network

Article

Signature-Based Security Analysis and Detection of IoT Threats in Advanced Message Queuing Protocol

Mohammad Emran Hashimyar, Mahdi Aiash *, Ali Khoshkholghi and Giacomo Nalli

Department of Computer Science, Middlesex University, London NW4 4BT, UK;
mh1492@live.mdx.ac.uk (M.E.H.); a.khoshkholghi@mdx.ac.uk (A.K.); g.nalli@mdx.ac.uk (G.N.)
* Correspondence: m.aiash@mdx.ac.uk; Tel.: +44-(0)20-8411-5220

Abstract: The Advanced Message Queuing Protocol (AMQP) is a widely used communication standard in IoT systems due to its robust and reliable message delivery capabilities. However, its increasing adoption has made it a target for various cyber threats, including Distributed Denial of Service (DDoS), Man-in-the-Middle (MitM), and brute force attacks. This study presents a comprehensive analysis of AMQP-specific vulnerabilities and introduces a statistical model for the detection and classification of malicious activities in IoT networks. Leveraging a custom-designed IoT testbed, realistic attack scenarios were simulated, and a dataset encompassing normal, malicious, and mixed traffic was generated. Unique attack signatures were identified and validated through repeated experiments, forming the foundation of a signature-based detection mechanism tailored for AMQP networks. The proposed model demonstrated high accuracy in detecting and classifying attack-specific traffic while maintaining a low false positive rate for benign traffic. Notable results include effective detection of RST packets in DDoS scenarios, precise classification of MitM attack patterns, and identification of brute force attempts on AMQP systems. This research highlights the efficacy of signature-based approaches in enhancing IoT security and offers a benchmark for future machine learning-driven detection systems. By addressing AMQP-specific challenges, the study contributes to the development of resilient and secure IoT ecosystems.

Keywords: AMQP; IoT; traffic; signature; cyberattacks

check for updates

Academic Editor: Chin-Tser Huang

Received: 2 January 2025
Revised: 2 February 2025
Accepted: 7 February 2025
Published: 17 February 2025

Citation: Hashimyar, M.E.; Aiash, M.; Khoshkholghi, A.; Nalli, G. Signature-Based Security Analysis and Detection of IoT Threats in Advanced Message Queuing Protocol. *Network* **2025**, *5*, 5. https://doi.org/10.3390/network5010005

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Internet of Things (IoT) has transformed modern industries, enhancing efficiency in smart homes, healthcare, transportation, and critical infrastructure. At the core of IoT communication are application-layer protocols that enable seamless interactions between devices. Among these, the Advanced Message Queuing Protocol (AMQP) has emerged as a key messaging standard due to its reliable message queuing, routing, and delivery capabilities [1]. AMQP is widely used in smart environments, industrial automation, and healthcare applications to support real-time exchange of sensor data, control commands, and event notifications. Its ability to ensure structured message queuing and guaranteed delivery makes it a preferred choice for IoT ecosystems. However, this central role in IoT communication also makes AMQP an attractive target for cyber threats. Attackers exploiting AMQP vulnerabilities can intercept, manipulate, or disrupt IoT messages, leading to data breaches, operational failures, and security risks. For example, a compromised industrial automation system could receive altered control signals, causing unsafe operations, while a breach in a healthcare network could lead to delayed or falsified medical alerts, endangering patient safety. Additionally, unsecured AMQP deployments are susceptible to Denial-of-Service (DoS) attacks, which can overload IoT devices and disrupt network functionality. As IoT adoption grows, ensuring AMQP security is critical to maintaining data integrity, availability, and confidentiality across connected systems.

Despite its advantages, AMQP introduces significant security challenges. Io T devices utilizing AMQP are increasingly targeted by cyberattacks due to inherent vulnerabilities in the protocol. These vulnerabilities have led to various documented attacks, including the following:

- Distributed Denial of Service (DDoS): Overwhelming a system with traffic from compromised devices, causing service disruptions (e.g., CVE-2021-22116) [2].
- Man-in-the-Middle (MitM): Intercepting communications between devices to eavesdrop or manipulate data (e.g., CVE-2019-3845, CVE-2018-11087) [3].
- Brute Force: Systematic attempts to guess login credentials, leading to unauthorized access (e.g., CVE-2023-24448) [4].

These attacks pose serious threats to the reliability and security of IoT systems, undermining their functionality and trustworthiness. However, research into AMQP-specific vulnerabilities remains limited. Current studies often generalize IoT security issues or focus on other protocols, such as MQTT or CoAP, leaving AMQP underexplored. This has resulted in several gaps in the literature:

- Limited Focus on AMQP-Specific Threats: Existing research predominantly addresses broader IoT security challenges, with insufficient emphasis on the unique vulnerabilities of AMQP [5].
- Outdated and Narrow Datasets: Most datasets do not reflect recent attack methodologies or encompass the full range of AMQP-specific vulnerabilities. Some datasets focus on older attack types, such as IP sweeping and smurfing [6,7].
- Theoretical Models Over Practical Scenarios: Many studies rely on analytical approaches, failing to account for the complexity of real-world IoT environments [5].
- Inadequate Cross-Protocol Considerations: Research often isolates protocols, ignoring interactions between AMQP and other IoT protocols that could introduce new vulnerabilities [8].

To bridge these gaps, this study focuses on securing the AMQP protocol within IoT environments by adopting a practical, protocol-specific approach. The contributions of this research include the following:

- Developing a Controlled Testbed: A simulated IoT environment is created to reproduce AMQP-specific attacks, including DDoS, MitM, and brute force scenarios.
- Generating Comprehensive Datasets: New datasets are created to capture network traffic patterns and log data for various attack types [6,7].
- Designing a Statistical Model: A robust model is developed to identify and validate unique attack signatures within the AMQP protocol [5].
- Evaluating Real-World Impact: The variations and impacts of attacks on IoT systems are analyzed, providing actionable insights for improving AMQP security [9].

This research aims to enhance the security of IoT systems by addressing the unique vulnerabilities of AMQP, ultimately contributing to the development of more resilient and trustworthy IoT ecosystems.

2. Research Gaps and Challenges

The study by [10] primarily focuses on the security vulnerabilities associated with the Advanced Message Queuing Protocol (AMQP), identifying a range of attacks, including replay and Masquerade attacks, Modification attacks, and Denial-of-Service (DoS) attacks.

However, this research lacks a practical component, as no datasets or lab-based experiments were developed to simulate or analyze these attacks. Instead, the analysis relies on a comprehensive review of existing literature and prior research. While the study provides valuable insights into the security challenges of AMQP, the absence of experimental validation and novel dataset generation limits its practical applicability.

In [11], the research explores various types of attacks across multiple IoT communication protocols, such as Cross-Site Scripting, Malicious Code Injection, Cinderella Attacks, and Big Data Handling vulnerabilities. Despite the broad scope, the work does not offer protocol-specific analysis or experimental validation. Additionally, no datasets or lab tests were developed to simulate these attacks. The study's focus on application-layer protocols is insightful but lacks specificity, as attacks like Denial of Service, Phishing, and Sniffing are not analyzed within the context of individual protocols. This omission hinders the ability to monitor network traffic or identify unique attack signatures.

The work by [12] examines various attacks on IoT application protocols, highlighting issues such as Malicious Code Injection, Denial-of-Service (DoS), Phishing, and Sniffing attacks. However, similar to the aforementioned studies, no datasets or lab-based experiments were developed to conduct these attacks. While the study addresses critical security concerns, including Man-in-the-Middle (MitM) and DoS attacks that exploit privileged access, it does not provide an in-depth exploration of attacks specific to AMQP. This gap reduces the study's relevance to AMQP-focused security challenges.

The research by [13] addresses the security of IoT devices across different layers of the IoT architecture. The study identifies attacks such as Access Control Attacks, Backdoor Attacks, and Keylogger Attacks. However, the work relies on a comparative analysis of existing literature and does not involve the development of datasets or lab tests to validate findings. Although the research highlights significant security challenges faced by IoT devices, the lack of experimental validation and novel dataset creation limits its contribution to practical security solutions.

Finally, the study by [14] investigates privacy and security in data exchange within IoT networks, identifying attacks such as Malicious Code Injection, Node Tampering, Remote Configuration Exploits, and Web Application Scanning. While the research is comprehensive, it lacks practical experimentation, as no datasets or lab tests were developed to simulate these attacks. Without analyzing network traffic traces or identifying unique attack features, the findings remain largely theoretical, limiting their applicability to real-world IoT security scenarios.

Common limitations across the reviewed works include the following:

- Lack of Experimental Validation: None of the studies developed lab-based experiments or datasets to validate their findings.
- Absence of Protocol-Specific Focus: Few studies specifically address AMQP vulnerabilities or analyze attacks within the context of individual protocols.
- Theoretical Approaches: Much of the research is based on literature reviews and lacks practical implementation or novel dataset creation to enhance its applicability.

This study seeks to address these limitations by developing a practical, AMQP-specific approach that includes lab-based experimentation, dataset generation, and the identification of unique attack signatures to enhance the security of IoT systems.

3. Literature Review

Application-layer protocols such as AMQP, MQTT, and CoAP form the backbone of IoT communication, each tailored to specific operational needs. While MQTT and CoAP have been extensively studied, AMQP's vulnerabilities and unique implications in IoT environments demand focused analysis.

3.1. Protocol Overview

- **AMQP (Advanced Message Queuing Protocol)**: A robust message-oriented middleware designed for high-throughput, reliable communication. AMQP's transactional guarantees and brokered architecture make it ideal for enterprise-grade IoT applications but introduce specific security concerns [10,15].
- **MQTT (Message Queuing Telemetry Transport)**: A lightweight publish–subscribe protocol optimized for constrained environments with low bandwidth, high latency, and limited resources [16].
- **CoAP (Constrained Application Protocol)**: A REST-based protocol designed for resource-constrained devices and networks, enabling efficient web-like interactions [17,18].

3.2. Vulnerabilities

AMQP-Specific Vulnerabilities:

1. **Broker Overload**: AMQP's reliance on brokers introduces risks such as exhaustion of resources during Distributed Denial-of-Service (DDoS) attacks [5].
2. **Replay Attacks**: AMQP's session-oriented nature is susceptible to packet replay, where attackers resend captured packets to disrupt communication [10].
3. **Queue Manipulation**: Attackers may exploit unsecured queues to reroute, drop, or duplicate messages, compromising data integrity [11].
4. **Man-in-the-Middle (MitM) Attacks**: Without robust encryption, AMQP is vulnerable to MitM attacks, allowing adversaries to intercept and alter messages [14].

MQTT Vulnerabilities:

1. Limited authentication mechanisms make it prone to brute force attacks and unauthorized access [16].
2. Susceptibility to low-rate DoS attacks like SlowITe, exploiting MQTT's lightweight nature [19].

CoAP Vulnerabilities:

1. Susceptibility to amplification attacks due to large responses to small requests, exploiting the lack of mandatory encryption [20].
2. High eavesdropping risks, as it typically uses UDP and lacks transport-layer encryption by default [21].

3.3. Security Measures for AMQP

Given its critical role in enterprise-grade IoT systems, AMQP requires robust security measures tailored to its architecture:

- **Replay Protection**: Implementing message sequencing or unique identifiers to detect and discard replayed packets [10].
- **Broker Hardening**: Rate-limiting, load balancing, and resource allocation to prevent broker exhaustion during DDoS attacks [5].
- **Encrypted Queues**: Using TLS for secure message queuing and routing, ensuring end-to-end confidentiality [7].
- **Access Controls**: Role-based access control (RBAC) for restricting unauthorized queue access.
- **Enhanced Authentication**: Adopting mutual TLS or token-based mechanisms to prevent impersonation [9].
- **Activity Monitoring**: Real-time monitoring of message queues to detect anomalies indicative of manipulation [8].

As shown in Table 1, while MQTT and CoAP are optimized for constrained environments, AMQP's focus on high-throughput, transactional reliability introduces distinct vulnerabilities. Addressing these challenges through robust countermeasures ensures AMQP can serve as a secure and resilient foundation for IoT applications in critical enterprise settings.

Table 1. Comparison of AMQP, MQTT, and CoAP protocols.

Feature	AMQP	MQTT	CoAP
Primary Use Case	Enterprise-grade messaging	Lightweight communication	RESTful IoT interactions
Transport	TCP	TCP	UDP
Authentication	Strong (e.g., TLS, RBAC)	Limited	Limited
Encryption	Mandatory (TLS recommended)	Optional (TLS/SSL)	Optional (DTLS)
Key Vulnerabilities	Broker exhaustion, replay, MitM	Brute force, low-rate DoS	Amplification, eavesdropping

4. Overview of AMQP in IoT Systems

The Advanced Message Queuing Protocol (AMQP) was first developed in 2003 by John O'Hara at JPMorgan Chase to facilitate efficient and reliable message exchange across diverse applications and systems. Operating at the application layer of the communication stack, AMQP is designed for interoperability, enabling seamless communication between platforms developed in different programming languages.

AMQP uses message brokers (e.g., the RabbitMQ server) to act as intermediaries, receiving messages from producers and distributing them to designated consumers. Its architecture features two primary components:

- Producers: Entities that generate and send messages.
- Consumers: Entities that receive and process messages.

The broker plays a central role, facilitating communication through two key mechanisms:

- Message Queues: Temporarily store messages until they are retrieved by consumers.
- Exchanges: Route messages to appropriate queues based on predefined rules.

AMQP supports flexible communication patterns, including publish/subscribe and point-to-point, making it adaptable for a wide range of applications. Its ability to prioritize and ensure the durability of messages enhances reliability in distributed systems.

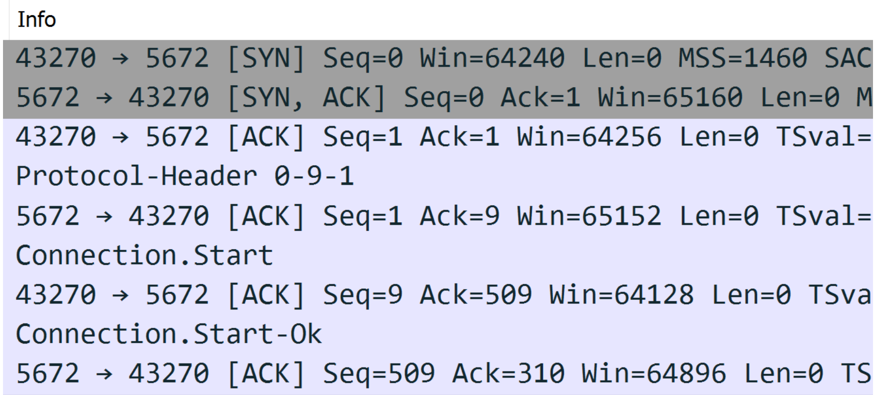
The IoT has transformed interactions with technology by connecting devices and applications seamlessly, enabling innovations in smart homes, healthcare, and industrial automation. However, this interconnectedness introduces significant security challenges, as data transmitted across networks become vulnerable to attacks. Protocols like AMQP are particularly susceptible to threats, given their critical role in managing communication in IoT environments.

To address these challenges, secure communication is a priority. AMQP supports Transport Layer Security (TLS) and Secure Socket Layer (SSL) to encrypt communications between devices and servers, ensuring data confidentiality and integrity. Despite these measures, the growing complexity of IoT networks necessitates more advanced and efficient intrusion detection systems (IDSs) tailored for application-layer protocols like AMQP [22].

By providing robust messaging features and supporting secure communication, AMQP remains a vital protocol in IoT systems. However, ongoing research and innovation are needed to address its vulnerabilities and ensure its reliability in increasingly complex network environments.

Analysis of AMQP Normal Traffic

Figure 1 illustrates the communication process between an AMQP server and an IoT device, as captured using the Wireshark tool. The process begins with a TCP three-way handshake (SYN, SYN/ACK, and ACK), establishing the initial connection. Following this, the AMQP protocol is employed to connect to the RabbitMQ server using commands such as connection.start, connection.tune, and channel.open. These actions are confirmed by acknowledgment (ACK) flags to ensure reliable communication. IoT devices then declare a queue and use the basic.consume command to subscribe and publish messages, with ACK flags verifying the successful exchange.

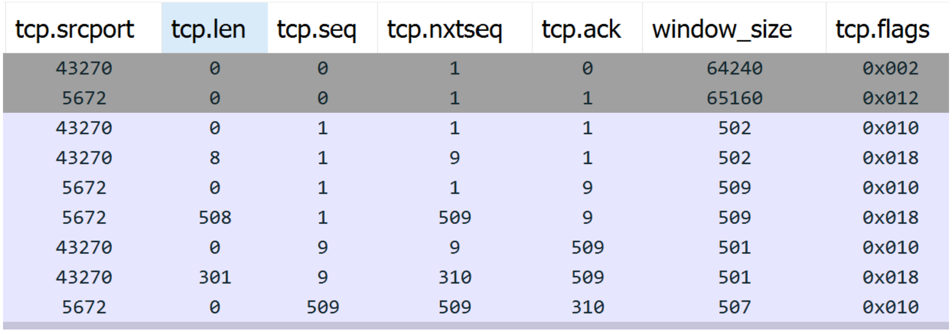


```
Info
43270 → 5672 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SAC
5672 → 43270 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 M
43270 → 5672 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=
Protocol-Header 0-9-1
5672 → 43270 [ACK] Seq=1 Ack=9 Win=65152 Len=0 TSval=
Connection.Start
43270 → 5672 [ACK] Seq=9 Ack=509 Win=64128 Len=0 TSva
Connection.Start-Ok
5672 → 43270 [ACK] Seq=509 Ack=310 Win=64896 Len=0 TS
```

Figure 1. Normal network traffic packets for AMQP.

Figure 2 provides an in-depth view of normal network traffic data exchanged between an AMQP server and an IoT device. The data exchange begins at line 6 in the capture. Key elements of the TCP connection are displayed, including the following:

TCP Length: The column shows a value of 508. Sequence Number: The initial sequence number is 1, and the next sequence number increments to 509, reflecting the addition of the TCP length. ACK Number: The ACK value is 9, corresponding to prior traces in the network traffic. The window size for the IoT device consistently starts at 64,240, while the AMQP server's window size begins at 65,160. These window sizes adjust dynamically based on the data traces as the systems continue the data exchange process. The figures demonstrate how reliable communication is maintained during normal traffic between the AMQP server and the IoT device.



tcp.srcport	tcp.len	tcp.seq	tcp.nxtseq	tcp.ack	window_size	tcp.flags
43270	0	0	1	0	64240	0x002
5672	0	0	1	1	65160	0x012
43270	0	1	1	1	502	0x010
43270	8	1	9	1	502	0x018
5672	0	1	1	9	509	0x010
5672	508	1	509	9	509	0x018
43270	0	9	9	509	501	0x010
43270	301	9	310	509	501	0x018
5672	0	509	509	310	507	0x010

Figure 2. Normal network data exchange traffic for AMQP.

5. Attack Description and Explanation

This section provides a detailed description and explanation of the attacks conducted against the AMQP IoT application protocol. These attacks, simulated in our testbed, address scenarios not yet covered in the current research literature [23]. Below is an overview of the attacks investigated.

5.1. Experiment Setup

The Internet of Things (IoT) testbed used in this study simulates a real-world deployment of IoT networks. The testbed consists of various IoT devices, an AMQP server, an attacker's computer, network hardware, and the AMQP communication protocol.

- **Hardware Configuration**:
  - The primary lab machine is a Dell XPS Windows 11 laptop with 1 TB SSD, an Intel Core i5-11270U processor, 16 GB of RAM, and a 4 GB NVIDIA RTX GPU.
  - Virtual machines (VMs) are managed using VirtualBox. An Ubuntu-based VM serves as the AMQP server, with 4 GB of RAM, 20 GB of storage, and 2 CPU cores.
  - Each IoT device is simulated using a VM allocated 2 GB of RAM, 15 GB of storage, and 1 CPU core.
  - Two attacker computers are configured with 4 CPU cores, 8 GB of RAM, and 80 GB of storage for launching attacks.
- **Network Configuration**
  - The virtual machines reside in a private network configured within VirtualBox's internal networking mode to ensure isolated, controlled traffic flow.
  - The AMQP broker (RabbitMQ server) is assigned a fixed private IP (e.g., 192.168.180.230), with all IoT devices and attacker nodes connected through a virtual LAN (vLAN).
  - Packet capture and analysis were performed using Wireshark and tcpdump within the virtualized network.
- **Platform for Simulating Attacks**
  The attack modeling and execution were performed within a controlled VirtualBox-based network, ensuring traffic flows accurately reflect AMQP-specific attack scenarios. The Ettercap tool was employed for MiTM attacks, while Hping3 and BruteMQ were used for DoS and brute force attacks, respectively.

This setup allows the simulation of real-world attacks against AMQP-based systems while monitoring and analyzing the network traffic.

5.2. Explanation of DoS Attack

A Denial-of-Service (DoS) attack aims to make a network or system inaccessible to its intended users. This is achieved by overwhelming the target with an excessive volume of data or sending malformed packets that disrupt normal operations, ultimately causing system failure.

In the context of AMQP, DoS attacks exploit the protocol's reliance on messaging brokers, such as RabbitMQ, by overwhelming them with connection requests or data packets. This prevents legitimate messages from being processed, leading to system crashes or unresponsiveness.

For our experiments, we used the **hping3** tool to execute SYN flood attacks on the AMQP server. The attacker sends a high volume of SYN packets to port 5672, which is designated for AMQP communication. These packets initiate a TCP handshake but do not complete it, exhausting the server's resources and rendering it inaccessible.

Command for Hping3 Attack:

```
hping3 -d 90 -p 5672 -S –flood 192.168.180.230
```

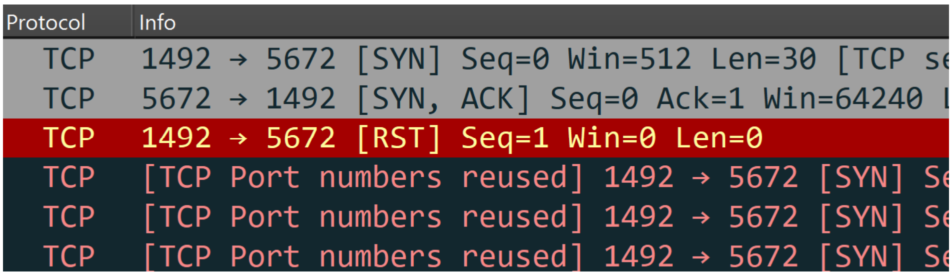
Options:

- `-d`: Data size (in bytes).
- `-p`: Target port number.
- `-S`: SYN flag.
- `-flood`: Continuous packet sending.
- `192.168.180.230`: Target IP address.

5.2.1. Experiment Results and Analysis

Experiment 1: SYN Flood Attack

Figure 3 demonstrates the impact of an SYN flood attack during the TCP three-way handshake process. The attacker manipulates the SYN segment to overwhelm the target server. Despite variations in attack parameters, the SYN flag remains constant as the attacker repeatedly sends SYN packets, causing the server to respond with SYN/ACK packets. The connection is then terminated by an RST packet, allowing the attack to restart and flood the server.

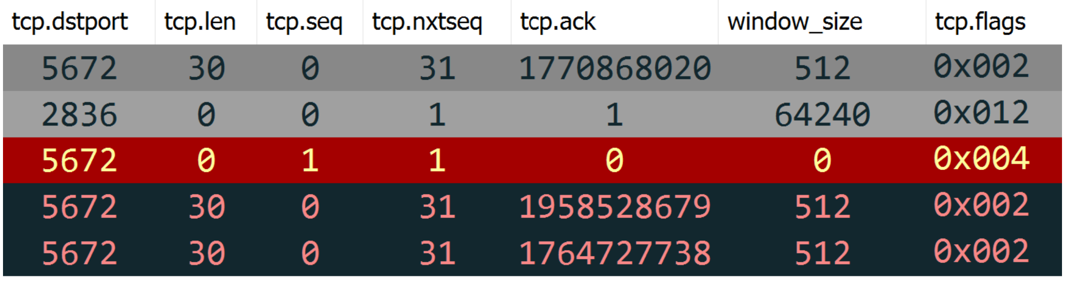


Protocol	Info
TCP	1492 → 5672 [SYN] Seq=0 Win=512 Len=30 [TCP se
TCP	5672 → 1492 [SYN, ACK] Seq=0 Ack=1 Win=64240 L
TCP	1492 → 5672 [RST] Seq=1 Win=0 Len=0
TCP	[TCP Port numbers reused] 1492 → 5672 [SYN] Se
TCP	[TCP Port numbers reused] 1492 → 5672 [SYN] Se
TCP	[TCP Port numbers reused] 1492 → 5672 [SYN] Se

Figure 3. DoS attack TCP handshake flags.

Figure 4 shows the malicious packets sent during the AMQP DoS attack. Key features include the following:

- **TCP Length**: Typically zero during malicious SYN packets.
- **Sequence and Acknowledgment Numbers**: Randomized and manipulated to bypass detection.
- **Window Size**: The attacker uses a fixed size of 512, while the server defaults to 64240 due to its inability to detect the attack.

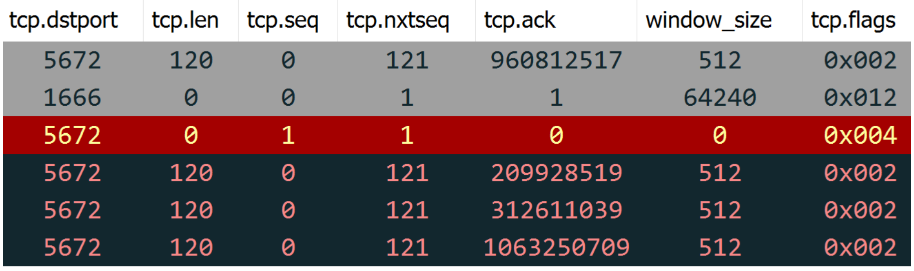


tcp.dstport	tcp.len	tcp.seq	tcp.nxtseq	tcp.ack	window_size	tcp.flags
5672	30	0	31	1770868020	512	0x002
2836	0	0	1	1	64240	0x012
5672	0	1	1	0	0	0x004
5672	30	0	31	1958528679	512	0x002
5672	30	0	31	1764727738	512	0x002

Figure 4. DoS attack data exchange packets (Experiment 1).

Experiment 2: Malicious Data in SYN Packets

In this experiment, the attacker sends 120 bytes of data within SYN packets, violating the TCP protocol. The target responds to these packets without detecting the anomaly. Figure 5 illustrates the attack, where manipulated SYN packets contain payload data while the server continues responding with SYN/ACK packets.

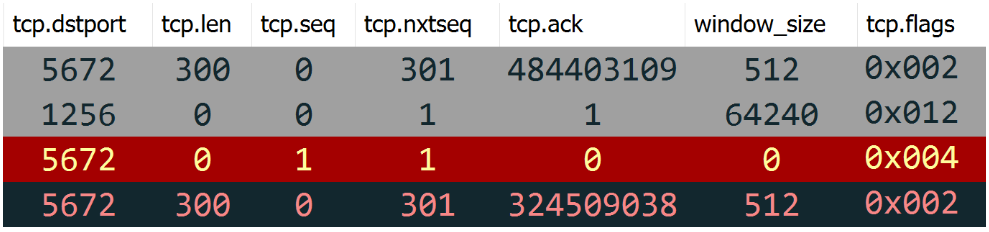


tcp.dstport	tcp.len	tcp.seq	tcp.nxtseq	tcp.ack	window_size	tcp.flags
5672	120	0	121	960812517	512	0x002
1666	0	0	1	1	64240	0x012
5672	0	1	1	0	0	0x004
5672	120	0	121	209928519	512	0x002
5672	120	0	121	312611039	512	0x002
5672	120	0	121	1063250709	512	0x002

Figure 5. DoS attack data exchange packets (Experiment 2).

Experiment 3: Larger Malicious Data Payloads

The attacker increases the payload size to 300 bytes in the SYN packets. Despite the abnormal data, the target system fails to identify the malicious nature of these packets and continues processing them, as shown in Figure 6.



tcp.dstport	tcp.len	tcp.seq	tcp.nxtseq	tcp.ack	window_size	tcp.flags
5672	300	0	301	484403109	512	0x002
1256	0	0	1	1	64240	0x012
5672	0	1	1	0	0	0x004
5672	300	0	301	324509038	512	0x002

Figure 6. DoS attack data exchange packets (Experiment 3).

5.2.2. Traits and Signature of DoS Attacks

The analysis revealed key characteristics of AMQP DoS attacks, summarized in Table 2. The methods and features used are as follows:

- TCP Flag Analysis: The model evaluates TCP flags such as SYN, ACK, and RST, which are critical in identifying connection-based anomalies. For instance, a high frequency of SYN packets without corresponding ACK responses is indicative of SYN flood attacks.
- Packet Frequency Monitoring: By observing the rate of incoming packets over a specific time window, the model detects sudden spikes in traffic volume, characteristic of volumetric DoS attacks.
- Sequence Number Irregularities: The statistical model monitors the continuity of sequence numbers in packet flows. Disruptions or repeated numbers often indicate replay attacks or DoS activities attempting to disrupt normal communication.
- Payload Size Analysis: The model detects anomalous traffic patterns by examining payload sizes. Abrupt changes in payload size, especially a large number of packets with empty payloads, are often linked to malicious traffic.

These anomalies form the basis for identifying DoS attack signatures.

Table 2. Abnormal traffic features of AMQP DoS attacks.

Features	Abnormal Traffic
TCP Handshake	RST ∣ SYN ∣ SYN/ACK
TCP Segment Features (Length ∣ Seq ∣ NxtSeq ∣ Ack)	0 ∣ Randomized ∣ 1 ∣ Randomized
Window Size	512 (Attacker) ∣ 64,240 (Target)
Syslog Logs	TCP SYN flood detected on port 5672

5.3. Explanation of Man-in-the-Middle (MiTM)

Man-in-the-Middle (MiTM) attacks are a type of cyberattack where an adversary intercepts and manipulates communication between two parties, deceiving them into believing they are directly connected. This attack can also be considered a form of eavesdropping, as the attacker gains unauthorized access to and control over the communication. In the context of the Advanced Message Queuing Protocol (AMQP), MiTM attacks can be particularly disruptive. For instance, an attacker could fabricate temperature data and transmit them to a remote server or disable unsecured HVAC systems during extreme heat, creating critical challenges for service providers reliant on vulnerable models [7].

In this study, the **Ettercap** tool was used to perform an MiTM attack, enabling the interception of communication between the client and the AMQP server.

Command for Ettercap Attack:

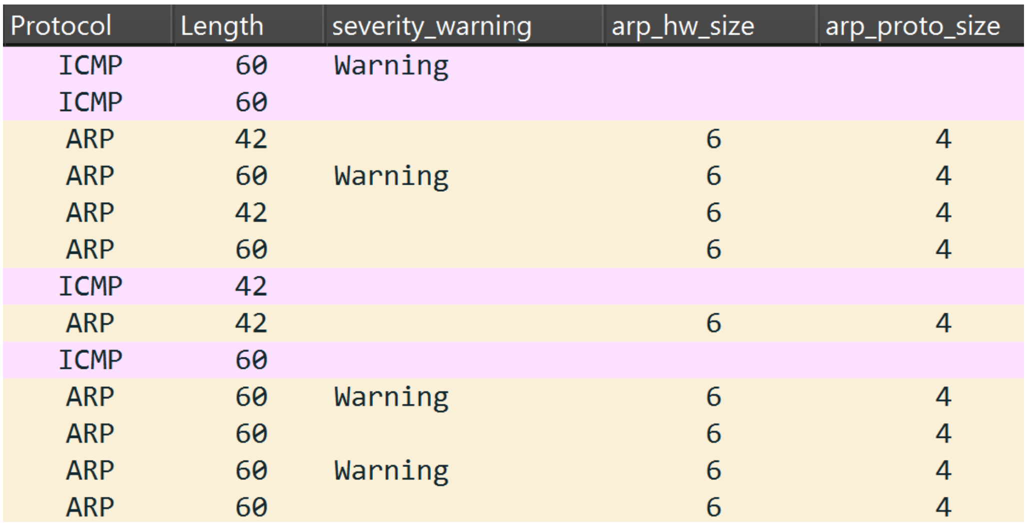
ettercap -T -M arp /192.168.180.1/ /192.168.180.10/

Explanation of the Command:

- `-T`: Enables text mode.
- `-M arp`: Specifies ARP poisoning as the attack method.
- `/192.168.180.1/ /192.168.180.10/`: Defines the IP addresses of the target and victim machines.

Figure 7 presents the malicious traffic captured during the MiTM attack. The attack begins with ICMP packets, used to test the connection between the attacker and the target. Analysis of the captured data highlights key indicators of the attack:

- **Severity Warnings**: Two instances of warnings related to potential IP address duplication were detected.
- **Packet Length**: The `tcp.len` parameter consistently measured 60 bytes across all traces.
- **MAC Address Size**: Both the attacker and victim systems maintained a hardware size of 6 bytes.
- **Protocol Size**: Remained uniformly at 4 bytes throughout all tests.



Protocol	Length	severity_warning	arp_hw_size	arp_proto_size
ICMP	60	Warning		
ICMP	60			
ARP	42		6	4
ARP	60	Warning	6	4
ARP	42		6	4
ARP	60		6	4
ICMP	42			
ARP	42		6	4
ICMP	60			
ARP	60	Warning	6	4
ARP	60		6	4
ARP	60	Warning	6	4
ARP	60		6	4

Figure 7. Analysis of AMQP MiTM packets.

5.3.1. Analysis

Under normal traffic conditions, the communication between an IoT device and an AMQP server begins with a TCP three-way handshake: the IoT device sends an SYN request, the server responds with an SYN/ACK, and the device acknowledges with an ACK. Once this connection is established, data exchange proceeds with correct flags and window sizes.

In the case of an AMQP MiTM attack, the attacker uses ICMP echo requests containing big-endian (BE) and little-endian (LE) sequence numbers (e.g., 32,487 and 59,262). The AMQP server responds with ICMP echo replies, often containing similar endianness indicators to ensure proper data interpretation. Following this, the attacker spoofs the IP and MAC addresses of the target, allowing interception of the communication.

Key observations during the attack include the following:

- The binding IP and MAC addresses are identified as spoofed once a response is received.
- If no response is obtained, the packet is considered dropped.

- The syslog indicates warnings from the network manager regarding IP conflicts during the attack, further confirming malicious activity.

5.3.2. Traits and Signature

Several features were identified as indicators of an AMQP MiTM attack. These include the following:

- **Severity Warnings**: Triggered by ARP protocol warnings due to duplicate IP and MAC addresses.
- **Echo Requests and Replies**: Used by the attacker to establish communication and monitor responses.
- **Frame Length**: Consistently measured at 60 bytes.
- **Syslog Messages**: Alerts about changes to AMQP IP address states.

These features are often used in attack signatures to detect potential MiTM attacks, as they reflect suspicious or malicious network behavior. Additional indicators, such as anomalies in routing tables or unexpected changes in MAC address tables, further support the identification of MiTM activity.

Table 3 summarizes the abnormal traffic features observed during the AMQP MiTM attack, forming the basis for its signature.

Table 3. Abnormal traffic features of AMQP MiTM attack.

Features	Abnormal Traffic
TCP Three-Way Handshake	Echo Ping Request ∣ Echo Ping Reply
Severity Level	Warning
Frame Length	60 bytes
Syslog Messages	AMQP IP Address State Changed

5.4. Explanation of AMQP Brute Force Attack

Brute force attacks are a type of cyberattack where unauthorized users attempt to infiltrate a system by repeatedly guessing username and password combinations or uncovering protected information through exhaustive attempts. Attackers often employ various techniques to improve the accuracy of their guesses. Within the AMQP protocol, queue message handling is a vulnerable process that attackers can exploit to gain unauthorized access. By leveraging brute force methods, attackers can obtain the credentials needed to access the system [5].

Brutemq is a tool designed to brute force RabbitMQ server (AMQP) queue messaging services. The tool works by connecting to a given host on the AMQP port (5672/TCP) and attempting to guess passwords from a predefined list [24].

Command for Brutemq Attack:

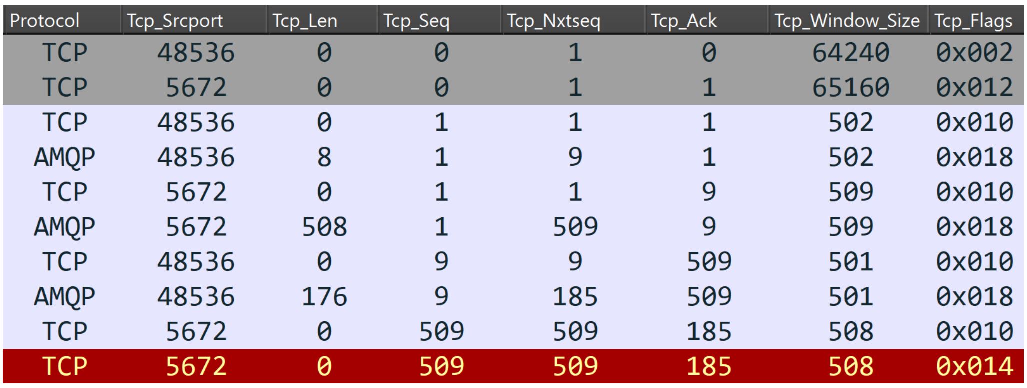
```
brutemq amqp -d passwords.txt -u admin -e 192.168.180.10:5672 -t 500
```

Explanation of the Command:

- `-d`: Specifies the password list file.
- `-u`: Indicates the username (e.g., `admin`).
- `-e`: Specifies the target host and port.
- `-t`: Sets the timeout for the attack.

5.4.1. Brute Force Attack Signature

In Figure 8, the analysis reveals that the first three lines of malicious traffic illustrate the TCP three-way handshake connection. The process begins with the IoT device sending an SYN packet to the AMQP server. The server responds with an SYN/ACK packet, and the IoT device validates the connection with an ACK packet. The IoT device then sends an AMQP protocol version 0-9-1 packet with the "PSH, ACK" flag to the server. The server confirms the request with an ACK and establishes the connection with a `connection.Start` packet. The IoT device responds with `connection.Start-Ok`, including the username and password for verification. The server immediately verifies the connection with an ACK packet.



Protocol	Tcp_Srcport	Tcp_Len	Tcp_Seq	Tcp_Nxtseq	Tcp_Ack	Tcp_Window_Size	Tcp_Flags
TCP	48536	0	0	1	0	64240	0x002
TCP	5672	0	0	1	1	65160	0x012
TCP	48536	0	1	1	1	502	0x010
AMQP	48536	8	1	9	1	502	0x018
TCP	5672	0	1	1	9	509	0x010
AMQP	5672	508	1	509	9	509	0x018
TCP	48536	0	9	9	509	501	0x010
AMQP	48536	176	9	185	509	501	0x018
TCP	5672	0	509	509	185	508	0x010
TCP	5672	0	509	509	185	508	0x014

Figure 8. Analysis of AMQP brute force attack packets.

5.4.2. Analysis

This section examines the features identified in the brute force attack signature against the AMQP server. Key features of the attack include the following:

- `tcp.dstport=5672`.
- `tcp.len=0`.
- `seq.number=509`.
- `nxtseq.number=509`.
- `window.size=508`.
- `tcp.flag=0x014` in RST/ACK packets.

Repeated experiments confirmed that these values remained unchanged. These patterns occur when an attacker or IoT device attempts to establish a connection with the target machine, following default packet-specific settings for connection and authorization. Each incorrect login attempt results in the target system discarding the previous message, generating an RST/ACK packet with random sequence numbers to disrupt transmission.

The value of `window.size` consistently reflects the RST/ACK packet from the previous ACK packet when login details are incorrect. The syslog logs multiple attempts to connect on port 5672, indicating failed and successful login attempts. Additionally, the syslog detects and logs the correct credentials when successfully brute-forced.

5.4.3. Traits and Signature

Key features, such as `tcp.dstport=5672`, `tcp.len=0`, `seq.num=509`, `nxtseq.num=509`, `window.size=508`, and `tcp.flag=0x014`, are essential for identifying potential security threats, including brute force attacks. These features serve as a foundation for detecting anomalies associated with repeated login attempts. Below is a summary of the traits and their significance:

- **TCP Destination Port (**`tcp.dstport=5672`**)**: Critical for AMQP protocol connections.
- **TCP Length (**`tcp.len=0`**)**: Indicates no data in the packet payload.
- **Sequence and Next Sequence Numbers (**`seq.num=509`, `nxtseq.num=509`**)**: Tracks packet order.
- **Window Size (**`window.size=508`**)**: Reflects the data transmission capacity for the RST/ACK packet.

- **TCP Flags (`tcp.flag=0x014`)**: Used to reset connections during repeated failed attempts.

Consistent values in RST/ACK packets throughout repeated experiments indicate that the targeted machine terminates the connection using identical values. The attacker manipulates these packets to disrupt the system and prolong the attack, aiming to obtain valid login credentials.

5.4.4. Abnormal Traffic Features of AMQP Brute Force Attack

Table 4 summarizes the abnormal traffic features observed during the brute force attack, highlighting key indicators for detecting such attacks.

Table 4. Abnormal traffic features of AMQP brute force attack.

Features	Abnormal Traffic
TCP Three-Way Handshake	RST/ACK
RST/ACK Packet	`tcp.len=0 | seq=509 | nxtseq=509 | tcp.srcport=5672 | window.size=508`
Syslog	Invalid credentials on port 5672

6. Analysis and Validation of Statistical Model for AMQP IoT Attack Detection

This section presents the analysis and validation of the statistical model used to detect attacks within the AMQP IoT protocol. The methodology involves leveraging a dataset generated from network traffic captured during the experiments described earlier. This dataset provides the foundation for identifying the unique features and signatures of various attack types.

The dataset [25] was generated by capturing network traffic from IoT devices communicating via the AMQP protocol. Using our testbed setup, the data collection process included a wide range of scenarios, encompassing normal operations, malicious activities, and attack traffic. Packets were categorized into three classes:

- **Normal Traffic**: Data packets representing typical, unaltered communication between IoT devices and the AMQP server.
- **Malicious Traffic**: Packets exhibiting abnormal behavior or unauthorized activities indicative of potential security threats.
- **Attack Traffic**: Packets directly associated with known attack patterns, such as DoS, MiTM, or brute force attacks.

This classification enabled a comprehensive analysis of network behavior, facilitating the identification of the distinct features that define each attack signature.

To analyze the dataset and validate attack detection mechanisms, a Python-based statistical model script was developed. The script identifies key features from the captured traffic and classifies packets based on the signatures defined in earlier sections. The statistical model was tested against three types of datasets:

- **Benign Dataset**: Contains only normal traffic to evaluate false positive rates.
- **Malicious Dataset**: Comprises known malicious and attack traffic for accuracy testing.
- **Mixed Dataset**: Includes both normal and malicious traffic to simulate real-world scenarios and measure overall detection performance.

The statistical model also incorporates a visualization component to facilitate analysis and enhance understanding of network traffic patterns. A color-coded scheme is used to represent different traffic categories:

- **Blue**: Represents normal traffic.
- **Gray**: Indicates malicious traffic.

- **Red**: Denotes attack traffic.

This visual approach simplifies the evaluation process by enabling clear identification of anomalies and patterns within the traffic data. By combining automated classification with visual representation, the statistical model provides a robust framework for AMQP IoT attack detection.

6.1. Feature Selection and Weighting

The statistical model leverages a carefully selected set of features to ensure robust performance in detecting known attack types. Features such as TCP flags, sequence numbers, payload lengths, and packet sizes were chosen based on their relevance in identifying anomalous traffic patterns. These features were selected for the following reasons:

- TCP Flags: They indicate communication behaviors, such as connection initiation or termination, which are critical in identifying patterns related to DoS or MiTM attacks.
- Sequence Numbers: These help track packet flow, enabling the detection of anomalies like replay attacks or out-of-sequence packets.
- Payload Lengths: Variations in payload size can signal unusual traffic behaviors, often associated with brute force or protocol abuse.
- Packet Sizes: They serve as indicators of traffic irregularities, especially in cases of data exfiltration or malicious payload delivery.

The weighting of these features in the model was determined through iterative testing to optimize detection accuracy while minimizing false positives. Features contributing most significantly to attack identification, such as TCP flags and sequence numbers, were assigned higher weights.

Justification for this feature selection and weighting lies in their proven effectiveness in previous studies and their ability to generalize across a wide range of attack scenarios. Additionally, future work will explore integrating time-series data into the feature set to uncover evolving attack patterns over time, such as gradual resource exhaustion or incremental brute force attempts. Dynamic weighting schemes incorporating temporal trends could further improve model adaptability and precision.

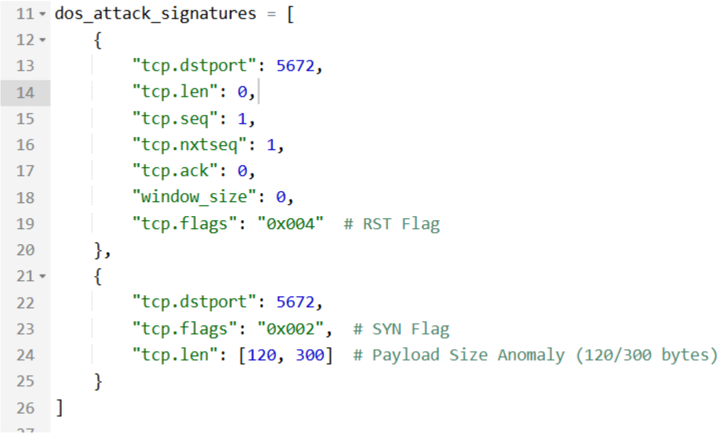
Signature Definition and Extraction

Signatures are quantitatively defined and extracted from traffic data based on distinct patterns and statistical anomalies indicative of specific attack types. This process involves the following:

- Feature-Based Quantification: Each signature is represented by a combination of features such as TCP flags, sequence numbers, payload lengths, and packet sizes. For instance, an SYN flood attack is characterized by a high frequency of SYN packets without corresponding ACK responses, while irregular sequence numbers may indicate replay attacks.
- Threshold Establishment: Statistical thresholds are set for each feature by analyzing normal traffic baselines. Any deviation beyond these thresholds is flagged as anomalous. For example, a spike in packet frequency or an unusual payload size may trigger an alert.
- Traffic Profiling: Traffic flows are monitored over time to identify repetitive patterns or behaviors indicative of malicious activity. These profiles form the foundation of the signature database.
- Empirical Validation: Known attack datasets and testbed simulations (e.g., using tools like hping3) are analyzed to identify consistent patterns that reliably distinguish malicious traffic from normal behavior. These findings are used to refine and validate the signature extraction process.

6.2. Detecting the DoS Attack

In order to test the effectiveness of our signature-based detection, we prepared a script that looks for the signatures in network traffic and identifies malicious packets as shown in Figure 9.



```
dos_attack_signatures = [
    {
        "tcp.dstport": 5672,
        "tcp.len": 0,
        "tcp.seq": 1,
        "tcp.nxtseq": 1,
        "tcp.ack": 0,
        "window_size": 0,
        "tcp.flags": "0x004"  # RST Flag
    },
    {
        "tcp.dstport": 5672,
        "tcp.flags": "0x002",  # SYN Flag
        "tcp.len": [120, 300]  # Payload Size Anomaly (120/300 bytes)
    }
]
```

Figure 9. DoS attack signature.

6.2.1. Detection Results for Normal and Malicious Datasets in DoS Attacks

Figure 10 illustrates the classification results of network traces into three categories: normal, malicious, and RST packets associated with a DoS attack. A custom script was developed to identify the specific attributes unique to malicious and RST packets, while all other traces were categorized as normal. The results show the model's ability to accurately distinguish between these categories, with the following distribution:

- A total of 27.28% of the traces were identified as normal traffic.
- A total of 71.50% of the traces were classified as malicious traffic.
- A total of 1.22% of the traces were recognized as RST DoS packets.

These results demonstrate the effectiveness of the script in identifying and classifying attack-specific traffic in the dataset.

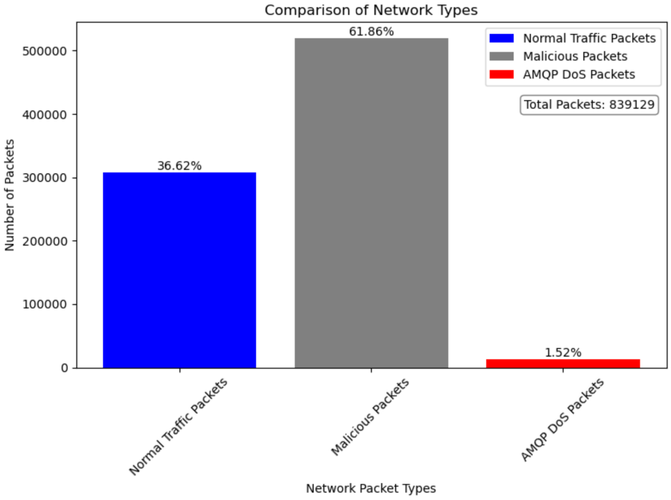


Figure 10. Classification results of network traces for DoS attacks (normal, malicious, and RST).

6.2.2. Detection Results for Normal Dataset in AMQP Traffic Packets

Figure 11 demonstrates the script's performance when applied to a dataset containing only normal traces of AMQP network traffic. The results reveal the script's ability to accurately classify benign packets, with the following:

- A 100% detection rate for normal traces.
- A 0% detection rate for malicious and RST traces.

This highlights the model's precision in processing normal datasets without generating false positives.

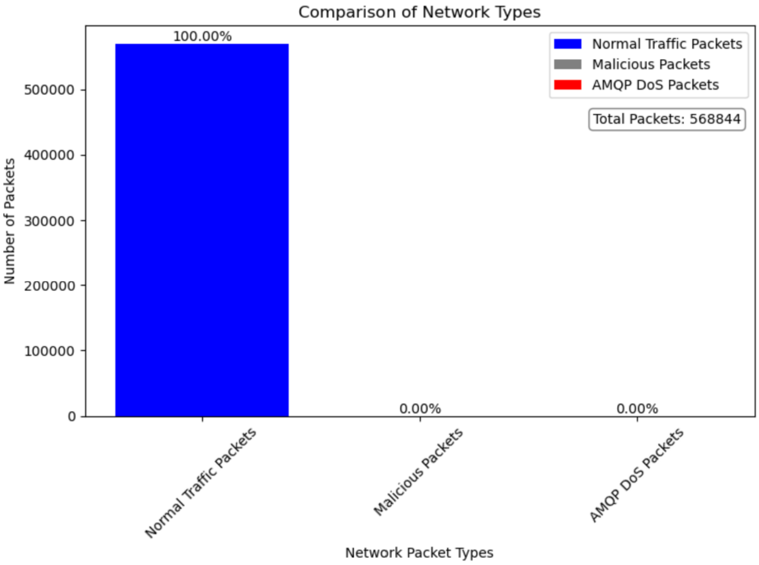


Figure 11. Detection of normal packets in AMQP traffic dataset.

6.2.3. Detection Results for Malicious Dataset in DoS Packets

Figure 12 presents the classification of a malicious dataset containing both normal and harmful traces. The script was applied to categorize RST, malicious, and normal traces sequentially. The results show the following:

- A total of 98.76% of the packets were classified as malicious traffic.
- A total of 1.13% of the packets were identified as RST DoS packets.
- A total of 0.10% of the packets were classified as normal traffic.

These results emphasize the robustness of the model in accurately identifying malicious traffic while minimizing false positives for normal packets.

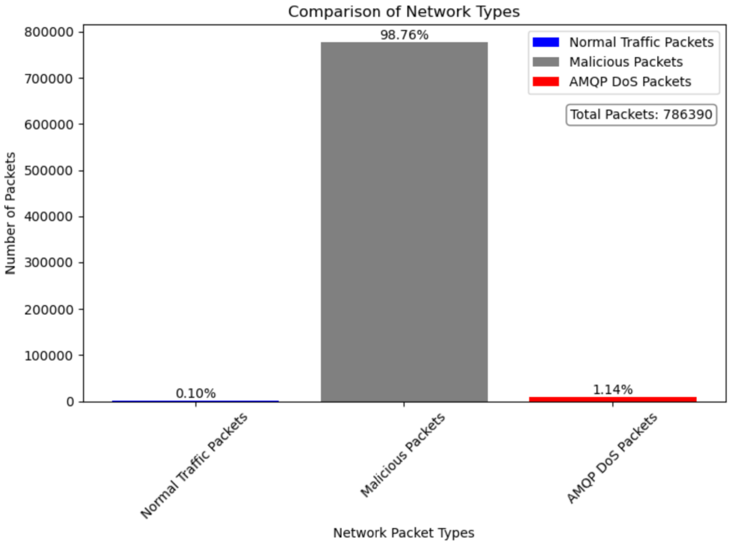
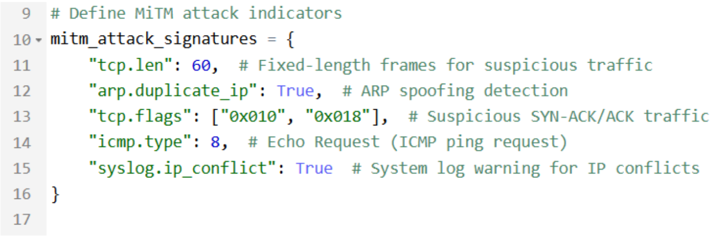


Figure 12. Detection results for malicious packets in DoS dataset.

6.3. Detecting the MiTM Attack

We prepared a script that looks for the signatures in network traffic and identifies malicious packets as shown in Figure 13.



```
# Define MiTM attack indicators
mitm_attack_signatures = {
    "tcp.len": 60,  # Fixed-length frames for suspicious traffic
    "arp.duplicate_ip": True,  # ARP spoofing detection
    "tcp.flags": ["0x010", "0x018"],  # Suspicious SYN-ACK/ACK traffic
    "icmp.type": 8,  # Echo Request (ICMP ping request)
    "syslog.ip_conflict": True  # System log warning for IP conflicts
}

```

Figure 13. MiTM attack signature.

6.3.1. Detection Results for Normal and Malicious Datasets in MiTM Attacks

Figure 14 presents the detection results of the statistical model applied to differentiate between normal, malicious, and duplicate IP packets associated with AMQP Man-in-the-Middle (MiTM) attacks. The script aggregates packets based on their IP addresses and categorizes them as malicious if duplicate IP addresses are detected. All other packets are classified as normal.

The results indicate the script's ability to accurately identify and classify MiTM attack signatures:

- A total of 99.95% of packets were classified as normal.
- A total of 0.02% of packets were identified as malicious.
- A total of 0.04% of packets were recognized as duplicate IP address MiTM packets.

These findings demonstrate the effectiveness of the script in identifying MiTM attack characteristics while maintaining a high accuracy rate.

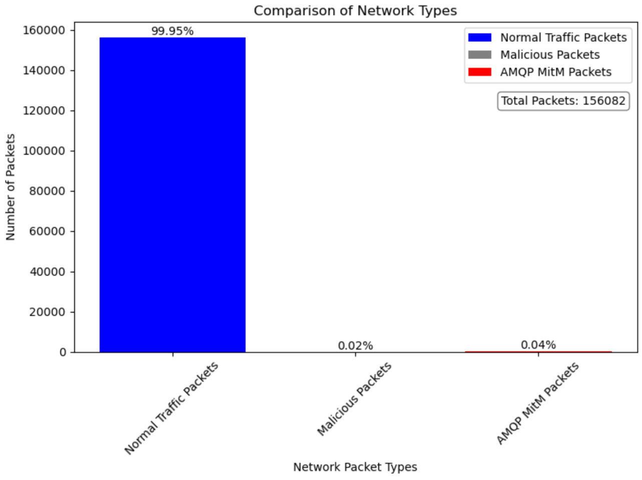


Figure 14. Detection results for MiTM attack packets in normal and malicious datasets.

6.3.2. Detection Results for Normal Dataset in AMQP Traffic Packets

Figure 15 illustrates the performance of the AMQP MiTM detection script when applied to a dataset containing only normal AMQP network traffic. The objective was to evaluate the script's ability to classify benign traffic accurately.

The detection results are as follows:

- Overall, 100% of normal packets were correctly identified.
- No MiTM attack or malicious packets were detected.

These results highlight the model's reliability in handling normal traffic scenarios without generating false positives.

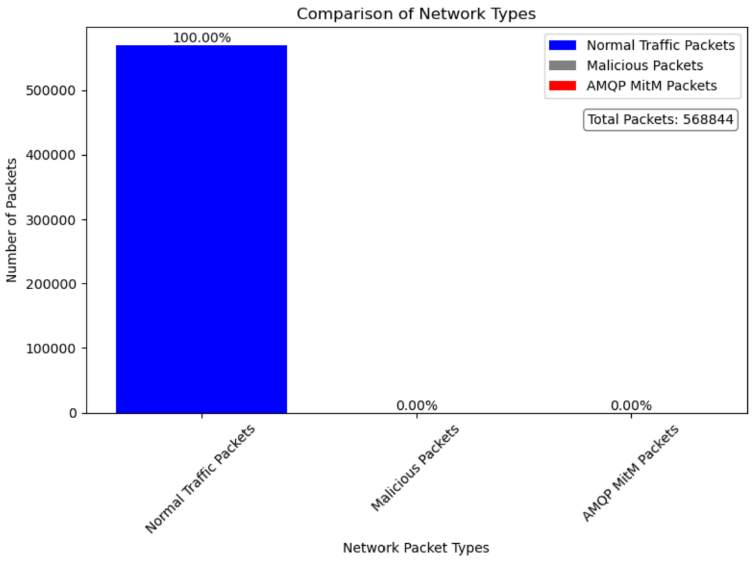


Figure 15. Detection results for normal packets in AMQP traffic.

6.3.3. Detection Results for Malicious Dataset in MiTM Packets

Figure 16 presents the analysis of a malicious dataset using the AMQP MiTM detection script. The results were analyzed to evaluate the model's precision and false positive ratio.

Key findings include the following:

- A total of 63.54% of packets were identified as malicious.
- A total of 30.21% of these malicious packets were associated with MiTM attacks.
- A total of 6.25% of packets were classified as normal traffic.

These results emphasize the model's robustness in detecting MiTM attacks within a malicious dataset while maintaining a low false positive rate for normal traffic.

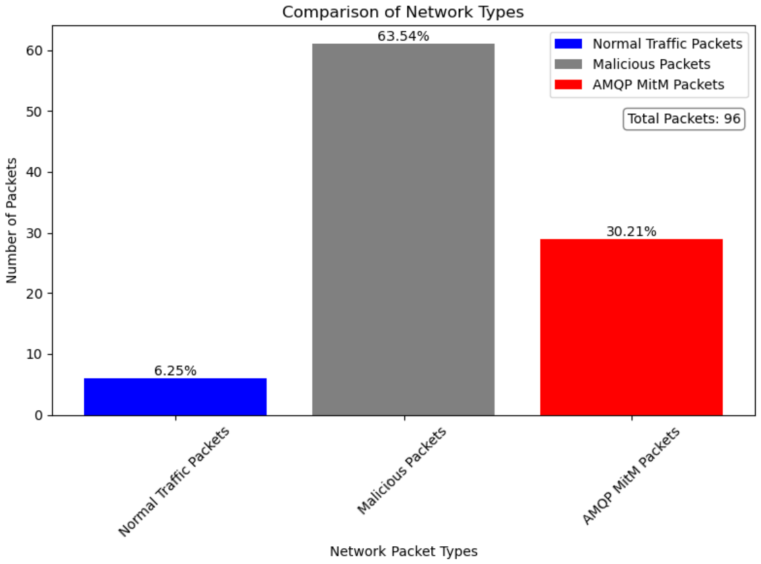
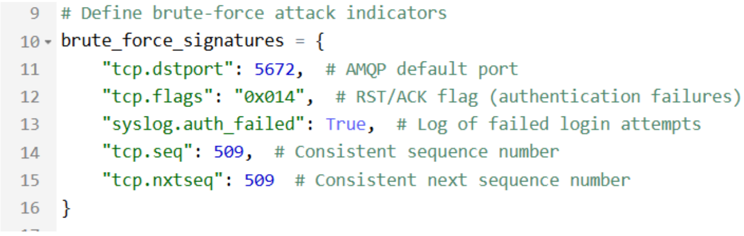


Figure 16. Detection results for malicious packets in MiTM dataset.

6.4. Detecting the Brute Force Attack

The script belwo looks for the signatures in network traffic and identifies malicious packets as shown in Figure 17.



```
# Define brute-force attack indicators
brute_force_signatures = {
    "tcp.dstport": 5672,  # AMQP default port
    "tcp.flags": "0x014",  # RST/ACK flag (authentication failures)
    "syslog.auth_failed": True,  # Log of failed login attempts
    "tcp.seq": 509,  # Consistent sequence number
    "tcp.nxtseq": 509  # Consistent next sequence number
}
```

Figure 17. Brute force attack signature.

6.4.1. Detection Results for Normal and Malicious Datasets in Brute Force Attacks

Figure 18 illustrates the classification results of network packets into normal, malicious, and AMQP RST/ACK packets associated with brute force attacks. A custom algorithm was developed to identify the unique characteristics of brute force attack signatures, with all other packets categorized as normal.

The statistical model produced the following results:

- A total of 95.77% of the packets were classified as normal traffic.
- A total of 3.83% of the packets were identified as malicious traffic.
- A total of 0.40% of the packets were classified as AMQP RST/ACK brute force attack packets.

These results demonstrate the script's accuracy in detecting and distinguishing brute force attack packets within the dataset.

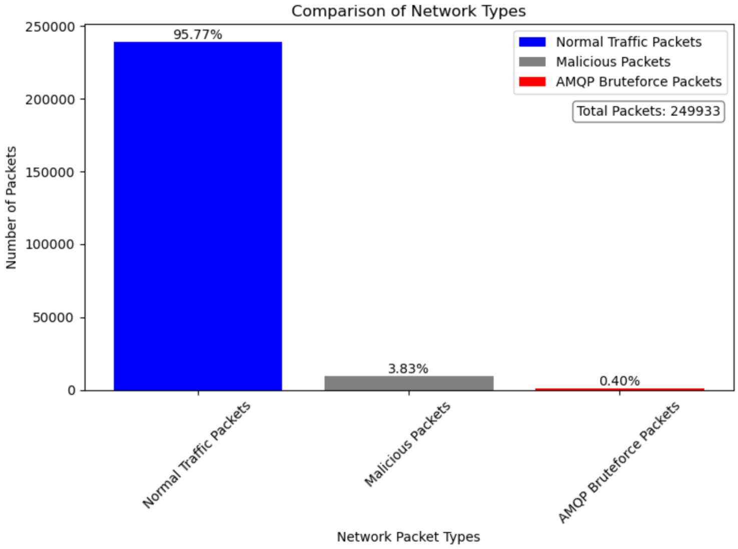


Figure 18. Detection results for normal and malicious packets in brute force attacks.

6.4.2. Detection Results for Normal Dataset in AMQP Traffic Packets

Figure 19 presents the performance of the AMQP brute force detection script when applied to a dataset containing only normal AMQP traffic. The results indicate the following:

- A 100% detection rate for normal packets.
- A 0% detection rate for malicious and brute force attack packets.

These findings highlight the script's ability to accurately classify normal traffic without generating false positives.

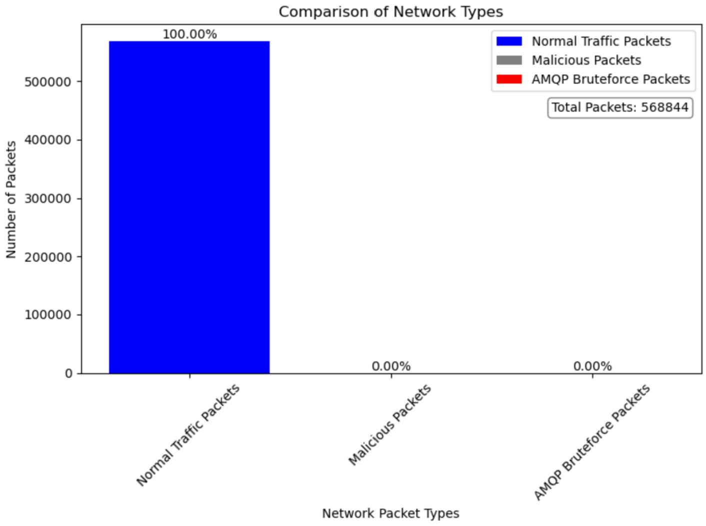


Figure 19. Detection results for normal packets in AMQP brute force dataset.

6.4.3. Detection Results for Malicious Dataset in Brute Force Packets

Figure 20 depicts the detection results when the AMQP brute force detection script was applied to a malicious dataset. The evaluation revealed the following:

- A total of 90.42% of the packets were classified as malicious traffic.
- A total of 9.56% of the packets were identified as brute force attack packets.
- A total of 0.02% of the packets were classified as normal traffic.

These results demonstrate the model's high precision in detecting malicious events and brute force attacks within the dataset, while maintaining a low false positive rate for normal traffic.

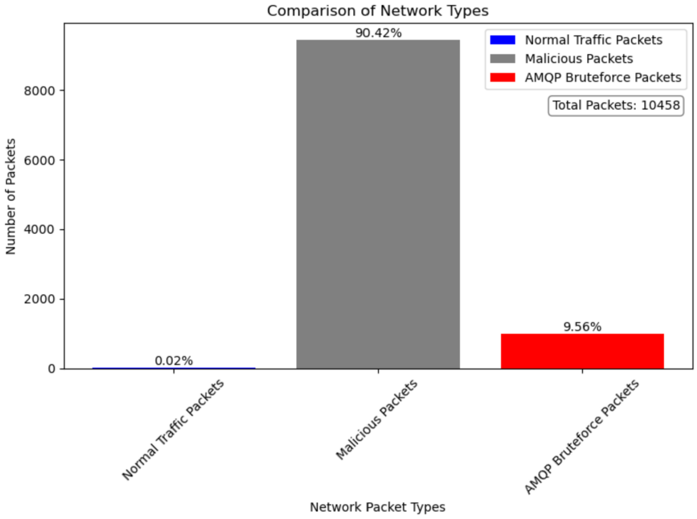


Figure 20. Detection results for malicious packets in AMQP brute force dataset.

6.5. Summary of the Statistical Model Performance

The statistical model was rigorously evaluated across three attack scenarios—DoS, MiTM, and brute force attacks—using datasets containing normal, malicious, and mixed traffic. The model's performance metrics, including detection rates, false positives, and classification accuracy, are summarized and analyzed below.

6.5.1. Detection Accuracy Across Attack Scenarios

The model demonstrated high detection accuracy for attack-specific traffic while maintaining robust performance for normal traffic. Key performance indicators are detailed as follows:

- **DoS Attacks:**
  - Detection Rate:
    - A total of 71.50% of packets were classified as malicious.
    - A total of 1.22% of packets were correctly identified as RST DoS-specific packets.
  - False Negatives: 27.28% of malicious traffic misclassified as normal packets, highlighting areas for refinement in recognizing subtle attack traces.
- **MiTM Attacks:**
  - Detection Rate:
    - A total of 0.04% of packets were accurately classified as duplicate IP MiTM traffic.
    - A total of 0.02% were categorized as general malicious traffic.
  - False Positives: Minimal, with 99.95% of normal traffic correctly classified, indicating strong reliability in non-attack scenarios.
- **Brute Force Attacks:**
  - Detection Rate:
    - A total of 3.83% of packets were identified as malicious.
    - A total of 0.40% were recognized as RST/ACK brute force packets.
  - False Positives: Negligible, with 95.77% of normal traffic classified correctly, although attack detection rates suggest opportunities to enhance granularity.

6.5.2. Performance on Normal Traffic Datasets

The statistical model consistently achieved a 100% detection rate for normal traffic across all scenarios, with no false positives for malicious or attack-specific packets. This highlights its precision in handling benign datasets, making it a reliable tool for minimizing disruptions in normal operations.

6.5.3. Performance on Malicious Traffic Datasets

For malicious datasets, the model exhibited strong attack detection capabilities, although variation in classification rates was observed:

- MiTM attacks: 63.54% of malicious packets were detected, with 30.21% accurately linked to MiTM signatures.
- Brute force attacks: 90.42% of malicious packets were detected, with 9.56% specifically associated with brute force signatures.
- False negatives: A small portion of malicious packets in each scenario was misclassified as normal traffic, suggesting scope for refinement in edge cases.

6.5.4. Insights and Areas for Improvement

This study serves as the starting point for a broader research effort. While the current statistical model effectively detects attack-specific traffic and maintains high precision for normal datasets, certain enhancements recommended by the reviewers were not included in this phase due to scope constraints, dataset limitations, and computational feasibility. The primary objective of this study was to validate a foundational AMQP security model, ensuring an efficient, interpretable, and scalable approach. However, our ongoing research is actively working on integrating machine learning (ML) and deep learning (DL) techniques, which will address several of these recommendations. The following explains why certain aspects were not included and how future work will improve upon them:

- Advanced Metrics (Precision, Recall, AUC, Confusion Matrices): While incorporating advanced evaluation metrics would provide a more detailed assessment of model performance, this study prioritized practical detection measures—specifically detection accuracy and false positive/negative rates—to validate the feasibility of AMQP-based intrusion detection. Including additional metrics would have required expanded dataset validation and computational adjustments. However, our ongoing ML/DL research will inherently incorporate these advanced evaluation methods to ensure more robust performance assessment.
- Time-Series Analysis for Attack Evolution Detection: This study focused on static and event-based attack patterns, which are the most common in AMQP environments. While time-series analysis would provide deeper insights into evolving attack behaviors, it requires specialized modeling techniques, significantly larger datasets, and increased computational resources. Implementing time-series-based detection would have expanded the project's scope beyond its primary objectives. However, our next research phase will employ Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models to analyze long-term AMQP attack patterns, addressing this gap.
- Detection of Subtle Attack Patterns (e.g., Misclassified DoS Traces): The current model is designed to detect major attack categories in AMQP-based IoT networks. Identifying subtle attack patterns, such as low-rate DoS attacks misclassified as normal traffic, requires adaptive thresholding and more sophisticated anomaly detection methods. These refinements were excluded to maintain the study's focus on validating AMQP security at a fundamental level. However, our ongoing research explores deep learning-based anomaly detection techniques, such as autoencoders and Generative Adversarial Networks (GANs), to improve detection of stealthy, low-profile attacks.
- Granular Malicious Packet Classification for Brute Force and MitM Attacks: Improving the granularity of malicious packet classification requires more refined feature extraction and deep multi-label classification approaches. While the current study prioritizes broader attack categorization, our ongoing research is incorporating Convolutional Neural Networks (CNNs) and Transformer-based models to automatically extract deeper network features, enabling more precise attack classification. These enhancements were not included in this study due to dataset limitations and the need for additional training data, which are currently being curated.

While these enhancements were beyond the scope of this study, they form the core direction of our ongoing research. By leveraging ML/DL-driven methodologies, future iterations of this work will provide a more adaptive and intelligent framework for securing AMQP-based IoT environments.

6.5.5. Practical Implications

The model's ability to maintain high accuracy while minimizing false positives ensures its applicability in real-world IoT network environments. Key strengths include the following:

- Reliable differentiation between normal and malicious traffic.
- Effective detection of distinct attack signatures, supporting targeted threat mitigation.
- Robust performance across varied traffic scenarios, providing flexibility for deployment in dynamic IoT environments.

This analytical summary highlights the statistical model's strengths, identifies areas for further refinement, and reinforces its value as a tool for securing AMQP IoT protocols.

7. Conclusions

This study explored the security challenges of the AMQP protocol in IoT networks and introduced a statistical model for detecting and classifying malicious activities. By utilizing a purpose-built testbed, realistic scenarios of Denial-of-Service (DoS), Man-in-the-Middle (MiTM), and brute force attacks were simulated to generate extensive datasets of normal, malicious, and mixed traffic. Unique attack signatures were identified and rigorously validated, forming the basis for a practical and accurate detection mechanism tailored to the AMQP protocol.

The statistical model demonstrated robust performance in distinguishing normal and malicious traffic patterns with minimal false positives. For DoS attacks, the model reliably identified RST-specific packets, while for MiTM and brute force attacks, it achieved precise classification of attack-related traffic. These findings emphasize the effectiveness of signature-based approaches in bolstering the security and reliability of AMQP-driven IoT systems.

However, this research also highlights areas requiring further attention. While the model excelled in detecting known attack patterns, addressing more complex or subtle variations remains a challenge. To overcome these limitations, future work will focus on integrating advanced machine learning (ML) techniques capable of adapting to evolving threats. The statistical model developed in this study will serve as a benchmark for evaluating the performance of these ML-driven approaches, paving the way for a dynamic and comprehensive intrusion detection system (IDS).

By combining the precision of signature-based detection with the adaptability of ML techniques, future efforts aim to create a resilient security framework for AMQP protocols. This approach will enhance the protection of IoT ecosystems, ensuring their reliability and robustness against an increasingly sophisticated cyber threat landscape.

Author Contributions: Conceptualization, M.A. and A.K.; methodology, M.E.H., M.A., A.K. and G.N.; software, M.E.H.; validation, M.A., A.K. and G.N.; formal analysis, M.A.; investigation, M.E.H.; resources, M.E.H.; data curation, M.E.H. and M.A.; writing—original draft preparation, M.A.; writing—review and editing, M.A.; visualization, M.E.H.; supervision, M.A., A.K. and G.N.; project administration, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

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

References

1. Advanced Message Queuing Protocol (AMQP)—Complex Security. Available online: https://knowledge.complexsecurity.io/protocols/amqp/ (accessed on 1 January 2025).
2. National Vulnerability Database. CVE-2021-22116: RabbitMQ AMQP Server Denial-of-Service Vulnerability. Available online: https://nvd.nist.gov/vuln/detail/CVE-2021-22116 (accessed on 1 January 2025).
3. National Vulnerability Database. CVE-2018-11087: RabbitMQ AMQP Client Man-in-the-Middle (MitM) Vulnerability. Available online: https://nvd.nist.gov/vuln/detail/CVE-2018-11087 (accessed on 1 January 2025).
4. BruteMQ: An Exotic Service Bruteforce Tool. Available online: https://github.com/codexlynx/brutemq (accessed on 1 January 2025).
5. Alaiz-Moreton, H.; Aveleira-Mata, J.; Ondicol-Garcia, J.; Munoz-Castaneda, A.; Garcia, I.; Benavides, C. Multiclass classification procedure for detecting attacks on MQTT-IoT protocol. *arXiv* **2019**, arXiv:2402.03270. [CrossRef]

6. Amouri, A.; Alaparthy, V.; Morgera, S. A machine learning based intrusion detection system for mobile Internet of Things. *Sensors* **2020**, *20*, 461. [CrossRef] [PubMed]
7. Haripriya, A.; Kulothungan, K. Secure-MQTT: An efficient fuzzy logic-based approach to detect DoS attacks in MQTT protocols for the Internet of Things. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019* , 90. [CrossRef]
8. Liu, J.; Kantarci, B.; Adams, C. Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to the NSL-KDD dataset. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, Linz, Austria, 3 July 2020; pp. 25–30.
9. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network intrusion detection for IoT security based on learning techniques. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2671–2701. [CrossRef]
10. McAteer, I.; Malik, M.; Baig, Z.; Hannay, P. Security vulnerabilities and cyber threat analysis of the AMQP protocol for the internet of things. In Proceedings of the 15th Australian Information Security Management Conference, Perth, Australia 5–6 December 2017.
11. Gerodimos, A.; Maglaras, L.; Ferrag, M.; Ayres, N.; Kantzavelou, I. IoT: Communication protocols and security threats. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 1–13. [CrossRef]
12. Swamy, S.; Jadhav, D.; Kulkarni, N. Security threats in the application layer in IOT applications. In Proceedings of the 2017 International Conference On I-SMAC (IoT in Social, Mobile, Analytics And Cloud) (i-SMAC), Palladam, Tamil Nadu, India, 10–11 February 2017; pp. 477–480.
13. Özalp, A.; Albayrak, Z.; Çakmak, M.; ÖzdoĞan, E. Layer-based examination of cyber-attacks in IoT. In Proceedings of the 2022 International Congress On Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 9–11 June 2022; pp. 1–10.
14. Masoodi, F.; Alam, S.; Siddiqui, S. Security & privacy threats, attacks, and countermeasures in the Internet of Things. *Int. J. Netw. Secur. Appl.* **2019**, *11*, 1–19.
15. Nebbione, G.; Calzarossa, M. Security of IoT application layer protocols: Challenges and findings. *Future Internet* **2020**, *12* , 55. [CrossRef]
16. Andy, S.; Rahardjo, B.; Hanindhito, B. Attack scenarios and security analysis of MQTT communication protocol in IoT systems. In Proceedings of the 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 19–21 September 2017; pp. 1–6.
17. Tariq, M.; Khan, M.; Kim, D. Enhancements and challenges in CoAP. *Sensors* **2020**, *20*, 6391. [CrossRef] [PubMed]
18. Thamer, A.; Aboubaker, L.; Mahdi, A. Security analysis of the constrained application protocol in the Internet of Things. In Proceedings of the Second International Conference on Future Generation Communication Technologies (FGCT 2013), London, UK, 12–14 December 2013.
19. Vaccari, I.; Aiello, M.; Cambiaso, E. SlowITe, a novel denial of service attack affecting MQTT. *Sensors* **2020**, *20*, 2932. [CrossRef]
20. Dinculeană, D.; Cheng, X. Vulnerabilities and limitations of MQTT protocol used between IoT devices. *Appl. Sci.* **2019**, *9*, 848. [CrossRef]
21. Karagiannis, V.; Chatzimisios, P.; Vazquez-Gallego, F.; Alonso-Zarate, J. A survey on application layer protocols for the internet of things. *Trans. IoT Cloud Comput.* **2015**, *3*, 11–17.
22. Santhosh Kumar, S.; Selvi, M.; Kannan, A. A comprehensive survey on machine learning-based intrusion detection systems for secure communication in the Internet of Things. *Comput. Intell. Neurosci.* **2023**, *2023*, 1–24. [CrossRef]
23. Glaroudis, D.; Iossifides, A.; Chatzimisios, P. Survey, comparison, and research challenges of IoT application protocols for smart farming. *Comput. Netw.* **2020**, *168*, 107037. [CrossRef]
24. Asharf, J.; Moustafa, N.; Khurshid, H.; Debie, E.; Haider, W.; Wahab, A. A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics* **2020**, *9*, 1177. [CrossRef]
25. Aiash, M. AMQP Network Traffic Dataset: Normal and Malicious IoT Communications. *Netw. Eng.* **2025**. [CrossRef]

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