Testing and assessing the security of the MQTT protocol in IoT environments under simulated cyberattacks*

Maryna Derkach[1,*,†], Ivan Horyn[1,†], Inna Skarga-Bandurova[1,2,†], and Yuriy Skorenkyy[1,†]

[1] Ternopil Ivan Puluj National Technical University, 56 Ruska str., 46025 Ternopil, Ukraine
[2] Oxford Brookes University, Headington Rd, Headington, Oxford, OX3 0BP, Oxford, UK

Abstract

The rapid development of the Internet of Things (IoT) has significantly increased the demand for secure and reliable data exchange protocols. MQTT (Message Queuing Telemetry Transport) is one of the most widely used lightweight messaging protocols in IoT environments due to its simplicity and low bandwidth consumption. However, its minimalist design introduces several security challenges in practical deployments. This paper presents a security testing of the MQTT protocol, including its architecture, vulnerabilities, and protection mechanisms. Experimental studies based on real-world attack scenarios, such as man-in-the-middle attacks, denial-of-service attacks, and unauthorized subscriptions, were conducted. The assessment results confirm the high vulnerability of unsecured MQTT deployments and demonstrate the effectiveness of cryptographic protection, authentication, and access control mechanisms. The proposed approach can be applied to enhance the security of IoT systems.

Keywords

MQTT, IoT, communication protocols, penetration testing, security, cyberattacks, cryptographic protection.

1. Introduction

The rapid growth of Internet of Things (IoT) technologies is transforming modern digital ecosystems by enabling large-scale data collection, automation, and real-time monitoring across domains such as smart cities, healthcare, industrial automation, and transportation [1-3]. These systems require lightweight communication protocols capable of operating in reliable resource-constrained with network environments [4, 5].

One of the most widely adopted protocols in this context is MQTT, which is valued for its simplicity and scalability [6-8]. However, its design prioritizes performance and minimalism design than built-in security mechanisms. IoT communication security is actively researched due to the growing number of connected devices and the increasing scale of cyberattacks. For example, this study [9] investigates a wide range of MQTT vulnerabilities, including replay attacks, man-in-the-middle (MiTM) attacks, denial-of-service (DoS/DDoS) attacks, and proposes a training "cyber range" platform for evaluating and testing MQTT security, highlights the critical importance of strengthening protection and serving as a guideline for further research in this field. Consequently, subsequent studies have focused on device authentication, lightweight cryptographic algorithms, and secure key management. Among the three major security threats — information leakage, weak authentication, and denial of service — the authors of [10] provided quantitative indicators of the immaturity of the IoT ecosystem, showing that 9.44% of backends disclose information and that 99.84% of MQTT and XMPP backends use insecure transport protocols (only 0.16% use transport-layer security (TLS), of which 70.93% rely on vulnerable versions), indicating a significant real-world vulnerability to passive and active network threats [11]. These findings provide evidence of

* CSDP'2026: Cyber Security and Data Protection, May 7, 2026, Lviv, Ukraine
* Corresponding author.
† These authors contributed equally.
m_derkach@tntu.edu.ua (M. Derkach); ivan.horyn.0@gmail.com (I. Horyn); iskarga-bandurova@brookes.ac.uk (I. Skarga-Bandurova); skorenkyy@tntu.edu.ua (Y. Skorenkyy)
0000-0001-8977-2776 (M. Derkach); 0009-0003-1235-3962 (I. Horyn); 0000-0003-3458-8730 (I. Skarga-Bandurova); 0000-0002-4809-9025 (Y. Skorenkyy)

widespread security misconfigurations in real-world deployments, which is a crucial factor in assessing the threat landscape for MQTT in IoT communications.

As a result, researchers began to analyze MQTT architecture and its potential vulnerabilities. The study [12] presents a comprehensive review of authentication mechanisms in MQTT, covering approaches ranging from basic username/password authentication to advanced methods such as client certificates, OAuth 2.0 integration, and token-based authentication, as well as recent advances in broker implementations. It has also been demonstrated that the MQTT broker represents a critical point of failure, as its compromise may lead to complete control over communications. Furthermore, the lack of mandatory encryption and authentication enables message interception and modification. Although the use of TLS, token-based authentication, and advanced access control mechanisms is recommended, these solutions may introduce performance overhead and implementation complexity in constrained environments. The authors [13] proposed a scheme based on the Mosquitto MQTT broker to provide mutual authentication at the transport layer in IoT environments, reducing the number of messages exchanged between devices, which is particularly important in bandwidth-constrained networks. Meanwhile, the work [14] provides a practical analysis of the security-performance trade-off by empirically evaluating the feasibility of TLS 1.3-secure MQTT communication on constrained hardware using modern cipher suites. The authors quantify CPU usage, latency, power consumption, and throughput, demonstrating that although security introduces overhead, secure MQTT remains feasible even for performance-sensitive environments. Consequently, achieving an optimal balance between security and performance in MQTT-based systems remains a significant challenge, as such systems must ensure robust cryptographic protection, authentication, and access control to prevent not only data leakage but also disruptions to physical processes and critical services.

The aim of this study is to perform a security testing of the MQTT protocol in IoT environments, which includes identifying its vulnerabilities and assessing the effectiveness of modern protection mechanisms by simulating cyberattacks. The research analyses the protocol architecture, key vulnerabilities, and the effectiveness of modern protection mechanisms through experimental testing in a controlled laboratory environment.

2. MQTT architecture and key vulnerabilities

MQTT is a lightweight publish/subscribe messaging protocol designed for resource-constrained devices and low-bandwidth networks. Its architecture consists of publishers, subscribers, and a broker that mediates communication between them [15]. Clients, which may act as publishers, subscribers, or both, establish connections with a broker.

Publishers transmit messages associated with specific topic, while subscribers register their interest in selected topics with the broker. The broker filters incoming messages and forwards them to all clients subscribed to the corresponding topics, enabling scalable and flexible communication without requiring participants to know each other's network location or operational status. The protocol supports three qualities of service (QoS) levels that determine message delivery reliability but does not provide built-in mechanisms for ensuring data confidentiality, integrity, or authenticity [16]. In most deployments, security is therefore implemented through external mechanisms such as TLS.

By default, MQTT permits anonymous connections and does not enforce topic-level access control, which may result in unauthorized access, data leakage, and the injection of malicious commands into the system.

Table 1 shows analysis of key MQTT vulnerabilities and their potential exploits.

Table 1
MQTT protocol vulnerability analysis

Category	Vulnerability	Description	Potential Impact
Authentication / Authorisation	Lack of authentication	The standard does not require authentication; anonymous connections are allowed.	Unauthorised access to the broker, resource exhaustion, anonymous malicious activities.
	Weak / default credentials	Use of guessable, hard-coded, or default passwords.	Client compromise, unauthorised access/publication of data, device impersonation.
	Lack of granular authorisation	No topic-level access control; all access is allowed by default.	Unauthorised data access/injection, device takeover.
	Unrestricted wildcard usage	Allowing # and + wildcards in subscriptions without proper restrictions.	Mass data leakage, client overload, increased broker load.
Confidentiality	Plaintext transmission	Message payloads are not encrypted by default.	Interception and eavesdropping of sensitive data (passwords, commands, telemetry).
	Misconfigured TLS	TLS not used, weak ciphers, unvalidated certificates.	Man-in-the-middle (MitM) attacks, interception of encrypted traffic.
Availability	Open ports (especially 1883)	Public exposure of MQTT ports without firewalls or authentication.	Easy discovery by scanners, high susceptibility to DoS and brute-force attacks.
	DoS attack (CONNECT flood)	Flooding the broker with CONNECT packets to exhaust resources.	Broker resource exhaustion (CPU, memory), denial of service to legitimate clients.
	DoS attack (PUBLISH flood)	Flooding the broker with PUBLISH messages to overload the system.	Broker resource exhaustion, network congestion, subscriber overload.
	Low-rate DoS attacks	Exploiting protocol logic to cause denial of service with minimal traffic.	Broker disruption, harder detection compared to flood attacks.
Data Integrity	Lack of integrity verification	The protocol lacks built-in mechanisms to ensure data integrity.	Injection of malicious/corrupted data, incorrect system behaviour.

3. Methodology and experimental environment

To evaluate common attack vectors against MQTT and assess the effectiveness of various security mechanisms, an isolated experimental environment simulating an IoT infrastructure was developed. The infrastructure was deployed using Docker containerization on a virtual machine running Ubuntu 24.04.2 LTS [17-19]. All components were integrated into a dedicated Docker network (mqtt-net) with the subnet 172.20.0.0/24, enabling controlled network interactions. This deployment approach flexibility, reproducibility, and isolation of test components, including the Mosquitto MQTT broker, Python-based clients (publisher/subscriber), and penetration testing tools such as Metasploit Framework and Wireshark [20]. Three MQTT broker configurations were sequentially evaluated:

1. A default configuration without encryption and authentication.
2. A password-based authentication configuration.
3. A security configuration with encrypted communication and authentication.

Each configuration was subjected to attacks including passive traffic interception, denial-of-service (DoS) attacks, and man-in-the-middle (MiTM) attacks. The experimental setup enables a comparative assessment of the security level provided by each configuration.

3.1. A default configuration

In the first stage, the Mosquitto MQTT broker was deployed using default settings: open port 1883, no authentication, no access control lists (ACLs), no restrictions on connections or message publishing, and unencrypted traffic. The most basic yet critical vulnerability of this configuration is the ability for any network participant to intercept and read transmitted data. Using a simple eavesdropper script (mqtt-eavesdropper), traffic interception was demonstrated, revealing sensitive telemetry data (e.g., temperature, humidity) and metadata (sensor ID, location) in plaintext (Figure 1).

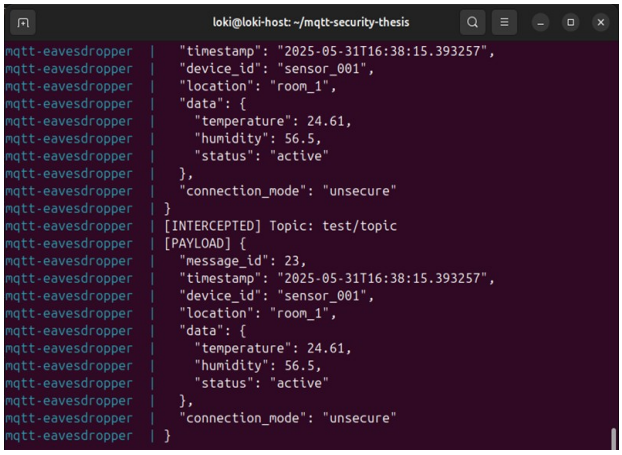

Figure 1: Interception of unencrypted MQTT traffic by an eavesdropper.

Network traffic analysis with Wireshark confirmed that MQTT transmits all information, including topics and payloads, without encryption, creating a serious risk of confidential data leakage [21]. The Metasploit mqtt_login module was then used to scan the test broker, which detected the open port 1883 and confirmed that unauthorized access was possible (Anonymous access ALLOWED). A denial-of-service (DoS) attack was launched using the Metasploit mqtt_resource_exhaustion module. As a result, the broker's memory and more than 87% of the container's allocated CPU resources were exhausted. Broker logs confirmed service termination due to memory exhaustion, causing the legitimate publisher to lose connectivity and resulting in a complete denial of service. The MiTM attack further demonstrates not only traffic interception but also message manipulation. Using ARP spoofing, traffic between the publisher and the broker was redirected through the attacker's host, where a specialized script intercepted MQTT packets, altered payload, and forwarded compromised data, demonstrating the feasibility of undetected data manipulation under this configuration [22,23].

3.2. A password-based authentication configuration

At this stage, the Mosquitto broker was configured to require username and password authentication for all clients [24]. Attempted DoS attacks against the password-protected broker were unsuccessful, as the broker rejected unauthenticated connections attempts without consuming significant resources [25]. As a result, resource utilization remained low and configuration stability was preserved. However, this protection remains vulnerable to brute-force attacks when weak credentials are used. Using Metasploit with a dictionary of common usernames and passwords, authentication credentials were successfully obtained (SUCCESS: testuser:testpass), as shown in Figure 2, highlighting the importance of strong and unique passwords.

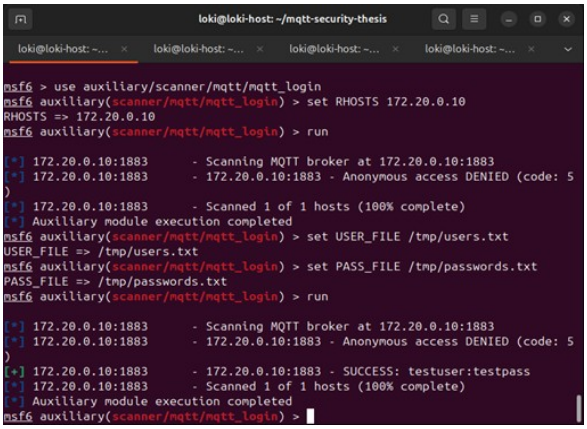

Figure 2: Successful credential brute-force attack using Metasploit.

Importantly, despite the authentication, traffic was still transmitted in plaintext, leaving the configuration susceptible to MiTM attacks. Traffic analysis using Wireshark confirmed that the data remained unencrypted. An attacker on the same network could intercept authenticated session and inject malicious data through session hijacking and message modification.

3.3. A secure configuration with encrypted communication and authentication

The final test employed a multi-layered security configuration. The broker accepted connections on port 8883, used TLS to encrypt the communication channel, and required both login authentication and a valid client certificate, implementing mutual authentication (mTLS) [26]. Under this configuration, automated Metasploit scanning and DoS attacks were unsuccessful, as the attacking clients were unable to establish a connection to the broker due to the absence of valid TLS certificates required to complete the TLS handshake process. Meanwhile, the broker logs recorded numerous errors, indicating that unauthorized connection attempts were successfully blocked at an early stage, while resource consumption on the broker remained minimal. Owing to TLS encryption, MiTM attempts did not result in readable or modifiable payloads due to TLS encryption [27]. As shown in Figure 3, although an attacker can intercept the traffic, it cannot be read or modified.

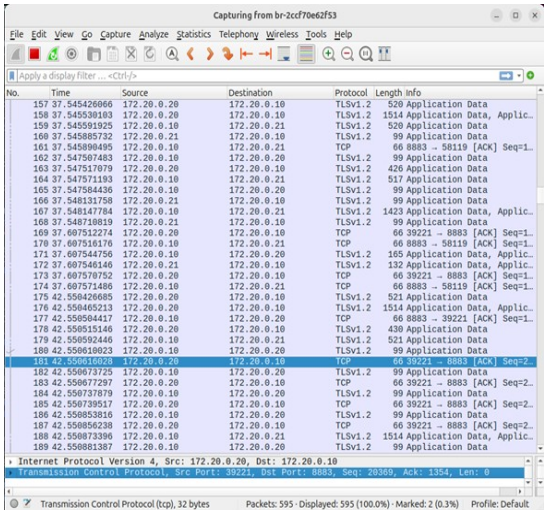

Figure 3: Monitoring encrypted traffic in Wireshark.

Traffic analysis in Wireshark confirms that only encrypted packets (Application Data) are transmitted instead of plaintext data. Mutual authentication further protects against spoofing attacks by ensuring that both the client and the broker verify each other's certificates.

4. Results

The results obtained across the evaluated configurations reveal clear differences in attack resilience and broker stability as security mechanisms are progressively applied. Experimental results were assessed in terms of attack success rates, message delivery latency, and broker stability. Due to the centralized MQTT architecture, the broker represents a single point of failure. A combined attack targeting connection pool exhaustion (SlowITe DoS), memory consumption (message flooding), and CPU utilization was launched [28]. Resource consumption statistics for the Docker container are presented in Table 2.

Table 2.
Broker resource utilization statistics in Docker.

Configuration	CPU, %	Memory, %
Default configuration	87.59	100
Password-based authentication configuration	1.55	0.08
Secure configuration (TLS + mTLS)	5.49	0.43

Under the default MQTT configuration, the attack resulted in severe service degradation. Docker container statistics showed CPU utilization reaching 87.59%, while the allocated memory (1 GB) was fully exhausted, resulting in a complete service outage and preventing legitimate clients from publishing data. In contrast, under the password-based authentication configuration, the DoS attack was ineffective. The broker rejected unauthenticated connections, and resource consumption remained minimal (CPU load 1.55%, allocated memory usage 0.08%). Conversely, enforcing authentication and cryptographic protection limited attacker interaction at early stages of communication and significantly reduced the impact of resource-exhaustion attempts. In our case, the broker immediately terminated the connection, the attacker could not complete the TLS handshake without a valid client certificate, and resource consumption remained negligible (CPU load 5.49%, allocated memory usage 0.43%).

The observed attack outcomes reflect three recurring weaknesses in unsecured MQTT deployments:

1. The use of default IoT device configurations, including weak or unchangeable passwords enables successful credential-guessing attacks in the password-based configuration, as demonstrated in the password-based configuration.
2. The absence of transport-layer encryption on smart devices, allows full disclosure and manipulation of MQTT messages, exposing sensitive telemetry and control data.
3. Limited defensive mechanisms at the broker level increase susceptibility to DoS attacks that target connection handling and resource allocation.

These shortcomings are typical of unsecured MQTT implementations, when combined, substantially increase the attack surface of MQTT-based IoT systems. The results indicate that a layered security approach combining encryption and mutual authentication offers significantly stronger resilience against the evaluated attack scenarios under the tested conditions.

5. Conclusions

This study evaluated the security of the MQTT deployments under different protection mechanisms in an IoT-oriented environment. The results demonstrate that default MQTT configurations expose systems to severe security risks, including traffic interception, message manipulation, unauthorized access, and denial-of-service attacks. Introducing password-based authentication mitigates some attack vectors by restricting unauthenticated access but does not provide confidentiality or integrity protection and remains vulnerable to credential compromise. The strongest protection was observed when transport-layer encryption and mutual authentication

were applied. TLS ensured confidentiality and integrity of transmitted data, while client certificate validation significantly reduced the feasibility of unauthorized access, spoofing, and resource-exhaustion attacks. These findings indicate that, at a minimum, authentication should be enforced in MQTT deployments, while TLS combined with mutual authentication provides substantially stronger protection for real-world IoT systems.

Declaration on Generative AI

Portions of this manuscript were prepared with the assistance of AI-based language tools, including OpenAI's ChatGPT and Grammarly, for tasks such as editing, grammar correction, and improving technical writing. All scientific content, architectural design, experimental methodology, and analysis were implemented, and validated by the authors. The AI was not used to generate data, conduct experiments, or formulate original research ideas.

References

[1] I. Ullah, Q. H. Mahmoud, Design and Development of RNN Anomaly Detection Model for IoT Networks, IEEE Access 10 (2022) 62722–62750. doi:10.1109/ACCESS.2022.3176317

[2] J. J. Sellers, R. J. Astore, et al, Understanding Space: An Introduction to Astronautics, 2nd Edition, McGraw-Hill Primis Custom Pub; 2nd edition, 2000.I. Hamid, M. M. H. Rahman, AI, Machine Learning and Deep Learning in Cyber Risk Management. Discov. Sustain. 6, 389 (2025). doi:10.1007/s43621-025-01012-3

[3] R. C. Poonia, et al., Real-Time Cyber-Physical Risk Management Leveraging Advanced Security Technologies, in: Yang, XS., Sherratt, S., Dey, N., Joshi, A. (eds) Proceedings of 9th Int. Congress on Information and Communication Technology. ICICT. Lecture Notes in Networks and Systems, vol 1011, 2024. Springer. doi:10.1007/978-981-97-4581-4_25

[4] M. Mylrea, et al., Insider Threat Cybersecurity Framework Webtool & Methodology: Defending against Complex Cyber-Physical Threats, in: 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2018, 207–216. doi:10.1109/SPW.2018.00036

[5] W. Chua, et al., Web Traffic Anomaly Detection using Isolation Forest. Informatics, 11(4), 83. doi:10.3390/informatics11040083

[6] S. Tamy, et al., An Evaluation of Machine Learning Algorithms to Detect Attacks in Scada Network, in: 2019 7th Mediterranean Congress of Telecommunications (CMT), Fez, Morocco, 2019, 1–5. doi:10.1109/CMT.2019.8931327

[7] J. Feng, et al., Tensor Recurrent Neural Network with Differential Privacy, IEEE Trans. Comput. 73(3) (2024) 683–693. doi:10.1109/TC.2023.3236868

[8] S. Kerimkhulle, et al, Fuzzy Logic and Its Application in the Assessment of Information Security Risk of Industrial Internet of Things. Symmetry 15(10) (2023) 1958. doi:10.3390/sym15101958

[9] Y. Kostiuk, et al., Tools for Providing Information Security from Hidden Threats in Cloud Computing Infrastructure. Cybersecur. Edu. Sci. Technol. 4(28) (2025) 633–655. doi:10.28925/2663-4023.2025.28.857

[10] K. Al-Dosari, N. Fetais, Risk-Management Framework and Information-Security Systems for Small and Medium Enterprises (SMEs): A Meta-Analysis Approach. Electronics, 12(17) (2023) 3629. doi:10.3390/electronics12173629

[11] A. Yayla, L. Haghnegahdar, E. Dincelli, Explainable Artificial Intelligence for Smart Grid Intrusion Detection Systems, IT Prof. 24(5) (2022) 18–24. doi:10.1109/MITP.2022.3163731

[12] M. Nakip, E. Gelenbe, Online Self-Supervised Deep Learning for Intrusion Detection Systems, IEEE Trans. Inf. Forensics Secur. 19 (2024) 5668–5683. doi:10.1109/TIFS.2024.3402148

[13] A. Vedant, et al., Detecting Cyber Attacks in a Cyber-physical Power System: A Machine Learning Based Approach, in: 2022 Global Energy Conference (GEC), Batman, Turkey, 2022, 272–277, doi:10.1109/GEC55014.2022.9986990

[14] F. Alzhouri, et al., A Smart Network Intrusion Detection System for Cyber Security of Industrial IoT, in: 2023 4th Int. Conf. on Intelligent Data Science Technologies and Applications (IDSTA), Kuwai, Kuwait, 2023, 67–75. doi:10.1109/IDSTA58916.2023.10317861

[15] A. Dhakad, et al., Real Time Network Traffic Analysis using Artificial Intelligence, Machine Learning and Deep Learning: A Review of Methods, Tools and Applications, in: 2023 Int. Conf. on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2023, 372–378. doi:10.1109/ICSSAS57918.2023.10331855

[16] R. Sinha, P. Agrawal, A. Rasool, Hyperparameter Tuned Cloud Based Cyber Physical Attack Detection using Stacking Ensemble Learning, in: 2024 IEEE Int. Students' Conf. on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2024, 1–6. doi:10.1109/SCEECS61402.2024.10482067

[17] H. S. Divyashree, et al., Enhanced DDoS Attack Detection in Cyber-Physical Systems Within Supply Chain 4.0 using Optimized TSO-DBN Deep Learning Technique, in: 2024 IEEE Int. Women in Engineering (WIE) Conf. on Electrical and Computer Engineering (WIECON-ECE), Chennai, India, 2024, 151–156. doi:10.1109/WIECON-ECE64149.2024.10914761

[18] T. Roberts, S. J. Tonna, Artificial Intelligence, Machine Learning, and Deep Learning Models for Risk Management, in: Risk Modeling: Practical Applications of Artificial Intelligence, Machine Learning, and Deep Learning , Wiley, 2022, 31–54. doi:10.1002/9781119824961.ch3

[19] S. Ankalaki, et al., Cyber Attack Prediction: From Traditional Machine Learning to Generative Artificial Intelligence, IEEE Access 13 (2025) 44662–44706. doi:10.1109/ACCESS.2025.3547433

[20] I. Opirskyy, et al., Detection of Network Attacks based on the RedHunt Virtual Machine, Adv. Cybersecur, 1st ed., 2025, 310–330.

[21] P. Skladannyi, et al., Development of Modular Neural Networks for Detecting Different Classes of Network Attacks. Cybersecur. Edu. Sci. Tech. 3(27) (2025) 534–548. doi:10.28925/2663-4023.2025.27.772

[22] V. P. M. R, V. S. Vardhan, S. S, V. k. K, M. M. R. G, AI-Driven Cyber Threat Detection and Log Analysis, in: 2025 Int. Conf. on Inventive Computation Technologies (ICICT), Kirtipur, Nepal, 2025, 676–681. doi:10.1109/ICICT64420.2025.11004938

[23] Y. Kostiuk, et al., Models and Technologies of Cognitive Agents for Decision-Making with Integration of Artificial Intelligence, in: Modern Data Science Technologies Doctoral Consortium (MoDaST 2025), vol. 4005, 2025, 82–96.

[24] O. Harasymchuk, et al., Intelligent Cyber Defence Systems: Detection of Ransomware and Protection of Wireless Networks based on Artificial Intelligence Technologies, Technology Center PC, 2025. doi:10.15587/978-617-8360-22-1

[25] O. Harasymchuk, et al., Modern Methods of Ensuring Information Protection in Cybersecurity Systems using Artificial Intelligence and Blockchain Technology, Technology Center PC, 2025. doi:10.15587/978-617-8360-12-2

[26] Y. Kostiuk, et al., Effectiveness of Information Security Control using Audit Logs, in: Cybersecurity Providing in Information and Telecommunication Systems (CPITS), vol. 3991, 2025, 524–538.

[27] E. Caville, W. W. Lo, S. Layeghy, M. Portmann, Anomal-E: A Self-Supervised Network Intrusion Detection System based on Graph Neural Networks, Knowl.-based Syst. 258 (2022) 110030.

[28] Y. Kostiuk, et al., Integrated Protection Strategies and Adaptive Resource Distribution for Secure Video Streaming over a Bluetooth Network, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3826 (2024) 129–138.

[29] V. S. S. R. Nallapareddy, Modified AI-based Learning Approach to Secure Data using Deep Learning Algorithms, in: 2024 Int. Conf. on Intelligent Computing and Emerging Communication Technologies (ICEC), Guntur, India, 2024, 1–7. doi:10.1109/ICEC59683.2024.10837269

[30] S. Vasylyshyn, et al., A Model of Decoy System based on Dynamic Attributes for Cybercrime Investigation, East.-Eur. J. Enterpr. Technol. 1(121) (2023) 6–20. doi:10.15587/1729-4061.2023.273363

[31] Y. Kostiuk, et al., Ensuring Cybersecurity and High-Speed Data Transmission in Wireless Networks, Inf. Secur. 30(3) (2024) 365–375.

[32] Y. Kostiuk, et al., Intelligent System for Simulation Modeling and Research of Information Objects, in: 1st Workshop Software Engineering and Semantic Technologies (SEST), vol. 4053 (2025) 237–251.

[33] P. Skladannyi, et al., Model and Methodology for the Formation of Adaptive Security Profiles for the Protection of Wireless Networks in the Face of Dynamic Cyber Threats, in: Workshop Cyber Security and Data Protection (CSDP), vol. 4042 (2025) 17–36.

[34] J. Hutchins, D. Ferrer, J. Fillers, C. Schuman, Black-Box Adversarial Attacks on Spiking Neural Network for Time Series Data, in: Int. Conf. on Neuromorphic Systems (ICONS), Arlington, VA, USA, 2024, 229–233. doi:10.1109/ICONS62911.2024.00040

[35] O. Partyka, A. Partyka, A. L. Imoize, AI for Web Application Security. Handbook of Cybersecurity Challenges and Solutions for Emerging Technologies (2026). doi:10.1201/9781003640790-14

[36] I. Opirskyy, O. Partyka, A. Partyka, A. L. Imoize, AI for Wireless Network Protection. Handbook of Cybersecurity Challenges and Solutions for Emerging Technologies (2026). doi:10.1201/9781003640790-15