catkin_ws/src/turtlebot3/turtlebot3_teleop...
Gazebo
turtlebot3_navigation.rviz* - RViz
Displays
Global Options
Global Status: Ok
Grid
RobotModel
tf
LaserScan
Image
Default - rqt
File Plugins Running Perspectives Help
Image View
/camera/rgb/image_raw
Smooth scaling
Gray
Steps: 1 Real Time Factor:

Fig. 3. Capturing of camera data by Turtlebot3.

This effort aimed to provide a more comprehensive understanding of the robot's interactions with its environment. The process of configuring and utilizing these sensors was simplified by the *rqt package* which offered user-friendly tools for setup. The cameras, acting as the robot's eyes, diligently recorded its visual observations while the author efficiently stored this data using ROS bags. These bags functioned as comprehensive repositories, capturing a wide array of sensor information and detailed accounts of the robot's movements. Moreover, the recorded data's adaptability was highlighted by the ease of playback and analysis, achievable through ROS commands or the convenient rqt tool (see Figure 4). This gave us a powerful means to revisit and scrutinize robot behaviors and sensor data.

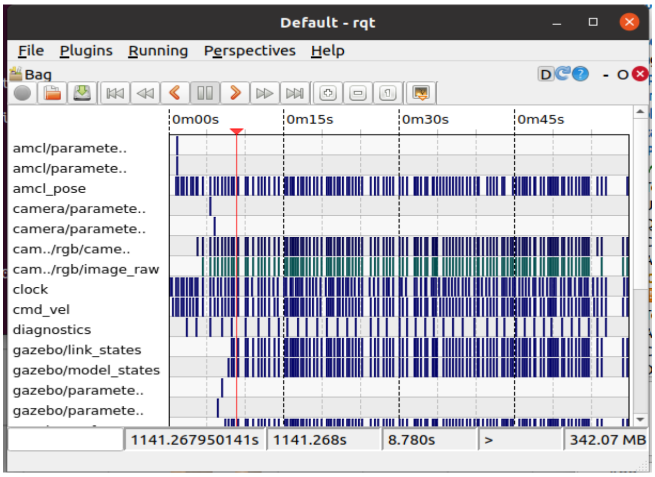


Fig. 4. An example of rqt bag.

Recording TurtleBot3 data using ROS bags, especially when incorporating SLAM, provides a comprehensive dataset that captures the robot's interactions with the environment. Additionally, it captures its self-estimated localization and mapping. This recorded data is crucial for robotic developers and researchers, facilitating in-depth analysis, algorithm development, and refinement of robot behavior.

However, several challenges were encountered. The simulation environment was resource-intensive, and limited knowledge hindered the ability to obtain extensive autonomous data. Issues with lagging and reduced performance were also faced. Moreover, we had to fine-tune Gazebo profile configurations using the command *gz physics -s* to balance performance and data quality [23]. Accurate calibration of sensors, such as cameras, and synchronization with robot movements required meticulous attention to detail. Researchers should be prepared to address these challenges to ensure the quality and reliability of recorded data.

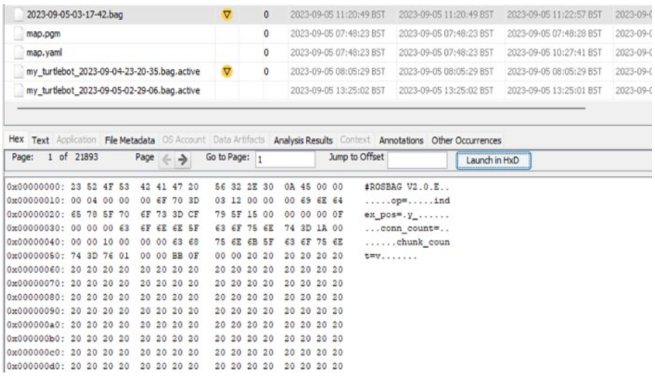
Despite these obstacles, our research provided valuable insights into the complexities of robot data collection. It also highlighted the importance of addressing performance issues to ensure the acquisition of high-quality autonomous data. We strongly emphasize the significance of having a deeper understanding of the ROS and Gazebo ecosystem as well as optimizing the virtual environment for efficient data generation and collection.

V. Findings and Discussion

In this section, findings regarding the analysis of the retrieved artefacts and their significance in robot forensics are discussed.

A. *ROS Bag Analysis*

An analysis using the Autopsy tool was conducted to examine the structure and contents of a *ROS Bag file* extracted from the virtual disk (.vmdk). The .vmdk image was loaded into Autopsy and FTK Imager for forensics analysis and verification. The findings from this analysis revealed crucial information about the internal organization and metadata of this file. The analysis primarily focused on hexadecimal lines extracted from the file which contained critical details regarding the file's structure and metadata (see Figure 5).



2023-09-05-03-17-42.bag	▽	0	2023-09-05 11:20:49 BST	2023-09-05 11:20:49 BST	2023-09-05 11:22:57 BST	2023-09-(
map.pgm		0	2023-09-05 07:48:23 BST	2023-09-05 07:48:23 BST	2023-09-05 07:48:28 BST	2023-09-(
map.yaml		0	2023-09-05 07:48:23 BST	2023-09-05 07:48:23 BST	2023-09-05 10:27:41 BST	2023-09-(
my_turtlebot_2023-09-04-23-20-35.bag.active	▽	0	2023-09-05 08:05:29 BST	2023-09-05 08:05:29 BST	2023-09-05 08:05:29 BST	2023-09-(
my_turtlebot_2023-09-05-02-29-06.bag.active			2023-09-05 13:25:02 BST	2023-09-05 13:25:02 BST	2023-09-05 13:25:01 BST	2023-09-(

Page: 1 of 21893 Page Go to Page: 1 Jump to Offset

```
0x00000000: 23 52 4F 53  42 41 47 20  56 32 2E 30  0A 45 00 00   #ROSBAG V2.0.E..
0x00000010: 00 04 00 00  00 6F 70 3D  03 12 00 00  00 69 6E 64   .....op=.....ind
0x00000020: 65 78 5F 70  6F 73 3D CF  79 5F 15 00  00 00 00 0F   ex_pos=.y_......
0x00000030: 00 00 00 63  6F 6E 6E 5F  63 6F 75 6E  74 3D 1A 00   ...conn_count=..
0x00000040: 00 00 10 00  00 00 63 68  75 6E 6B 5F  63 6F 75 6E   ......chunk_coun
0x00000050: 74 3D 76 01  00 00 BB 0F  00 00 20 20  20 20 20 20   t=v.......
0x00000060: 20 20 20 20  20 20 20 20  20 20 20 20  20 20 20 20
0x00000070: 20 20 20 20  20 20 20 20  20 20 20 20  20 20 20 20
0x00000080: 20 20 20 20  20 20 20 20  20 20 20 20  20 20 20 20
0x00000090: 20 20 20 20  20 20 20 20  20 20 20 20  20 20 20 20
0x000000a0: 20 20 20 20  20 20 20 20  20 20 20 20  20 20 20 20
0x000000b0: 20 20 20 20  20 20 20 20  20 20 20 20  20 20 20 20
0x000000c0: 20 20 20 20  20 20 20 20  20 20 20 20  20 20 20 20
0x000000d0: 20 20 20 20  20 20 20 20  20 20 20 20  20 20 20 20
```

Fig. 5. ROS bag analysis.

These findings explain several key aspects. Firstly, the initial set of hexadecimal values *#ROSBAG V2.0\nE\0\0* indicate the ROS Bag file's format version, which is crucial for determining the file's compatibility with ROS tools.

Additionally, the hexadecimal lines included data fields labelled *chunk_count*, *conn_count*, *index_pos* and *op*. These fields represent various parameters or properties related to the data stored within the bag file. Further analysis identified specific field-value pairs, such as *chunk_count = 0x00000010* and *conn_count = 0x0000000F*, which provide essential metadata about the bag file, including the number of data chunks and the count of connections or channels recorded within the file.

Furthermore, two extra fields were identified: *index_pos* with a value of *0x00000004* and *op* with a hexadecimal value of *0x03BB0F00*. These fields contain crucial information about the bag file's structure or content, which are integral to the *Header Section*. The analysis of this section of the ROS Bag File structure is a critical component containing metadata about the bag file, including *format version*, *compression specifications*, and *other file-level characteristics*.

Summing up, understanding and interpreting these parameters are essential for ensuring the file's integrity and compatibility with ROS tools which are of upmost importance in robotic data analysis and forensics.

B. Significance of ROS Bag Files in Digital Forensics

The analysis of a ROS Bag file using the Autopsy tool revealed critical insights into its structure and metadata. Key findings include identifying the file's format version and the presence of essential data fields such as *chunk_count* and *conn_count*. Additional fields like *index_pos* and *op* were also discovered. These findings primarily relate to the *Header Section* of the ROS Bag File Structure, underscoring its role in storing vital metadata about this file.

On the other hand, identifying the *ROS Bag file's format version* is crucial for determining its compatibility with ROS tools, ensuring seamless data processing and analysis. Data fields such as *chunk_count* and *conn_count* offer valuable metadata about the file, including the number of data chunks and connections recorded. This provides essential context for forensic investigators in interpreting the file's content and structure. Additionally, the fields *index_pos* and *op* have critical information related to the file's internal organization, which highlights the importance of the Header Section.

These findings hold substantial implications for digital forensic investigations and cybercrime in robotics. They empower forensic experts to thoroughly analyze, validate, and interpret data captured within ROS Bag files, enhancing their ability to detect potential cyber threats, malicious activities, or unauthorized operations. Especially, *understanding the format version* ensures that ROS tools can accurately process the file, maintaining the investigation's integrity. Additionally, the metadata on data chunks and connections assists investigators in reconstructing events and contextualizing the recorded data; thereby providing a clearer understanding of the situation under scrutiny.

Another crucial aspect to consider in cybercrime involving robots is the prevalent use of plain text communication within legacy systems utilizing ROS. Such communication methods can introduce vulnerabilities since data transmitted in an unencrypted format is prone to interception and tampering. To mitigate this risk, it is essential for organizations employing ROS in legacy systems to adopt encryption and robust security measures to safeguard data both in transit and at rest. Neglecting to implement these measures may leave robotic systems vulnerable to security breaches and unauthorized disclosure.

However, while the analysis offers valuable insights into the Header Section of the ROS Bag File Structure, it is important to acknowledge its limitations. The findings focus on the file's metadata and structure and may not unveil the precise content or significance of the recorded data. Furthermore, the analysis overlooks potential encryption, compression, or security measures that could impact the interpretation of the file's content. Further investigation may be necessary to explore these aspects thoroughly. Specifically, a significant challenge when working with ROS Bag files is the limitation of tools capable of efficiently parsing and analyzing their contents. While examining the file's structure and metadata are feasible tasks, accessing and deciphering the data they contain can pose significant challenges. This limitation brings up the necessity for developing specialized tools and techniques dedicated to extracting valuable insights from ROS Bag files.

VI. Conclusion and Future Work

This research aimed to explore and demonstrate the potential of utilizing robot-generated data, such as ROS Bag data, as valuable sources of evidence in digital forensic investigations. By identifying relevant artefacts and comprehensively understanding the behavior of robotic systems, we aimed to enhance the ability to successfully resolve crime cases involving robots.

In this work, we answered the research question of how robot-generated data can be used effectively in this context. For this purpose, the methods used included simulating robot activities, collecting structured data, and conducting direct observations, which resulted in a rich dataset. Also, the detailed analysis of a ROS Bag file using the Autopsy tool uncovered valuable insights into its structure and metadata, improving the understanding of how ROS Bag files can be used in digital forensics. We were also able to identify challenges associated with parsing and analyzing ROS Bag files, emphasizing the need for specialized tools. Additionally, we highlighted the vulnerability introduced by plain text communication within legacy ROS systems and recommended implementing encryption and security measures for data at rest and in transit. This research demonstrates the potential of robot-generated data as forensic evidence, making significant contributions to digital forensics, setting the foundation for future investigations and tool development in this emerging domain.

Finally, to further enhance the understanding of ROS Bag files in digital forensics investigations, future work could focus on refining methodologies for analyzing robot-generated data, including decrypting and decompressing the file's content, if applicable. Additionally, examining the timestamps and their synchronization with the forensic workstation's time zone could provide insights into the accuracy of temporal data within the file. Exploring the impact of encryption and security measures on data accessibility and interpretation could be also a valuable venue for further research for exploring legal and ethical aspects while conducting case studies to validate current findings in real-world scenarios. In general, any ongoing exploration of ROS Bag files and any other robot-generated data is essential to continually improve forensic practices in robotics as well as addressing the challenges of data parsing and security not only in legacy ROS systems, but also in related evolving technologies.

References

[1] M. Javaid, A. Haleem and R. S. Pratap, "Substantial capabilities of robotics in enhancing industry 4.0 implementation," *Cognitive Robotics,* vol. 1, pp. 58-75, 2021. https://doi.org/10.1016/j.cogr.2021.06.001

[2] B. Dieber, B. Breiling and S. Taur, "Security for the Robot Operating System," *Robotics and Autonomous Systems,* vol. 98, pp. 192-203, 2017. https://doi.org/10.1016/j.robot.2017.09.017

[3] J.-P. A. Yaacoub, H. N. Noura, O. Salman and A. Chehab, "Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations," *International Journal of Information Security,* vol. 21, pp. 115-158, 2022. https://doi.org/10.1007/s10207-021-00545-8

[4] I. Abeykoon, X. Feng and R. Qiu, "A Forensic Investigation of Robot Operating System," in *IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing (DASC/PiCom/DataCom/CyberSciTech)*, 2017. https://doi.org/10.1109/dasc-picom-datacom-cyberscitec.2017.77

[5] L. Joseph, "Mastering ROS for Robotics Programming," October 2021. [Online]. Available: https://learning.oreilly.com/library/view/mastering-ros-for/9781801071024/B17104_01_Epub_AM.xhtml#_idParaDest-29. [Accessed 15 February 2024].

[6] U. Shirode, A. Aher, P. Bale and A. M. Kadam, "A robotic framework for simulation and control of SCARA robot based on ROS," 2019. [Online]. Available: https://doi.org/10.2139/ssrn.3418758.

[7] M. Quigley, K. Conley, B. P. Gerkey and A. Y. Ng, "ResearchGate," ROS: an open-source Robot Operating System., 2009. [Online]. Available: https://www.researchgate.net/publication/233881999_ROS_an_open-source_Robot_Operating_System. [Accessed 25 April 2024]

[8] "What Is ROS?," 1 February 2023. [Online]. Available: https://roboticsbackend.com/what-is-ros/. [Accessed 15 May 2024].

[9] E. Fosch-Villaronga and T. Mahler, "Cybersecurity, safety and robots: Strengthening the link between cybersecurity and safety in the context of care robots.," *Computer Law & Security Review,* 2021. https://doi.org/10.1016/j.clsr.2021.105528

[10] *ABI Research and Data. "50,000 warehouses will be used by robots by 2025 as barriers to entry fall and AI innovation accelerates",* 2019. https://www.abiresearch.com/press/50000-warehouses-use-robots-2025-barriers-entry-fall-and-ai-innovation-accelerates/ [Accessed 25 June 2024]

[11] "Rogue Robots: Testing the limits of an industrial robot’s security.," 3 May 2017. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/rogue-robots-testing-industrial-robot-security. [Accessed 15 June 2024].

[12] A. Botta, S. Rotbei, S. Zinno and G. Ventre, "Cyber security of robots: A comprehensive survey.," *Intelligent Systems With Applications,* no. 18, 2023. https://doi.org/10.1016/j.iswa.2023.200237

[13] K. Cottrell, D. B. Bose, H. Shahriar and A. Rahman, "An Empirical Study of Vulnerabilities in Robotics.," in *IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC).,* 2021. https://doi.org/10.1109/compsac51774.2021.00105

[14] N. Goerke, D. Timmermann and I. Baumgart, "Who Controls Your Robot? An Evaluation of ROS Security Mechanisms," in *7th International Conference on Automation, Robotics and Applications (ICARA),* 2021. https://doi.org/10.1109/icara51699.2021.9376468

[15] D. Portugal, S. S. Pereira and M. S. Couceiro, "The role of security in human-robot shared environments: A case study in ROS-based surveillance robots," in *26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN),* 2017. https://doi.org/10.1109/roman.2017.8172422

[16] R. Toris, C. A. Shue and S. Chernova, "Message authentication codes for secure remote non-native client connections to ROS-enabled robots," in *IEEE International Conference on Technologies for Practical Robot Applications (TePRA).,* 2014. https://doi.org/10.1109/tepra.2014.6869141

[17] B. Dieber, R. White, S. Taurer, B. Breiling, G. Caiazza, H. I. Christensen and A. Cortesi, "Penetration Testing ROS," *In Studies in computational intelligence,* pp. 183-225, 2019. https://doi.org/10.1007/978-3-030-20190-6_8

[18] R. R. Teixeira, I. P. Maurell and P. Drews, "Security on ROS: analysing and exploiting vulnerabilities of ROS-based systems.," in *Latin American Robotics Symposium (LARS).,* 2020. https://doi.org/10.1109/lars/sbr/wre51543.2020.9307107

[19] J. R. McClean, C. J. Stull, C. R. Farrar and D. Mascareñas, "A preliminary cyber-physical security assessment of the Robot Operating System (ROS)," *Proceedings of SPIE - Defense, Security and Sensing,* 2013. https://doi.org/10.1117/12.2016189

[20] I. Abeykoon and X. Feng, "Challenges in ROS Forensics," in *IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI),* 2019. https://doi.org/10.1109/smartworld-uic-atc-scalcom-iop-sci.2019.00299

[21] V. Vilches, "Volatile memory forensics for the Robot Operating System.," arXiv., 2018. https://doi.org/10.48550/arXiv.1812.09492

[22] M. M. Basheer and A. Varol, "An overview of robot operating system forensics.," in *1st International Informatics and Software Engineering Conference (UBMYK).,* 2019. https://doi.org/10.1109/ubmyk48245.2019.8965649

[23] map_server, "ROS Wiki.," 23 March 2020. [Online]. Available: http://wiki.ros.org/map_server. [Accessed 18 August 2023].

AUTHORS

Shahrzad Zargari

Shahrzad is the principal lecturer of the cyber security and computer networks subject group lead at Sheffield Hallam University. Shahrzad has worked in the IT industry for over 15 years and gained a great deal of experience in computer hardware, software, and business management. Shahrzad's passion lies in the realm of digital forensics and security. She advocates collaboration among the government, industry, and academia. Her mission involves sharing information and nurturing the next generation of cybersecurity experts through education. Shahrzad is the director and steering committee member of the Yorkshire Cyber Security Cluster and the vice chair of BCS South Yorkshire Branch. Shahrzad is a member of the UKC3 cyber skills working group. Shahrzad is an experienced researcher (CENTRIC), having published book chapters and many papers in conferences, journals, and magazines. Additionally, she is the associate editor of the Information Security Journal: A Global Perspective at Taylor and Francis.

Tharmini Janarthanan

Tharmini, a Lecturer in Cybersecurity and Digital Forensics at Sheffield Hallam University is a dedicated Information Security Professional and a Certified ISO 27001:2022 Lead Auditor with over five years of experience. Tharmini's expertise encompasses information, technology, privacy risk management, and digital forensics. Her passion for cybersecurity drives her academic research in digital forensics, where she has undertaken numerous research projects and published papers in peer-reviewed international conferences, journals, and book chapters. In addition to her academic achievements, Tharmini is a member of CIISec and serves on the British Computing Society (BCS) committee in South Yorkshire, further demonstrating her credibility and involvement in the industry.