DOI: 10.1109/CIoT63799.2024.10757025

Beyond the Lens: False Data Injection Attacks on IIoT-Cameras through MQTT Manipulation

Wael Alsabbagh[1,2], Chaerin Kim[1], Nitin Sanjay Patil[1] and Peter Langendörfer[1,2]
[1]*Brandenburg University of Technology Cottbus-Senftenberg, Cottbus, Germany*
[2]*IHP – Leibniz-Institut für innovative Mikroelektronik, Frankfurt (Oder), Germany*
E-mail: (Wael.Alsabbagh, Chaerin.Kim, patilnit, Peter.Langendoerfer)@b-tu.de

***Abstract*—In the era of Industrial Internet of Things (IIoT), where interconnected devices streamline industrial processes, safeguarding the integrity and security of these systems becomes imperative. This paper delves into the exploration of vulnerabilities within IIoT-based systems, focusing on the manipulation of camera views through MQTT protocol exploitation. Our work unveils a sophisticated attack vector targeting the interaction between IIoT cameras and central control systems, leveraging weaknesses within the MQTT protocol. Through a meticulously designed attack scenario, we demonstrate how attackers can surreptitiously inject and manipulate camera feeds, presenting fabricated imagery to deceive operators and compromise system integrity. By exploiting vulnerabilities in MQTT communication, coupled with strategic manipulation of camera views, adversaries can obscure their activities, posing significant risks to industrial operations. Furthermore, we discuss comprehensive security measures and countermeasures to fortify IIoT systems against such attacks, aiming to mitigate potential threats and ensure the resilience of industrial infrastructures. All our attack codes as well as a proof-of-concept are publicly available for further research.**

***Index Terms*—IIoT, IIoT-Cameras, MQTT, Cyberattacks, Cybersecurity.**

I. Introduction

The Industrial Internet of Things (IIoT) has reshaped industrial processes by seamlessly integrating IoT-enabled devices, particularly Programmable Logic Controllers (PLCs) into traditional systems [1]. These PLCs serve as the backbone of IIoT infrastructure, enabling real-time monitoring, predictive maintenance, and remote control, thereby enhancing operational efficiency and productivity to unprecedented levels [2].

At the heart of IIoT systems lies the Message Queuing Telemetry Transport (MQTT) protocol, renowned for its lightweight and efficient messaging mechanism [3]. MQTT facilitates seamless communication between IoT devices and PLCs, forming the core of real-time data exchange in industrial environments. Its publish-subscribe model allows devices to asynchronously send and receive messages, ensuring swift and reliable communication even in bandwidth-constrained or unreliable network conditions.

However, as IIoT systems evolve, so do the security challenges. The adoption of MQTT introduces vulnerabilities that attackers can exploit to compromise system integrity and confidentiality. MQTT's lightweight nature, while advantageous for efficient communication, also makes it susceptible to various security threats such as unauthorized access, message tampering, and eavesdropping [4].

In this paper, we delve into the realm of IIoT security, focusing specifically on the manipulation of camera views through MQTT exploitation. The possibility of camera view manipulation is crucial to highlight because numerous reported incidents[1] involve compromising an IoT device and gaining access to its camera feed. By exploiting vulnerabilities within MQTT, attackers can clandestinely manipulate camera functionality, potentially compromising operational integrity and posing significant security risks to industrial environments.

Our investigation examines the intricacies of MQTT communication to uncover potential attack vectors that adversaries can leverage to subvert camera systems within IIoT environments. We explore the intersection of MQTT and camera systems, demonstrating how attackers can exploit MQTT vulnerabilities to manipulate camera views through covert messaging techniques. Our proof of concept [16] as well as the attack codes [17] are publicly available.

Moreover, we develop an attack scenario tailored to illustrate the critical importance of securing MQTT implementations in industrial environments. This scenario underscores the risks posed by the lack of authentication, integrity, and confidentiality in MQTT communications, resulting in the insertion of false data and potential destructive impacts on IIoT systems. Our investigation additionally showed that attackers lacking authentication knowledge regarding the MQTT broker can still execute a False Data Injection Attack (FDIA) which was not feasible in our former papers [3], [11]. In response to these security challenges, we discuss security recommendations aimed at fortifying IIoT camera systems against MQTT-based attacks. By incorporating secure authentication mechanisms, robust encryption techniques, and stringent access control measures, organizations can enhance the resilience of their IIoT infrastructures and mitigate emerging threats effectively.

The aim of this paper is to raise awareness of the security implications associated with MQTT in IIoT camera systems and empower stakeholders to adopt proactive security measures. By fostering a deeper understanding of MQTT vulnerabilities and their impact on industrial operations, we

978-1-7281-9023-5/21/$31.00 ©2021 IEEE

[1]http://www.insecam.org/en/

strive to pave the way for a more secure and resilient IIoT ecosystem.

The rest of the paper is organized as follows: Section II reviews related works. Section III describes the experimental setup. Section IV analyzes MQTT-enabled IIoT Cameras, focusing on security vulnerabilities. Section V explains our attack methodology. Section VI presents security recommendations and mitigation strategies. Finally, Section VII offers concluding remarks summarizing our research insights.

II. Related Work

In recent years, there has been a notable surge in research interest surrounding the manipulation of IIoT cameras within MQTT-based IIoT systems. Yanmao et al. [5] demonstrated how camera-based perception can be subtly manipulated, enabling attackers to create spurious objects or alter existing ones by remotely projecting adversarial patterns into cameras. They exploited common optical imaging system effects, including lens flare/ghost effects and auto-exposure control, to achieve this manipulation. Venkatesan et al. [6] provided a comprehensive review of existing and potential threats in video surveillance systems, including CCTV and IP-camera systems in IoT environments. Their work offers valuable insights for identifying security risks associated with system design and deployment, stimulating further research in this burgeoning field. Costin [7] conducted a systematic review of novel threats in video surveillance, CCTV, and IP-camera systems based on publicly available data. This review aids in understanding and identifying security and privacy risks associated with system development, deployment, and use. Another research group [8] performed a security assessment of robot cameras using the Robot Operating System (ROS), identifying numerous security flaws that could be exploited to intercept flows from robot cameras. They proposed an intrusion detection system to detect abnormal flows, enhancing camera system security. Bugeja et al. [9] focused on IoT cameras frequently integrated into smart home systems. Their research explored vulnerabilities associated with "remote access" and the exposure of "available resources (targets)" through public search, highlighting the heightened risk of exploitation for easily accessible internet-connected cameras.

Our research builds upon these foundations, narrowing the focus to FDIA scenarios targeting cameras in IIoT systems. By concentrating on the MQTT protocol, which often facilitates the transmission of sensitive data within IIoT networks, we aim to deepen the understanding of FDIA threats specific to camera devices.

III. Experimental Setup

To validate the attack strategy delineated in this paper, we utilized a Fischertechnik training factory [2], alternatively referred to as Lernfabrik 4.0 9V. This facility is specifically designed for the advancement of Industry 4.0 applications, as illustrated in Figure 1. The factory serves as a platform for simulating, comprehending, and implementing automated industrial processes on a miniature scale prior to full-scale deployment. It consists of four industrial workstations [12]:

- Vacuum Gripper Robot (VGR): Featuring a robot equipped with a vacuum gripper.
- High-Bay Warehouse (HBW): Housing an automated rack warehouse with 3x3 slots.
- Multi-Processing Station (MPO): Simulating an industrial oven and machining bench.
- Sorting Line and Color Detection (SLD): Equipped with a conveyor, color detection system, and part separator.

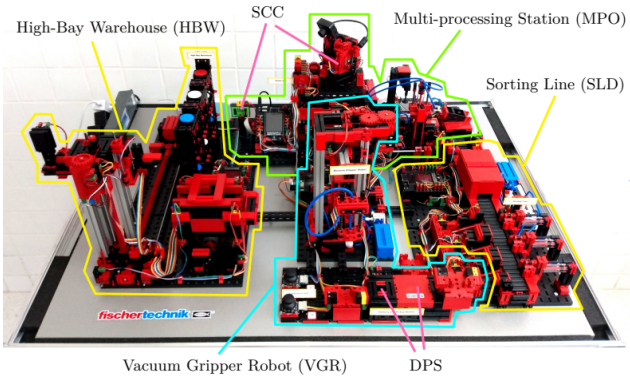

Fig. 1. Fischertechnik Lernfabrik 4.0 9V, adopted from [12].

The factory's setup extends beyond mere workstations, encompassing a variety of modules. These include a Delivery and Pickup module (DPS), a Near-Field Communication (NFC) reader/writer, an RGB camera module, and a Sensor Station with Camera (SSC), which is tailored to capture additional data. The production simulation entails the movement of small workpieces of varying colors (white, red, and blue) within the facility. Each workpiece is equipped with a distinct NFC identification tag, facilitating the logging of its color and production history, inclusive of timestamps. The factory setup is orchestrated around six Fischertechnik TXT 4.0 controllers, as depicted in Figure 2.

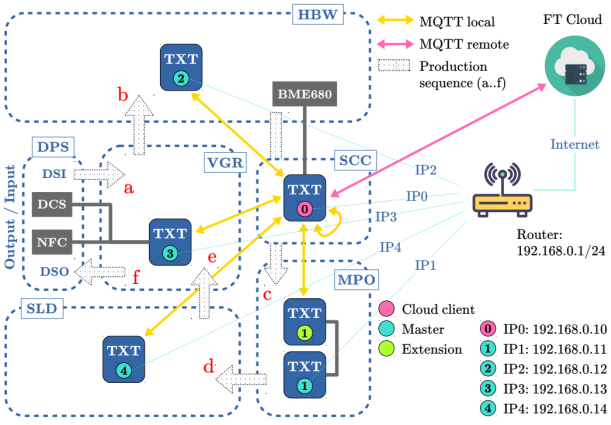

Fig. 2. Block Diagram of the Lernfabrik 4.0, adopted from [12].

[2]https://www.fischertechnik.de/de-de/industrie-und-hochschulen/technische-dokumente/simulieren/lernfabrik-4-0-9v-v-2

The bidirectional communication between the TXT controllers and the central TXT-0 is facilitated, with TXT-0 serving as an MQTT broker for controllers connected to a Wi-Fi network via a TP-Link Router. Configured as a cloud client, TXT-0 establishes a connection to Fischertechnik's cloud through a remote MQTT bridge. Users can access a cloud dashboard via a web browser to oversee and engage with specific factory production processes. Acting as a cloud master, TXT-0 is equipped with a Universal Serial Bus (USB) camera and environmental sensors to monitor air pressure, humidity, temperature, and gases. TXT-1 to TXT-4 controllers also operate as cloud masters, with TXT-1 featuring an extension. TXT-2 is responsible for managing the warehouse, TXT-3 controls the DPS module, and TXT-4 oversees the SLD station. The block diagram illustrates the standard manufacturing sequence from DSI to DSO, with each TXT controller executing compiled C++ code tailored for its station.

IV. MQTT-enabled IIoT Cameras

A. Overview

IIoT cameras represent a specialized class of devices tailored to address the unique demands of industrial settings, spanning from manufacturing plants and warehouses to critical infrastructure facilities. These cameras boost a suite of advanced features including high-resolution imaging, infrared capabilities, and edge computing functionalities geared towards facilitating real-time data processing within industrial environments. The integration of MQTT communication into IIoT camera systems yields a plethora of benefits:

- Reduced Latency: By harnessing MQTT's lightweight protocol, IIoT cameras can effectively minimize communication overhead, thereby facilitating low-latency data transmission between the cameras and backend systems. This feature is crucial for ensuring timely decision-making and response within industrial operations.

- Scalability: The publish-subscribe architecture of MQTT empowers IIoT camera networks with inherent scalability. This architecture enables the seamless expansion and adaptation of camera networks to accommodate dynamic changes in device connectivity and data volume, fostering flexibility and scalability within industrial environments.

- Interoperability: MQTT's standardized protocol facilitates seamless interoperability between IIoT cameras and other MQTT-compliant devices and platforms. This interoperability promotes easy integration and collaboration between disparate systems, thereby streamlining operational workflows and promoting ecosystem expansion.

-Reliability: MQTT's Quality of Service (QoS) levels provide assurances of reliable message delivery, bolstering data integrity and system robustness in IIoT camera deployments. This reliability ensures consistent and dependable communication between IIoT cameras and backend systems, essential for maintaining operational efficiency and security.

However, it's essential to acknowledge that while these benefits are significant, they also come with inherent risks. The adoption of IIoT cameras and MQTT communication opens the door for potential security vulnerabilities and cyber threats. Attackers may exploit weaknesses in camera systems or MQTT implementations to gain unauthorized access, manipulate data, or disrupt industrial operations.

B. Security Concerns

The MQTT protocol harbors various security vulnerabilities, as detailed in [10], which could potentially lead to diverse attack vectors against IoT and IIoT systems. Here, in the following we specifically highlight the security issues that render the MQTT-enabled IIoT cameras susceptible to FDIAs.

*1) **Absence of End-to-End Encryption**:* IIoT cameras relying on MQTT, by default, lack inherent end-to-end encryption. Unencrypted communication between cameras and brokers exposes messages to interception and manipulation. In our experimental setup, an attacker intercepts and alters the camera payload transmitted from TXT controllers to the MQTT broker (TXT-0), thereby introducing false information to the user.

*2) **Insecure Broker Configuration**:* Improperly configured MQTT brokers create vulnerabilities, allowing attackers to exploit weaknesses. For instance, if a broker permits anonymous connections or lacks robust access controls, attackers can inject false data, fabricating the views of the cameras connected to the systems. In our Lernfabrik example, an attacker leverages a misconfigured broker to gain administrative privileges, enabling manipulation of critical camera settings, injection of false images, etc.

*3) **Message Tampering**:* Without integrity checks, MQTT messages are susceptible to tampering during transit. In our Lernfabrik scenario, an attacker intercepts and alters control commands transmitted from a TXT controller to a mini motor of the IIoT camera module through the MQTT broker, inducing Fischertechnik to execute incorrect actions.

*4) **Authorization Issues Based on Topics**:* MQTT's topic-based publish-subscribe model is vulnerable to insecure authorization mechanisms. In our factory context, an attacker may subscribe to a topic containing critical data (e.g., IIoT camera), intercept legitimate MQTT messages, and inject manipulated data into the same topic, thereby resulting in the processing of false information.

*5) **Retained Messages Misuse**:* IIoT cameras utilizing MQTT's support for retained messages introduce a potential avenue for injecting false data if not configured and monitored properly. Without vigilance, retained messages can lead to unexpected behavior and compromise data integrity.

V. False Data Injection Attack Approach

Figure 3 provides a high-level overview of our FDIA attack scenario. It consists of three primary phases:

- Breaking the authentication of the TP-Link Router.
- Identifying Targeted Topics (Sniffing Attack).
- Topics Substitution.

In this work, we omit the detailed depiction of the first phase, which has been extensively covered in our previous paper [3]. Our emphasis in this paper is solely on phases 2 and 3, as elucidated in the following subsections.

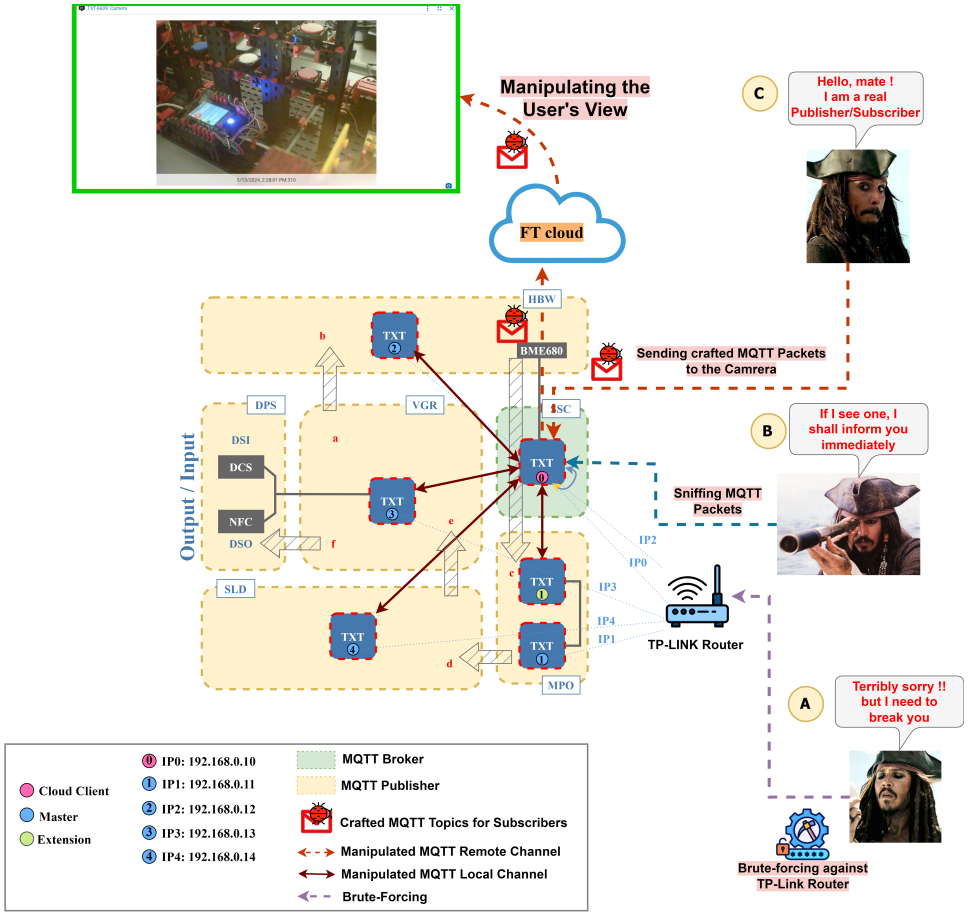

Fig. 3. High-level Overview of our False Data Injection Attack, aimed at manipulating the camera view

A. *Sniffing the MQTT Topics (MitM)*

In MQTT-based IIoT systems, an attacker with access to the MQTT network, can employ various techniques to eavesdrop on MQTT topics, potentially compromising the confidentiality of sensitive information. To this end, an attacker can utilize a packet sniffing tool, such as Wireshark[3], to capture and inspect MQTT packets traversing the network. This allows him/her to scrutinize the content of messages related to targeted topics. Subsequently, armed with a detailed understanding of the legitimate data flow, the attacker can inject false or deceptive information into the MQTT topics during transmission. This manipulative approach seeks to sow confusion among legitimate participants in the IIoT network, potentially leading to erroneous decision-making or compromising the integrity of critical processes relying on the exchanged MQTT messages. Figure 4 illustrates a *PUBLISH* message with the topic "i/cam". This message originates from the publisher, in this instance, the TXT-0 controller, and is transmitted to the MQTT broker, eventually reaching the remote subscriber, as represented by the user dashboard in our case.

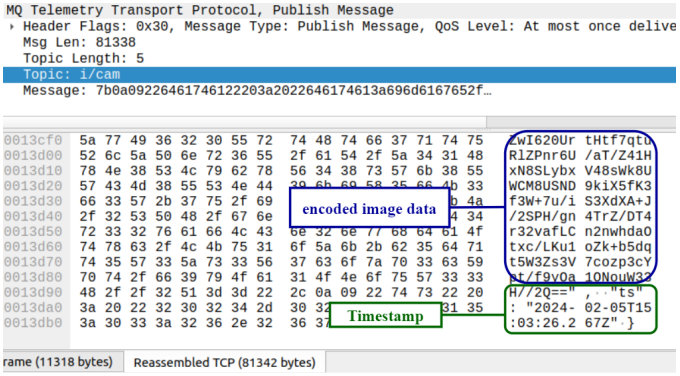

Fig. 4. MQTT *PUBLISH* Message "i/cam", including the encoded data of camera view.

The subscriber actively subscribes to this topic, enabling the retrieval of data from the message to update the dashboard accordingly. The purpose of the "i/cam" topic is to display the video feed on the dashboard. The video is generated using the "opencv" library, and the captured image, with a fixed size, is encoded into a string using the Base64 algorithm [4].

[3]https://www.wireshark.org/

[4]https://developer.mozilla.org/en-US/docs/Glossary/Base64

Subsequently, the publisher sends the message with the current timestamp. Upon receiving the packet, the subscriber decodes the retrieved Base64 data string and presents it on the camera view of the user dashboard. In the camera view, the timestamp in each packet is displayed alongside. The dynamic updating of the dashboard based on these fields ensures that users are promptly informed about the current camera view at the main station. Figure 5 illustrates the topics that we sniffed during our scan of the MQTT broker.

In our previous paper [11], we highlighted the necessity for attackers to breach the authentication of the MQTT broker to send malicious *PUBLISH* messages to the dashboard. In this work, we tackled this challenge by implementing the *mosquitto* [5] tool, an open-source MQTT message broker service. *Mosquitto* employs the MQTT protocol for device communication through message exchange. Among the various message brokers supporting MQTT, *mosquitto* distinguishes itself as a compact and efficient implementation of MQTT v3.1/3.1.1. Authentication in *mosquitto* is governed by the *mosquitto.conf* file, which serves as a centralized configuration point. This file can be located anywhere accessible to *mosquitto*. It provides three authentication options: password files, authentication plugins, and unauthorized/anonymous access. By default, *mosquitto* operates without a configuration file, relying on preset values. The simplest option is to forego authentication entirely, which is the default setting if no other options are specified. However, this default approach poses potential security vulnerabilities, as unauthorized access may compromise the integrity and confidentiality of IIoT systems. In some IIoT setups, administrators create custom configuration files and associate them with the MQTT binaries. Yet, even with modifications to the *mosquitto.conf* file, MQTT may still permit anonymous authentication.

B. Sending Manipulated MQTT Topics to the Camera

Figure 6 illustrates our FDIA attack strategy aimed at manipulating the camera view for the user. In this scenario, the attacker generates deceptive MQTT topics closely resembling the legitimate ones associated with user-dashboard data. The objective is to trick the user-dashboard into interpreting and displaying altered data as authentic. Specifically, for our Lernfabrik system, we devised a malicious topic named "*i/cam*". This topic is intended to showcase the camera view on the dashboard, providing dynamic real-time updates. However, a security vulnerability exists in the user-dashboard, as it fails to validate the authenticity of material identifiers. This loophole allows potential attackers to manipulate the "** $< datastring >$" field by inserting an encoded image that the attacker wishes to inject.

Despite this security flaw, the user-dashboard dynamically refreshes itself based on the information from these fields, ensuring users receive real-time updates for the camera view on the main station (SSC). Once the attacker substitutes genuine packets with manipulated ones, they inject the latter into the MQTT message stream. Consequently, the malicious topic infiltrates the user-dashboard of subscribed users, effectively replacing genuine data with manipulated information. In Figure 7, two screenshots of the user-dashboard are provided. The one highlighted in blue represents the original view, reflecting the current camera feed. However, users are presented with the view highlighted in red, displaying the falsified view generated by the attacker.

What exacerbates this attack scenario is that if the attacker transmits the normal view while executing the attack, it may obscure the user's perception. Alternatively, the attacker could expose critical industrial secrets of the manufacturer, which may be linked to their intellectual assets, by intercepting packets transmitting the camera view to the user-dashboard. In real-world IIoT systems, such a scenario could result in significant damage within the facility, while the user is continuously shown that the physical process is operating normally.

VI. Security Recommendations

Mitigating False Data Injection Attacks (FDIA) against MQTT-enabled IIoT cameras necessitates a robust set of security measures and best practices. Below, we outline key strategies to counter such attacks effectively, thereby ensuring the integrity and reliability of IIoT systems.

A. Message Validation and Filtering

1) ***Topic Structure Validation****:* Implement stringent checks on the structure of MQTT topics. Enforce clear and predefined naming conventions for topics, rejecting any messages that deviate from these predefined structures.

2) ***Regular Expression Patterns****:* Utilize regular expressions to define and validate topic patterns, allowing for flexible yet controlled matching. This ensures that subscribed topics adhere to the expected formats.

B. TLS/SSL Encryption

The communication channels between MQTT clients and brokers are susceptible to various network-based attacks, such as Man-in-the-Middle (MitM), replay attacks, and injection attacks. Implement Transport Layer Security (TLS) or its predecessor, Secure Sockets Layer (SSL), to encrypt MQTT communication [13]. This encryption prevents adversaries from intercepting and tampering with data in transit, thereby bolstering the security of IIoT cameras.

C. Topic Whitelisting

Implementing topic whitelisting restricts MQTT clients to subscribe only to specific topics, as proposed in [14]. This measure helps to control the flow of information and prevents unauthorized access to sensitive data, such as the encoded data representing the camera view in our case.

[5]https://mosquitto.org/

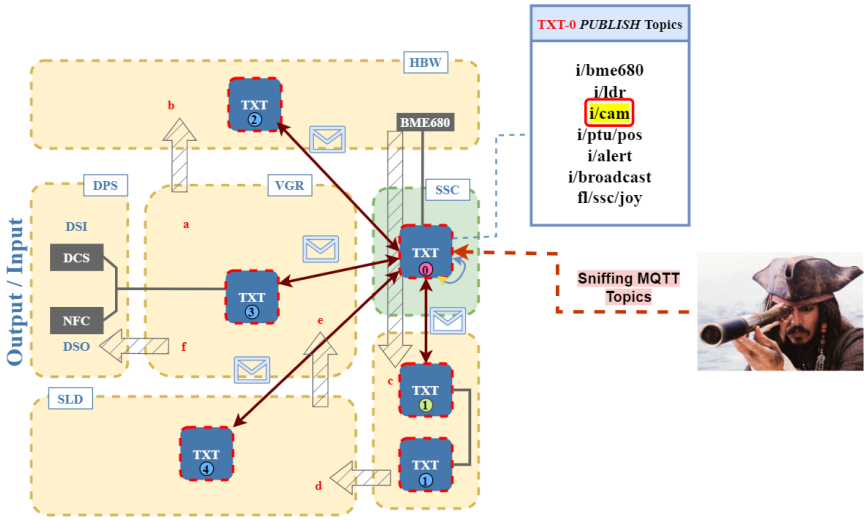

Fig. 5. MQTT *PUBLISH* Topics transmitted over the MQTT network of our Lernfabrik 4.0 9V.

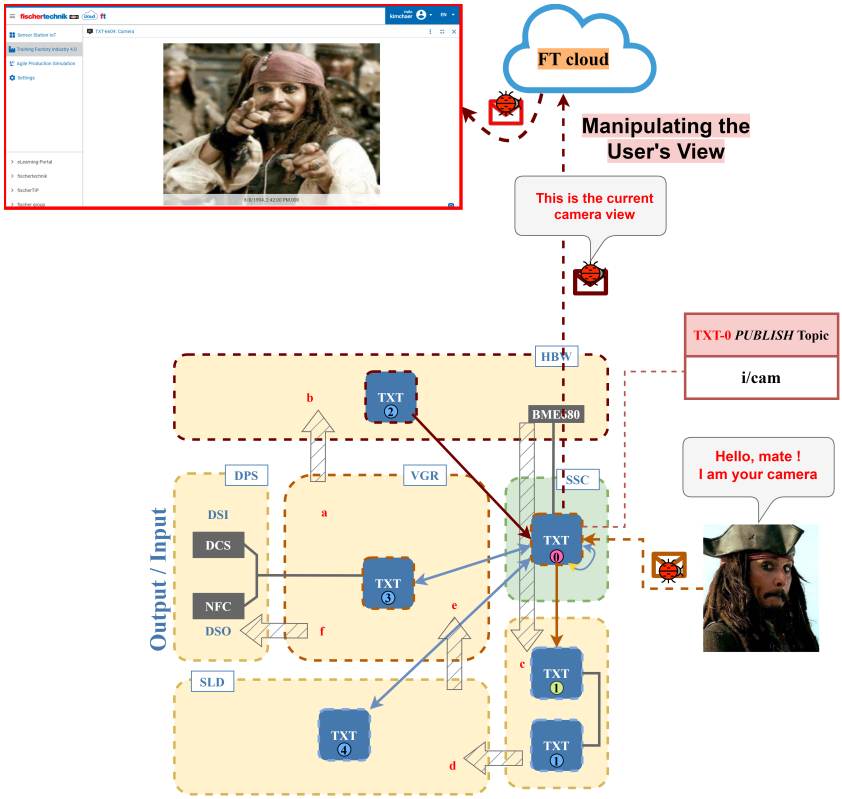

Fig. 6. FDIA attack, manipulating the camera view for the user.

D. Payload Validation

One critical vulnerability exploited in FDIA attacks is the lack of payload validation in MQTT packets. Implement an integrity check mechanism, as recommended in [15], to validate payloads against expected formats and detect malicious content. This ensures that MQTT payloads are not manipulated by attackers, enhancing the overall security of IIoT systems.

VII. Conclusion

This paper extensively examines scenarios of FDIA in MQTT-enabled IIoT camera systems, highlighting vulnerabilities in the MQTT protocol that can result in significant disruptions, impairing what users see and causing confusion. Our experiments illustrate the manipulation of MQTT payloads aimed at controllers, presenting users with false camera views. The severity of these findings underscores the potential harm

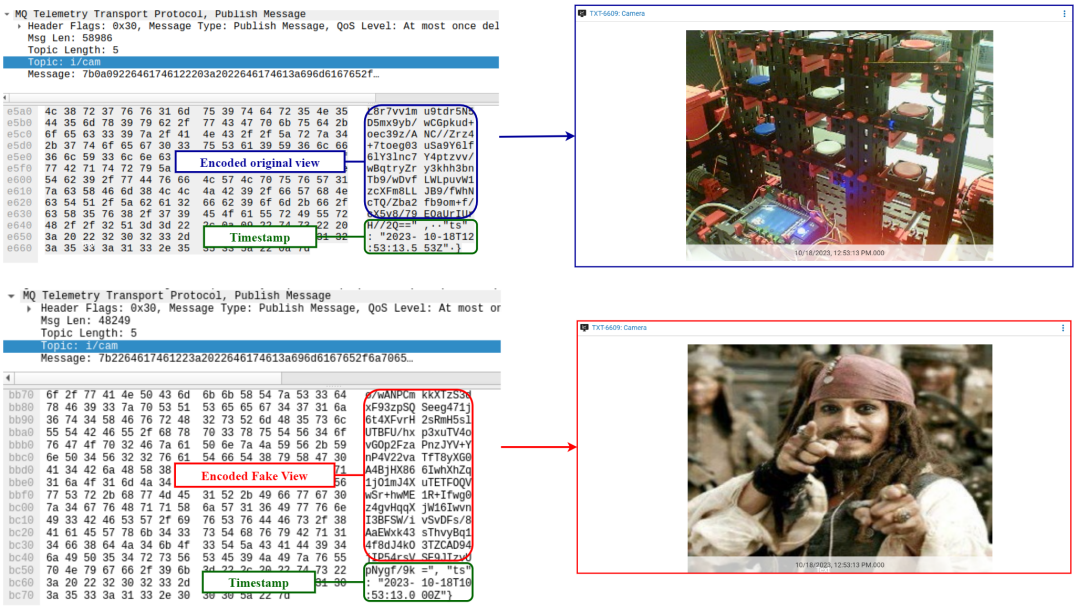

Fig. 7. Crafting the packet with topic "i/cam" to fabricate the user view on dashboard.

if such attacks were to occur in real IIoT systems. To address these risks effectively, we delved into security measures outlined in existing literature to bolster MQTT's overall security. It's crucial to implement these measures collectively to thwart the attack scenario we've outlined. By providing transparency and a commitment to further research, we have made our proof of concept and attack codes publicly accessible. This paper not only raises awareness of vulnerabilities in MQTT-based IIoT camera systems but also offers practical insights to strengthen defenses against potential threats. Sharing our findings and proposed solutions contributes to the ongoing discourse on IIoT system security, facilitating progress in safeguarding critical infrastructure.

REFERENCES

[1] W. Alsabbagh and P. Langendörfer, "Security of Programmable Logic Controllers and Related Systems: Today and Tomorrow," IEEE Open Journal of the Industrial Electronics Society, vol. 4, pp. 659-693, 2023, doi: 10.1109/OJIES.2023.3335976.

[2] W. Alsabbagh, C. Kim and P. Langendörfer, "Investigating the Security of OpenPLC: Vulnerabilities, Attacks, and Mitigation Solutions," in IEEE Access, vol. 12, pp. 11561-11583, 2024, doi: 10.1109/ACCESS.2024.3356051.

[3] W. Alsabbagh, C. Kim, and P. Langendörfer, "Silent Sabotage: A Stealthy Control Logic Injection in IIoT Systems," Submitted at the 5th Silicon Valley Cybersecurity Conference (SVCC 2024), June 17 - 19, 2024, South Korea ,doi: 10.13140/RG.2.2.33854.25925.

[4] S. Lakshminarayana, A. Praseed and P. S. Thilagam, "Securing the IoT Application Layer from an MQTT Protocol Perspective: Challenges and Research Prospects," in IEEE Communications Surveys & Tutorials, doi: 10.1109/COMST.2024.3372630.

[5] Y. Man, M. Li and R. Gerdes, "Remote Perception Attacks against Camera-based Object Recognition Systems and Countermeasures," ACM Trans. Cyber-Phys. Syst.8,2, Article 14, April 2024, 27 pages. doi: 10.1145/3596221.

[6] P. Vennam, et al., "Attacks and Preventive Measures on Video Surveillance Systems: A Review." Appl. Sci. 2021, 11, 5571, doi: 10.3390/app11125571.

[7] A. Costin, "Security of CCTV and Video Surveillance Systems: Threats, Vulnerabilities, Attacks, and Mitigations," In Proceedings of the 6th International Workshop on Trustworthy Embedded Devices (TrustED '16). Association for Computing Machinery, New York, NY, USA, 45–54, doi 10.1145/2995289.2995290.

[8] S. Lagraa, M. Cailac, S. Rivera, F. Beck and R. State, "Real-Time Attack Detection on Robot Cameras: A Self-Driving Car Application," 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 2019, pp. 102-109, doi: 10.1109/IRC.2019.00023.

[9] J. Bugeja, D. Jönsson and A. Jacobsson, "An Investigation of Vulnerabilities in Smart Connected Cameras," 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Athens, Greece, 2018, pp. 537-542, doi: 10.1109/PERCOMW.2018.8480184.

[10] A. Al-Ani, W. K. Shen, A. K. Al-Ani, S. A. Laghari and O. E. Elejla, "Evaluating Security of MQTT Protocol in Internet of Things," 2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Regina, SK, Canada, 2023, pp. 502-509, doi: 10.1109/CCECE58730.2023.10288857.

[11] W. Alsabbagh, C. Kim and P. Langendörfer, "A Payload of Lies: False Data Injection Attacks on MQTT-based IIoT Systems," to be presented at IECON 2024- 50th Annual Conference of the IEEE Industrial Electronics Society, Chicago, Illinois, USA, 2024, doi: 10.13140/RG.2.2.14002.58565.

[12] A. L. de Sousa and A. S. de Oliveira, "Order-Controlled Production Employing Multi-Agent and Flexible Job-Shop Scheduling on a Physical Simulation Platform," 2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE), São Bernardo do Campo, Brazil, 2022, pp. 229-234, doi: 10.1109/LARS/SBR/WRE56824.2022.9995868.

[13] I. L. B. M. Paris, M. H. Habaebi and A. M. Zyoud, "Implementation of SSL/TLS Security with MQTT Protocol in IoT Environment," Wireless Pers Commun 132, 163–182 (2023), doi: 10.1007/s11277-023-10605-y.

[14] E. G. Ch. Zhi et al. "MQTT Traffic Collection and Forensic Analysis Framework," Digital Forensics and Cyber Crime: 13th EAI International Conference, ICDF2C 2022, Boston, MA, November 16-18, 2022, Proceedings. Vol. 508. Springer Nature, 2023.

[15] Ch. Patel, N. Doshi, "A Novel MQTT Security framework In Generic IoT Model", Procedia Computer Science, v. 171, 2020, pp. 1399-1408, doi: 10.1016/j.procs.2020.04.150.

[16] [Online]. Available: https://youtu.be/-oeWzLR2Q2k.

[17] [Online]. Available: https://github.com/rnrn0909/beyondthelens.