Hacking the Backbone: Shell Reverse Attacks on IIoT Systems

Wael Alsabbagh
Brandenburg University of Technology
Cottbus-Senftenberg
Cottbus, Germany
wael.alsabbagh@b-tu.de

Chaerin Kim
Brandenburg University of Technology
Cottbus-Senftenberg
Cottbus, Germany
chaerin.kim@b-tu.de

Nitin Sanjay Patil
Brandenburg University of Technology
Cottbus-Senftenberg
Cottbus, Germany
patilnit@b-tu.de

Peter Langendörfer
IHP – Leibniz-Institut für innovative Mikroelektronik
Frankfurt (Oder), Germany
langendoerfer@ihp-microelectronics.com

Abstract

As Industrial Internet of Things (IIoT) systems grow, they face increasing risks from cyber threats like reverse shell attacks. These attacks exploit IIoT vulnerabilities, allowing unauthorized remote access and jeopardizing industrial operations. This paper examines how adversaries use malware to establish hidden connections to their servers, bypassing traditional firewalls. By exploiting command injection vulnerabilities, attackers deploy reverse shell scripts for persistent access. Our research highlights the role of weak credentials and brute force attacks in gaining initial access. We demonstrate deploying Python-based reverse shell payloads via SFTP protocol, disrupting operations and enabling unauthorized commands and data theft. Using experiments with the Fischertechnik Lernfabrik 4.0, we show the impact of these attacks. To counteract these threats, we recommend robust security measures such as network segmentation, patch management, and advanced intrusion detection systems. All attack codes and a proof-of-concept are publicly available.

CCS Concepts

• **Security and privacy** → *Security protocols.*

Keywords

IIoT, Reverse Shell, Cyberattacks, Cybersecurity, Industrial Control System (ICS)

1 Introduction

The Industrial Internet of Things (IIoT) is transforming industrial operations by enabling advanced connectivity, automation, and data exchange [1]. However, this increased connectivity also introduces significant security challenges. IIoT systems, comprising diverse sensors, actuators, and control systems, are vulnerable to sophisticated cyberattacks. Among these, reverse shell attacks are particularly severe, exploiting system vulnerabilities to gain unauthorized remote access and control. Reverse shell attacks exploit IIoT device vulnerabilities to establish a connection back to the attacker's server, bypassing traditional security measures like firewalls [2]. This method allows attackers to set up a command-and-control (C2) infrastructure, gaining full control over the target system.

This paper focuses on how adversaries deceive industrial operators into installing malware on IIoT devices, creating a persistent communication channel with the attacker's server. Such malware can be delivered via phishing emails, malicious websites, and compromised software updates. Additionally, command injection vulnerabilities in IIoT servers can be exploited to deploy reverse shell scripts, providing attackers with covert access to execute commands, exfiltrate data, and disrupt operations. The impact of these attacks can lead to operational disruptions, safety hazards, and significant financial losses. The main aim of this work is to highlight the security risks posed by reverse shell attacks on IIoT systems and provide stakeholders with the knowledge to implement effective security measures. By raising awareness of these vulnerabilities and their potential impact, we seek to contribute to a more secure and resilient IIoT environment. Our proof-of-concept and all related codes are published in [15] and [16].

Our attack approach was tested on a Fischertechnik 4.0 system[1] to illustrate the real-world implications of these vulnerabilities. It includes the following steps:

- **Bypassing Router Authentication:** The attacker exploits vulnerabilities in the router to gain unauthorized access.
- **Bypassing SFTP Authentication of the IIoT Device:** The attacker circumvents the security mechanisms of the IIoT device's file transfer system using SFTP.
- **Uploading Reverse Shell Payload:** The attacker uploads a malicious reverse shell script to the IIoT device.
- **Executing the Reverse Shell Payload:** The attacker runs the uploaded script to establish a shell connection back to their server.
- **Gaining Root Access:** The attacker escalates privileges to gain root access on the compromised IIoT device.
- **Injecting Control Commands:** Using the established reverse shell, the attacker injects malicious control commands to manipulate the IIoT system's operations.

The rest of the paper is organized as follows: Section 2 reviews related works. Section 3 describes the experimental setup. Section 4 details the vulnerabilities in IIoT systems that allow reverse shell attacks. Section 5 presents our attack scenario. Section 6 introduces security recommendations. Finally, Section 7 offers concluding remarks summarizing our research insights.

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

2 Related Work

In recent years, research on reverse shell attacks has been extensive, but few have focused on their implications within IIoT systems. Al-Hawawreh [5] examined security vulnerabilities in IIoT environments, particularly in routers and firewalls. Their findings revealed the systems' susceptibility to reverse shell attacks, where attackers exploit vulnerabilities to gain unauthorized access. By scanning for open ports and using default credentials or phishing tactics, attackers can breach defenses, posing a severe threat to industrial operations. Exploiting software vulnerabilities, like those in pfsense, allows attackers to establish reverse shell backdoors with root privileges, enabling remote command execution and potential system takeover. Building on this, another research group [6] demonstrated malware introduction into IIoT systems through trojan messages or downloads, opening TCP ports and facilitating access to critical components like the Secure Shell (SSH) Protocol. Manglani [7] highlighted the effectiveness of shellcode in controlling host systems within IIoT environments, noting that their shellcode evaded all tested antivirus software, emphasizing IoT security gaps. Kaushik et al. [8] discussed various malicious practices by hackers, focusing on web application and software vulnerabilities, and the need for proactive measures like regular software updates.

Our work differentiates itself by targeting vulnerabilities in the Secure File Transfer Protocol (SFTP) commonly used in IIoT devices. We show how attackers can exploit these vulnerabilities to upload a reverse shell script, establishing a connection with the compromised device, enabling attackers to take control and inflict damage on the system.

3 Experimental Setup

To validate the attack strategy outlined in this paper, we employed a Fischertechnik training factory, also known as Lernfabrik (Learning Factory) 4.0 9V[2]. This facility is specifically designed to support the development of Industry 4.0 applications, as illustrated in Figure 1. The factory serves as a platform for simulating, understanding, and implementing automated industrial processes on a miniature scale before full-scale deployment. It consists of four industrial workstations[10]:

- 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.

The factory setup includes more than just workstations; it features a variety of specialized modules. These modules comprise a Delivery and Pickup Station (DPS), a Near-Field Communication (NFC) reader/writer, an RGB camera module, and a Sensor Station with Camera (SSC) designed to capture additional data. The production simulation involves moving small workpieces of different colors (white, red, and blue) throughout the facility. Each workpiece is equipped with a unique NFC identification tag, which logs its color and production history, including timestamps. This setup is managed by six Fischertechnik TXT controllers, as shown in Figure 2.

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

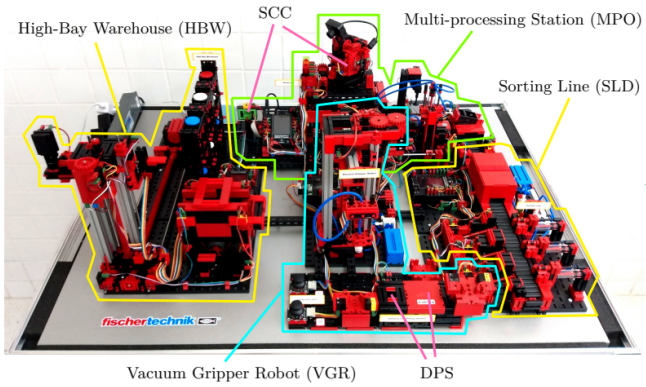

Figure 1: Fischertechnik Lernfabrik 4.0 9V, adopted from [10].

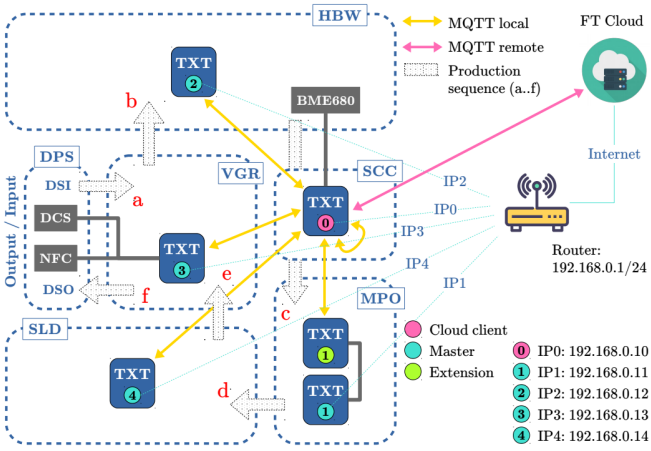

Figure 2: Block Diagram of the Lernfabrik 4.0, adopted from [10].

The bidirectional communication between the TXT controllers and the central TXT-0 is established through an MQTT broker, with TXT-0 serving this role for controllers connected to a Wi-Fi network via a TP-Link router. Configured as a cloud client, TXT-0 serves as an MQTT bridge connecting the entire system with the remote cloud. Users can access a cloud dashboard through a web browser to monitor and interact with specific factory production processes. As the primary cloud master, TXT-0 is equipped with a USB camera and environmental sensors to track air pressure, humidity, temperature, and gas levels. The TXT-1 to TXT-4 controllers also function as cloud masters, each with specific responsibilities: TXT-1 includes an extension, TXT-2 manages the warehouse, TXT-3 controls the DPS module, and TXT-4 oversees the SLD station. The block diagram depicts the standard manufacturing sequence from DSI to DSO,

with each TXT controller running compiled C++ code specific to its assigned station.

4 Reverse Shell Attacks in IIoT Systems

4.1 Reverse Shell Attacks

Reverse shell attacks are a potent method employed by cyber attackers to gain unauthorized access to a target system, enabling them to execute commands and control the compromised device remotely as depicted in figure 3.

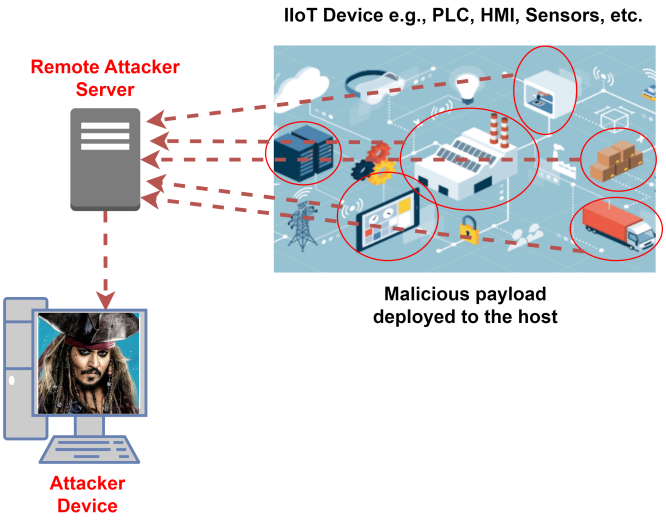

Figure 3: Reverse Shell Attack Scenario in IIoT systems

In the context of IIoT systems, which consist of interconnected networks of sensors, devices, and machinery used in industrial environments, reverse shell attacks pose a significant threat. They have the potential to disrupt critical operations, cause physical damage, and compromise sensitive data. Such attacks involve several critical stages that allow an attacker to gain unauthorized control over a target system. These stages include payload delivery, execution, command and control, and persistence. Each stage is essential for the successful implementation and maintenance of a reverse shell connection. Below, we detail the technical mechanisms involved in each stage:

- **Payload Delivery:** The attacker typically delivers a malicious payload to the target system through vectors such as phishing emails, compromised websites, or exploiting known vulnerabilities in software and firmware.

- **Execution:** Once the payload is executed on the target system, it establishes a connection back to a predefined IP address and port controlled by the attacker. This connection can be established using a variety of protocols such as TCP, UDP, or HTTP.

- **Command and Control:** Upon successful connection, the attacker gains a command shell on the compromised system, allowing them to execute commands and interact with the target environment as if they had direct access to the terminal. This can include commands to exfiltrate data, escalate privileges, or further compromise the system.

- **Persistence:** To maintain access and control over the compromised system, the attacker may deploy additional techniques to establish persistence, such as creating a backdoor or modifying system configurations to ensure the reverse shell connection is maintained even after system reboots.

4.2 Vulnerabilities Exploited in IIoT Systems for Reverse Shell Attacks

IIoT systems are characterized by extensive network connectivity and integration with Operational Technology (OT) infrastructure, making them susceptible to various vulnerabilities exploitable by malicious actors. Key vulnerabilities facilitating reverse shell attacks include:

4.2.1 ***Weak Authentication Mechanisms.*** Many IIoT devices and components utilize default or easily guessable credentials, such as factory-set usernames and passwords, or lack robust authentication mechanisms. Attackers can exploit these weak credentials to gain unauthorized access to IIoT devices and establish reverse shell connections.

4.2.2 ***Vulnerabilities in the SFTP Protocol.*** Weaknesses in the SFTP protocol can be exploited in several ways. Simple passwords, outdated cryptographic keys, and insufficient logging and monitoring capabilities make it easier for attackers to gain unauthorized access and inject malicious logic into transferred files. Vulnerabilities in the underlying SSH implementation, which SFTP relies on for security, can also be a target. Additionally, improper file and directory permissions may allow unauthorized access, and brute force attacks can compromise SFTP servers by guessing username and password combinations.

4.2.3 ***Inadequate Network Segmentation.*** Insufficient segmentation between IIoT devices and corporate networks allows attackers to pivot from less secure IIoT segments to more critical network segments. This facilitates the propagation of reverse shell malware and unauthorized access to sensitive systems.

4.2.4 ***Outdated Software and Firmware.*** IIoT devices often run outdated or unpatched software and firmware, which may contain known vulnerabilities that attackers can exploit to execute reverse shell attacks. Manufacturers may struggle to release timely security updates, leaving devices exposed to exploitation for extended periods.

4.2.5 ***Lack of Encryption.*** Inadequate or absent encryption mechanisms for data in transit between IIoT devices and backend systems expose sensitive information to interception and tampering by malicious actors. Attackers may exploit unencrypted communication channels to intercept authentication credentials and establish reverse shell connections.

4.2.6 ***Insufficient Access Controls.*** Poorly configured access controls and privilege management mechanisms may allow unauthorized users to escalate privileges and gain administrative access to IIoT devices, facilitating the installation and execution of reverse shell payloads.

5 Reverse Shell Attack Approach

Figure 4 provides a high-level overview of our reverse shell attack scenario that we conducted on our Fischertechnik system presented in figure 1. It consists of five primary phases:

- Breaking the authentication of the TP-Link Router
- Breaking the authentication of SFTP connection.
- Uploading reverse shell payload.
- Executing reverse shell payload to gain root user access.
- Performing control command injection.

In this work, we omit the detailed depiction of the first phase, which has been extensively covered in our previous paper [4]. Our emphasis in this paper is solely on phases 2, 3, 4 and 5, as elucidated in the following subsections.

5.1 Breaking the Authentication of SFTP Connection

TXT controllers, like many IIoT devices, have an SFTP interface that enables secure file transfers between the controller and remote users. These controllers simulate the functions of real hardware Programmable Logic Controllers (PLCs) such as Siemens S7-1500 PLCs, Rockwell Automation PLCs, and Beckhoff TwinCAT PLCs, which also use SFTP for file transfers over IIoT networks. In our study, we exploited the TXT-0 controller (see Figure 4) using FileZilla[3], an open-source software that creates a secure connection with the FTP server on the target TXT controller. This secure channel can be manipulated to transmit malicious executable files, like control logic programs, compromising the controller's integrity.

To carry out this attack, the attacker needs the target's IP address, username and password for SSH connection, and the port number (typically set to '22'). In our experiments, we used the default SSH credentials 'ROBOPro'. Even if different credentials are used, an attacker could retrieve them via a brute-force attack as shown in our previous work [4]. Once access is gained, the attacker can exploit the file upload functionality to upload a reverse shell payload to the controller.

5.2 Uploading Reverse Shell Payload

Attackers have three main options for preparing the payload: PHP, C/C++ binaries, and Python. While most embedded IIoT devices support PHP and Python, TXT controllers, which mainly manage actuator operations in ICS environments, might not support a full-fledged web server with PHP. Therefore, attackers are more likely to use Python or C/C++ binaries. Using Python's Berkeley sockets Application Programming Interface (API) to create a reverse shell is advantageous due to its simplicity and low overhead. In our Fischertechnik Learning Factory 4.0 setup, we successfully uploaded a python reverse shell payload to the TXT controller.

Once the reverse shell payload is uploaded and executed, it establishes an outbound connection to the attacker's remote server, bypassing network security measures that typically block incoming connections. This connection allows the attacker to execute commands remotely and maintain control over the TXT controller. The primary goal of the reverse shell is to ensure persistent access to the system, even if the attacker is no longer on the same network.

[3]https://filezilla-project.org/

The attacker sets up an external server to listen for connections from the reverse shell installed on the TXT controller. Tools like *ngrok*[4] can help to establish a persistent reverse shell connection to the TXT controller see figure 5.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ python server.py
Listening as 0.0.0.0:4444 ...
127.0.0.1:53162 Connected!
Enter the command you wanna execute:ls
TxtFactoryMain-Testing.cloud
TxtFactoryMain.cloud
TxtSmartHome.cloud
client.py
Enter the command you wanna execute:
```

Figure 5: Attacker's remote server is listening for the reverse shell connection.

This exploitation method not only allows the attacker to upload malicious files but also to extract critical data, such as control logic programs, files, and certificates from the compromised controller, significantly increasing the impact of the breach. For instance, we managed successfully to cause disruptions in system operations, resulting in inappropriate movements in the SSC (see [15]).

5.3 Executing the Shell Payload and Gaining Root Access

At this stage, the attacker has successfully uploaded the reverse shell payload to the TXT controller using the method described in section 5.2. The next step is to ensure the execution of this payload. There are several scenarios where an administrator might accidentally execute a python reverse shell payload, thereby unknowingly triggering a connection to the attacker's server. One potential scenario involves an administrator mistakenly executing a script during routine maintenance. For instance, if the administrator receives a phishing email masquerading as a system upgrade task, they might run the script without fully understanding its functionality. This action would inadvertently establish a reverse shell connection to an attacker-controlled server. Another scenario could occur during routine file management tasks. An administrator might unintentionally run a script that has been named similarly to a legitimate tool, resulting in the unintended execution of the reverse shell payload. In both cases, a lack of awareness or proper scrutiny of the script's origin and purpose could lead to the accidental activation of the reverse shell, thereby compromising the system's security.

To test this, we simulated a similar scenario in the Fischertechnik Learning Factory 4.0. When the reverse shell payload was executed, the attacker, operating from a different network, gained root shell access to the remote system (see [15]). This scenario highlights the potential for significant security breaches in IIoT systems due to seemingly minor oversights.

[4]https://ngrok.com/

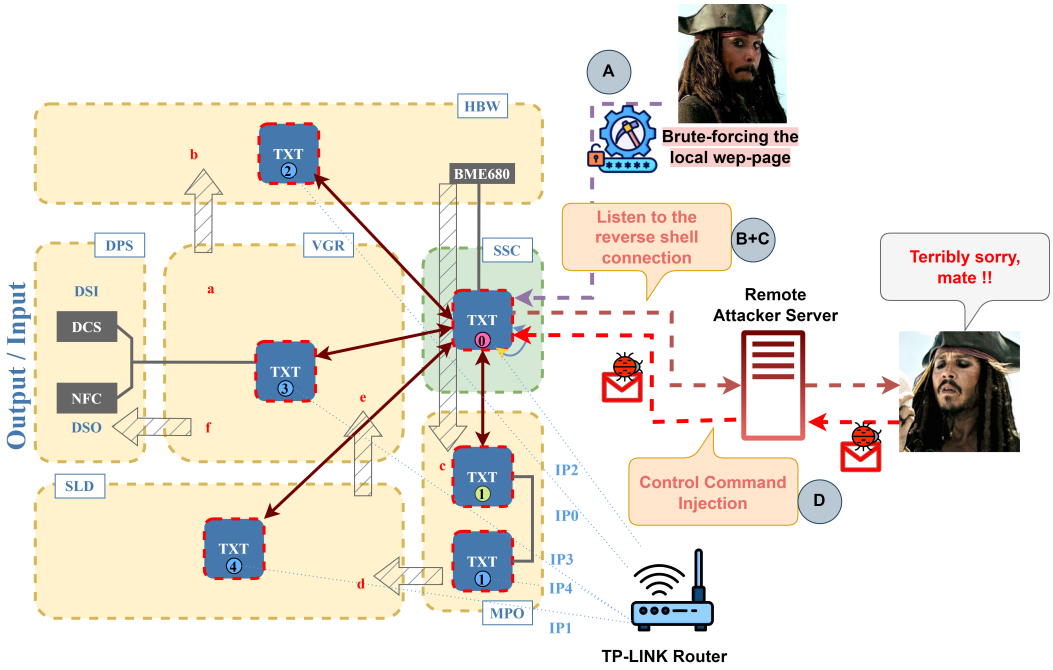

Figure 4: High-level overview of our Shell Reverse Attack.

5.4 Control Command Injection

In this phase, the attacker has gained complete control over the TXT as a root user. With root access, the attacker can execute any malicious programs on the TXT, significantly compromising the system's integrity. In many ICS and IIoT systems, actuators are controlled by calibration data files. For instance, in our Fischertechnik system (see figure 1), by modifying these calibration files, an attacker can alter the behavior of various components such as the SSC, HBW, VGR, DPS, MPO and SLD modules. To do so, we executed a so-called "TxtSmartHome.cloud" file without including the necessary calibration *JSON* file. This omission caused the system to operate with incorrect cycle counts, leading to a segmentation fault error that crashed the program as shown in figure 6. As a result of this crash, the camera module at the SSC began to move erratically, nearly detaching from the moving arm [15]. This malfunction illustrates a critical safety risk, as an attacker could intentionally manipulate the robotic arm to move in unpredictable and hazardous ways. Such behavior could lead to severe accidents, including equipment damage and potential injury to personnel within the factory environment.

6 Security Recommendations

To effectively mitigate reverse shell attacks on IIoT systems, a multifaceted approach to security is essential. Beyond basic connection encryption, robust security measures tailored to the distributed nature of IIoT hardware and software are imperative [11]. These measures should include:

- **Network Segmentation:** Implement network segmentation to compartmentalize IIoT devices and limit the scope of potential attacks.
- **Access Control:** Enforce strict access control policies, including strong authentication mechanisms and principle of least privilege principles, to restrict unauthorized access to IIoT devices and networks.
- **Regular Patching:** Maintain up-to-date firmware and software on IIoT devices to address known vulnerabilities and mitigate exploitation risks.
- **Behavioral Anomaly Detection:** Deploy behavioral anomaly detection mechanisms to identify unusual patterns of activity indicative of a reverse shell attack.
- **File Type Restrictions:** Allow only specific, authorized file types to be uploaded to IIoT devices to prevent the execution of malicious scripts or payloads.
- **File Verification:** Implement robust file verification mechanisms to ensure that uploaded files do not contain malicious code or disguised malware.
- **Malware Scanning:** Integrate malware scanning capabilities into IIoT systems to detect and quarantine any malicious files before they can be executed.

Furthermore, it is crucial to implement specific strategies to prevent the upload and execution of malicious payloads, including reverse shells [12]:

- **User Authentication:** Require user authentication before allowing file uploads to verify the identity and authorization of the user.
- **File Management Practices:** Adopt secure file management practices, such as renaming uploaded files and restricting their storage locations, to minimize the risk of exploitation.

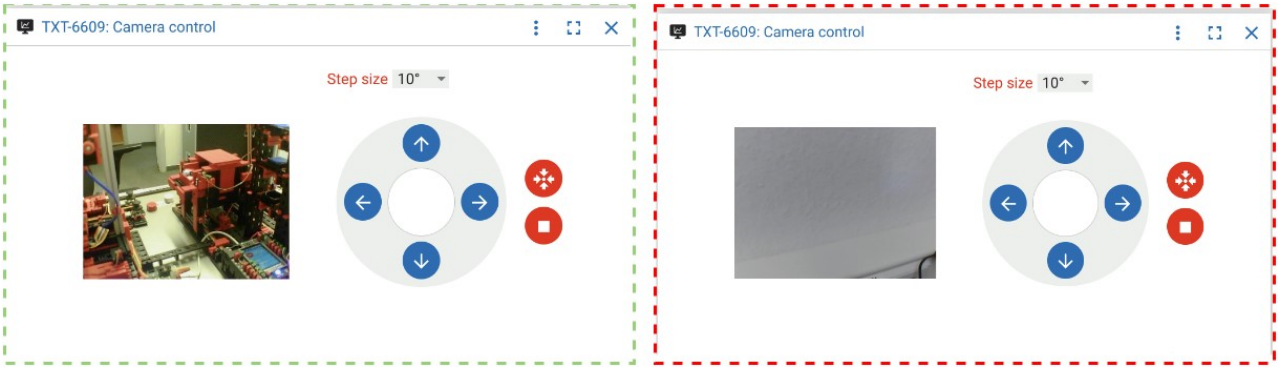

Figure 6: Dashboard shows the user the camera view of the Fischertechnik in two cases: 1) in normal operation (highlighted in green); 2) under control command Injection Attack based on our Reverse Shell Payload (highlighted in red)

Additionally, the deployment of Intrusion Detection Systems (IDS) [13, 14] can significantly enhance the security posture of IIoT systems by actively monitoring network traffic and identifying suspicious behavior indicative of a reverse shell attack.

While implementing these security measures adds complexity and incurs additional overhead in IIoT systems, the benefits far outweigh the drawbacks. Proactively addressing vulnerabilities and mitigating reverse shell attacks safeguards critical infrastructure and sensitive data. Investing in robust security measures enhances trust in IIoT deployments, fostering long-term resilience and reliability. Thus, careful consideration and resource allocation for these measures result in a more secure and resilient IIoT ecosystem.

7 Conclusion

In this paper, we explored vulnerabilities in IIoT systems that can be exploited for reverse shell attacks. Our investigations identified critical weaknesses, including weak authentication mechanisms, inadequate network segmentation, outdated software and firmware, lack of encryption, and insufficient access controls. These can be leveraged by attackers to gain unauthorized access, upload malicious payloads, and maintain persistent control over IIoT devices. Through a brute force attack, we demonstrated how easily an IIoT device's web authentication can be compromised, particularly with simple passwords. Once accessed, we exploited the SFTP protocol to deploy a reverse shell payload, enabling remote command execution and data exfiltration. Our experimental setup with the Fischertechnik Learning Factory 4.0 highlighted the practical implications, showing significant operational disruptions.

To mitigate these threats, IIoT system administrators should implement strong authentication protocols, regular updates, encrypted communications, and stringent access controls. Ongoing security audits and heightened awareness are also crucial to prevent the execution of malicious payloads. Addressing these vulnerabilities can significantly enhance the security of IIoT systems and protect critical industrial operations from sophisticated cyber threats.

References

[1] D. Serpanos, "Industrial Internet of Things: Trends and Challenges,"in Computer, vol. 57, no. 1, pp. 124-128, Jan. 2024, doi: 10.1109/MC.2023.3331552.

[2] F. Minna and F. Massacci, "SoK: Run-time security for cloud microservices. Are we there yet?," Computers & Security, Volume 127, 2023, doi: 10.1016/j.cose.2023.103119.

[3] 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.

[4] 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.

[5] M. Al-Hawawreh and E. Sitnikova, "Developing a Security Testbed for Industrial Internet of Things," in IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5558-5573, 1 April1, 2021, doi: 10.1109/JIOT.2020.3032093.

[6] C. Toma, C. Boja, M. Popa, M. Doinea, and C. Ciurea, "Viruses, Exploits, Malware and Security Issues on IoT Devices," In: Ryan, P.Y., Toma, C. (eds) Innovative Security Solutions for Information Technology and Communications. SecITC 2021. Lecture Notes in Computer Science, vol 13195. Springer, Cham, doi: 10.1007/978-3-031-17510-7_22.

[7] A. Manglani, T. Desai, P. Shah and V. Ukani, "Optimized Reverse TCP Shell Using One-Time Persistent Connection," In: Singh, P.K., Polkowski, Z., Tanwar, S., Pandey, S.K., Matei, G., Pirvu, D. (eds) Innovations in Information and Communication Technologies (IICT-2020). Advances in Science, Technology & Innovation. Springer, Cham. doi: 10.1007/978-3-030-66218-9_41.

[8] K. Kaushik, S. Aggarwal, Sh. Mudgal, and V. Mathur, "A novel approach to generate a reverse shell: Exploitation and Prevention," International Journal of Intelligent Communication, Computing, and Networks. 2. 83-93, doi: 10.51735/ijiccn/001/33.

[9] [Online]. Available: https://www.techtarget.com/searchsecurity/tip/What-reverse-shell-attacks-are-and-how-to-prevent-them.

[10] 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), Sao Bernardo do Campo, Brazil, 2022, pp. 229-234, doi: 10.1109/LARS/SBR/WRE56824.2022.9995868.

[11] A. M. Alnajim, S. Habib, M. Islam, S. M. Thwin and F. A. Alotaibi, "A Comprehensive Survey of Cybersecurity Threats, Attacks, and Effective Countermeasures in Industrial Internet of Things," Technologies 2023, 11, 161, doi: 10.3390/technologies11060161.

[12] K. Kaushik, S. Aggarwal, S. Mudgal, S. Saravgi and V. Mathur, "A novel approach to generate a reverse shell: Exploitation and Prevention", Int. J. Intell. Commun. Comput. Networks Open Access J., pp. 2582-7707.

[13] D. S. Panwar, J. Parashar, A. Jain, L. Meena, and S. Kapoor, "A Study on Reverse Bind Shells: Techniques, Advantages and Security Measures," JOURNAL OF INTELLIGENT SYSTEMS AND COMPUTING, 4(1), 17–26. [Online]. Available: https://scienceandtech.co.uk/journals/index.php/jiscom/article/view/35.

[14] M. Al-Hawawreh and E. Sitnikova, "Developing a Security Testbed for Industrial Internet of Things," in IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5558-5573, 1 April1, 2021, doi: 10.1109/JIOT.2020.3032093.

[15] [Online]. Available: https://www.youtube.com/watch?v=u8nhV5LXjJU

[16] [Online]. Available: https://github.com/rnrn0909/hack-the-backbone.git