Enhancing Data Security

In IoT Deployments

Thank you for downloading this HiveMQ whitepaper. Carahsoft is the master government aggregator for HiveMQ IoT solutions available via OMNIA, IPHEC, and other contract vehicles.

To learn how to take the next step toward acquiring HiveMQ's solutions, please check out the following resources and information:

For additional resources:
carah.io/hivemq-resources

For upcoming events:
carah.io/hivemq-events

For additional HiveMQ solutions:
carah.io/hivemq-solutions

For additional IoT solutions:
carah.io/IoT-Solutions

To set up a meeting:
HiveMQ@carahsoft.com
571-591-6210

To purchase, check out the contract vehicles available for procurement:
carah.io/hivemq-contracts

For more information, contact Carahsoft or our reseller partners:
HiveMQ@carahsoft.com | 571-591-6210

HIVEMQ

Enhancing Data Security

in IoT Deployments

Technical White Paper
January 2024

TABLE OF CONTENTS

Introduction

The number of connected IoT devices is increasing rapidly across many industries worldwide and is expected to reach 29.7 Billion by the end of 2027, according to IoT Analytics. The growth is expected predominantly in consumer-focused industries like the internet and media such as smartphones, along with connected vehicles, IT infrastructure, asset tracking and monitoring in transportation and logistics, and energy smart grids.

While this explosive growth signifies progress toward a connected world, it also raises concerns about the security of these devices and the overall ecosystem. **This whitepaper discusses the common data security challenges in IoT deployments and describes the methods organizations can adopt to enhance their IoT deployment security posture.** It also highlights how HiveMQ's upgraded Enterprise Security Extension (ESE) addresses these challenges to enhance the overall security posture of IoT deployments.

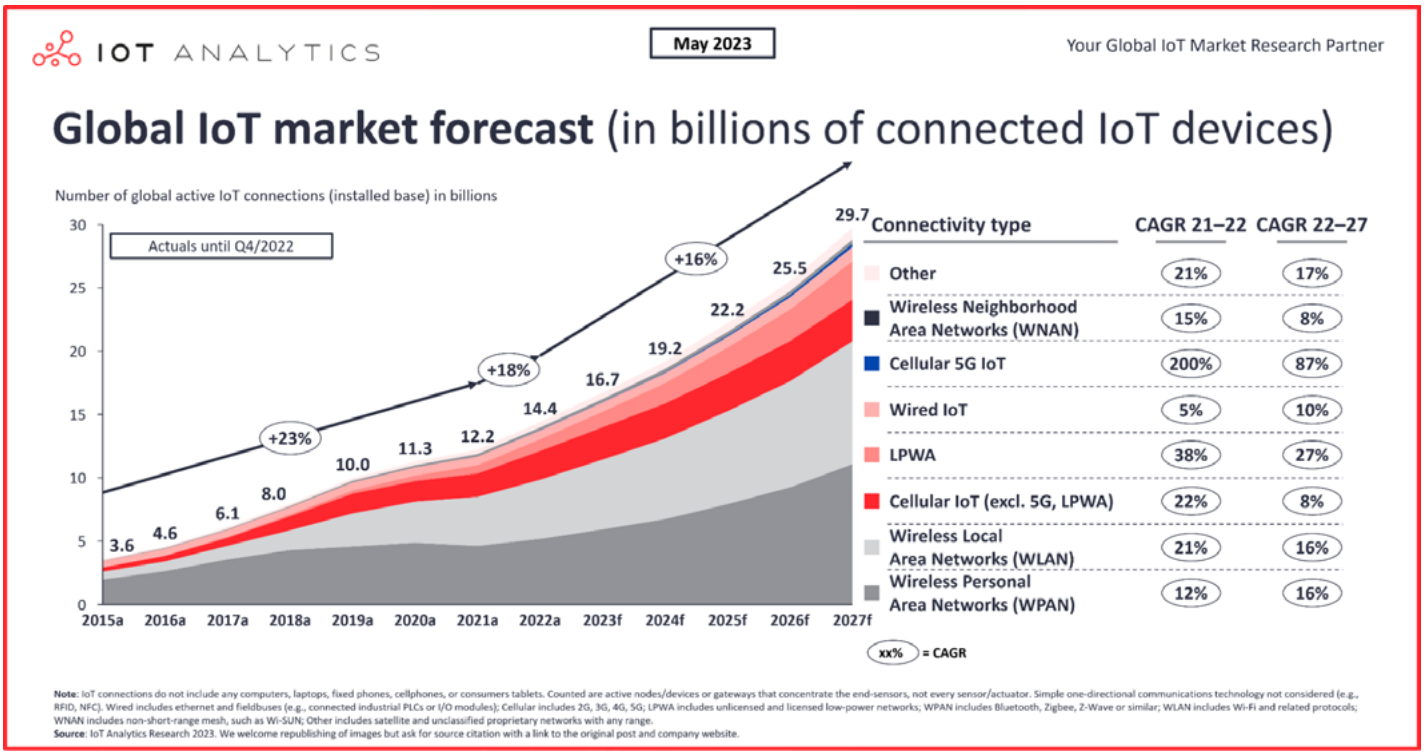
Let's dive into the five most common data security challenges in IoT deployments, how to solve them, and how HiveMQ can help.

Inadequate Authentication and Authorization of IoT Devices

Challenge: Do I Have Adequate Authentication and Authorization Controls to Strengthen the Security of My IoT Deployment?

Securing IoT deployments requires critical attention to device authentication and authorization. Malicious actors may attempt to create virtual devices, infiltrate deployments, and disseminate harmful data across the network. For example, they could exploit vulnerabilities to gain unauthorized access to other devices in the deployment, potentially taking control of critical devices such as industrial machinery, medical equipment, or smart infrastructure.

Therefore, organizations must have appropriate authentication and authorization controls in place to protect access to devices and brokers in their IoT deployment.



Source

	Authentication	Authorization
Purpose	• Verifies user identity	• Permits access to resources
Requirements	• Identity credentials based on knowledge, possession, and/or inherence • Authentication solution	• Authenticated identity and access control policies • Authorization solution
Responsibilities	• Network security staff determine which factors to adopt • Users provide authentication factors when requesting access	• Leadership sets security strategies • Departments and workgroups define access criteria • Network security staff implement and maintain access control system

How to Solve It?

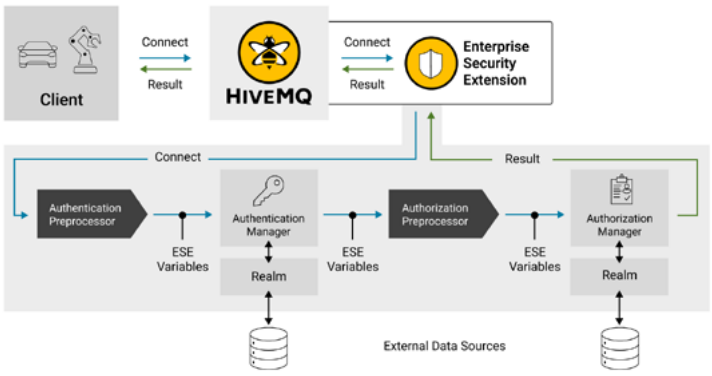
Organizations can employ robust authentication mechanisms to verify the identity of devices and users accessing the IoT network. For instance, to authenticate users, multi-factor authentication (a.k.a MFA or 2-FA) is a commonly used mechanism involving authentication by multiple sources. This can be via unique email or SMS codes.

For devices, security certificates or token-based systems are used to enhance identity verification by requiring the presentation of valid and unique credentials. Security certificates, commonly based on X.509 standards, authenticate devices through cryptographic keys, ensuring a secure and trusted connection. Token-based systems utilize dynamically generated tokens, often incorporating time-sensitive or one-time-use elements, providing an additional layer of authentication.

These mechanisms work in tandem to establish a robust identity verification process, thwarting unauthorized access and fortifying the overall security of the IoT deployment.

How Does HiveMQ Address Inadequate Authentication and Authorization?

One of the key functions of the Enterprise Security Extension (ESE) is the authentication and authorization of MQTT clients. HiveMQ ESE enhances HiveMQ Broker's capabilities by using external data sources for user and permission management, allowing authentication and authorization of MQTT clients. With the extension, you can also partition your server into realms, each with its own authentication and authorization scheme. This extension also allows you to control the way client connections are handled. You can customize the process to include authentication and authorization stages for your clients.



Authentication

The authentication process is built to verify whether a person, device, or application is who they say they are. In the ESE, authentication is overseen by "authentication managers." These managers utilize authentication variables, which can be string or byte data variables, to verify the identity of the client attempting to connect.

The ESE offers a highly flexible authentication system that can work with various sources of identity information. It can extract information from different sources, such as X.509 client certificates, Scopes from JWT tokens, and MQTT packets. This information is then used to verify identity against external identity systems such as databases, OAuth/OIDC servers, LDAP, and more.

Authorization

On the other hand, authorization defines which actions a specific person, device, or application is allowed to perform, and it can be split up into the concepts of roles and permissions. Authorization in the ESE is managed by "authorization managers." These managers handle the authorization of MQTT clients, including fine-grained access for topics, activities (publish/subscribe), QoS levels (retain/allowed), and shared subscriptions (allowed, groups, etc.). Learn more about HiveMQ's authorization controls.

The ESE authorizes and authenticates MQTT clients for IoT deployments and provides access to the HiveMQ Control Center, a web-based tool to monitor and manage HiveMQ MQTT brokers.

The ESE employs multiple methods for authentication and authorization purposes, including:

- SQL databases: As an external source of authentication and authorization data.
- OAuth 2.0 / JWT for MQTT Clients: MQTT clients can authenticate using JSON Web Tokens (JWT), which can be integrated into your MQTT infrastructure.
- OpenID Connect (OIDC): For HiveMQ Control Center Login, ESE integrates HiveMQ Control Center with OpenID Connect to enable enterprise-wide SSO capabilities.
  - Two-factor authentication (2-FA) is a part of OIDC and allows for a two-step authentication process using mechanisms such as SMS or one-time passwords (OTP).
  - HiveMQ customers also use external authentication platforms like Okta, Google AD, and Azure AD to authenticate users and devices.
- Support for any standard-compliant LDAP server: LDAP is an open standard protocol for interacting with directory servers. It is widely used to authenticate and store user, group, and application information. For example, the ESE supports the following LDAP servers:
  - Microsoft Active Directory Domain Services (AD DS) and Microsoft Active Directory Lightweight Directory Services (AD LDS)
  - OpenLDAP
  - Apache Directory Studio (ApacheDS)
- File-based authentication and authorization: This is a quick and easy way to set up the ESE to secure MQTT clients or the Control Center without the steep learning curve of dealing with a full-blown DBMS or an OAuth provider. It can also handle hashed passwords and provide dynamic reloading, enhancing security by securely storing and managing credentials while enabling real-time updates without service interruption

Fine-Grained Access Control

Challenge: How Can I Protect My Entire IoT Deployment from Malicious Actors Who Gain Access to Authorized Devices?

Protecting your entire IoT deployment from malicious actors who might gain access to authorized devices is crucial because these actors might manipulate data sent from IoT devices, steal or siphon critical operational data from IoT devices (data breaches), and potentially disrupt critical operations. Failure to address this challenge may lead to compromised integrity, privacy violations, and compromised functionality, risking the overall security and reliability of the IoT deployment.

How to Solve It?

Fine-grained access controls via Role-Based Access Control (RBAC) and user permissions help reduce the "attack surface area" in an IoT deployment. Organizations can limit the actions authorized devices can perform by assigning specific roles and permissions to users. In the event of a malicious actor gaining control of a trusted device, the impact is mitigated because the compromised device has restricted capabilities based on its assigned role and permissions.

For instance, a compromised device may have access only to a subset of IoT functionalities or data, preventing unauthorized manipulation of critical systems. RBAC ensures that even if one device is compromised, it cannot execute high-impact actions or access sensitive information beyond its defined role boundaries. This approach minimizes the

potential damage and restricts the lateral movement of malicious actors within the IoT network.

How Does HiveMQ Enable Fine-Grained Access Control via RBAC and User Permissions?

The ESE adds RBAC capabilities to the HiveMQ MQTT Broker. The ESE allows users to define roles and associated permissions, enabling fine-grained access control for MQTT clients.

The HiveMQ Broker features HiveMQ Control Center, a web-based administration tool designed to offer real-time insights into the health and performance of the MQTT broker.

Access Controls for the Control Center

HiveMQ offers access control for the HiveMQ Control Center through its ESE extension. You can use fine-grained access control based on user permissions or role-based access control (RBAC).

HiveMQ REST API

The HiveMQ REST API is an interface that allows users to interact programmatically with the HiveMQ MQTT broker over HTTP. It enables actions such as managing clients, subscriptions, and accessing metrics, providing a way to integrate and automate various aspects of the MQTT broker.

Access Controls for the REST API

HiveMQ features access controls for its REST API that allow you to restrict user access in several ways. For example:

1. Authenticate your REST API users with username and password.
2. Manage authenticated access to the HiveMQ REST API per endpoint.
3. Configure fine-grained access control for individual users and user roles.

Role-Based Access Controls for the REST API

Implementing role-based access control can make permission management more straightforward and improve performance. To provide precise access control for your users to the resources available through the HiveMQ REST API, you can assign particular permissions to users and roles. By granting specific REST API permissions to a role, all users with that role will be given the same permissions.

Insufficient Encryption

Challenge: How Can I Ensure Adequate Encryption for Data Transmitted Between IoT Devices and Backend Systems?

Having adequate data encryption in IoT deployments is crucial to safeguard sensitive information and maintain the integrity of communication channels. Without robust encryption measures, intercepted data poses a significant risk of unauthorized access and manipulation, potentially leading to data breaches, privacy violations, or compromise of critical systems.

How to Solve It?

Transport Layer Security (TLS) provides robust encryption in MQTT-based IoT deployments. When an MQTT client connects to HiveMQ Broker using TLS, a secure channel is established through a handshake process where encryption parameters and digital certificates are exchanged. This process authenticates both the client and the broker, creating a shared secret key for secure data transmission. TLS encrypts the data exchanged between MQTT clients and the broker, safeguarding against unauthorized access and ensuring the confidentiality and integrity of sensitive information.

How Does HiveMQ Ensure Adequate Encryption in IoT Deployments?

HiveMQ employs TLS to encrypt MQTT communication, ensuring a secure channel for data exchange between clients and the broker. The TLS handshake initiates when a client connects, negotiating encryption parameters such as algorithms and key exchange. Digital certificates are exchanged for mutual authentication, with the broker's certificate typically signed by a trusted Certificate Authority. The TLS handshake establishes a shared secret key that is used to encrypt the data transmission between clients

and the broker. HiveMQ also supports TLS versions 1.2 and 1.3, with configurable cipher suites to define acceptable encryption algorithms.

Below are some additional features in the ESE that strengthen security in IoT deployments:

- X.509 certificates: The extension provides preprocessors for authentication and authorization pipelines that enable sophisticated security use cases. It also supports a broader range of fields that can be extracted from X.509 certificates. In some scenarios, customers often need to use fields flexibly to generate information for the authentication and authorization of users. This flexibility is particularly valuable in addressing diverse customer scenarios that require nuanced authentication and authorization processes.
- Support cipher suites: These are sets of encryption, authentication, and key exchange algorithms used to secure network communications. You can limit HiveMQ to use specific cipher suites to meet enterprise security compliance policies.
- OpenSSL: OpenSSL is an open-source software library that provides a toolkit for implementing TLS/SSL protocols. It offers a wide range of cryptographic functions and utilities, making it a versatile tool for securing network communication, data integrity, and cryptographic operations. HiveMQ comes prepackaged with an OpenSSL implementation.

Device Identity Management

Challenge: How Do I Manage and Secure Access Credentials for Each IoT Device?

Effectively managing and securing access credentials for individual IoT devices in your deployment is imperative to prevent unauthorized devices from accessing your IoT deployment or malicious actors from gaining access to authorized devices in your IoT deployment. Without a robust credential management system, compromised credentials could lead to malicious actors manipulating critical operational data or controlling critical IoT network devices.

How to Solve It?

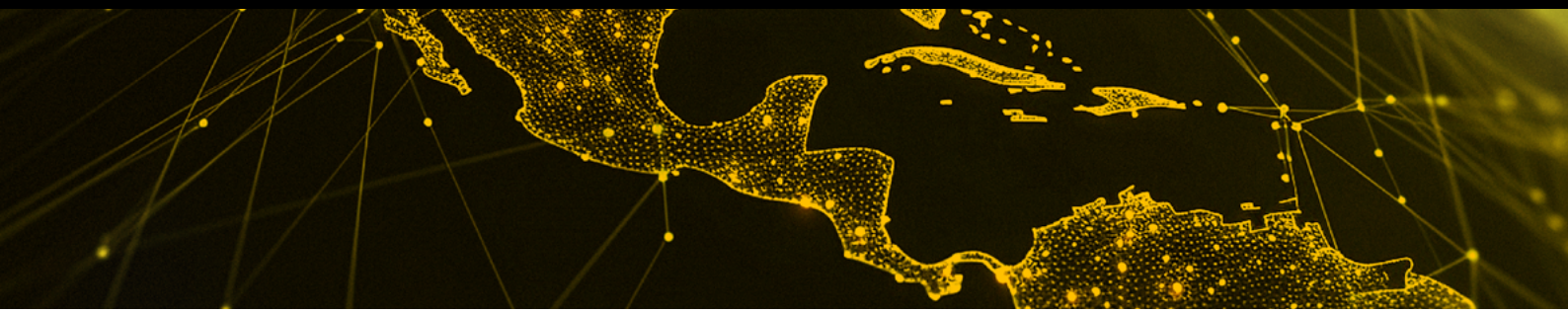
Implement a robust device identity management system to prevent spoofing or unauthorized device access. Such a system involves assigning unique and tamper-resistant identities to each device within the IoT network. This includes incorporating secure protocols, such as device certificates or unique identifiers, and implementing robust authentication mechanisms. Additionally, regular monitoring and updates to device identity information are crucial to maintaining the security of the devices in your IoT deployment.

How Does HiveMQ Help with Device Identity Management?

The Enterprise Security Extension (ESE) allows you to configure OCSP (Online Certificate Status Protocol) and CRL (Certificate Revocation List) client certificate revocation checking during the TLS handshake for secure TCP listeners and secure WebSocket listeners.

Here is how client certificate revocation checking helps with device identity management:

- Supports TLS handshake client certificate revocation checking with OCSP (Online Certificate Status Protocol) and CRL (Certificate Revocation List) for TLS listeners to the HiveMQ Enterprise Security Extension.
- In ESE, you can configure one or more revocation checks and reference TLS listener configurations in the broker.
  - When a client/device connects to HiveMQ Broker, certificates undergo checks against a list of revoked certificates. This revocation mechanism is crucial for IoT deployments utilizing certificates in securing client-to-broker communication, allowing the proactive revocation of client certificates before their natural expiration.



- By utilizing Certificate Revocation Lists (CRL) or the Online Certificate Status Protocol (OCSP), the certificates of clients or devices are cross-referenced with a list of revoked certificates. If a match is identified, connection attempts are denied.

Centralizing access control management for connected car applications is becoming increasingly important as it can significantly enhance security. With central access control management, ensuring that only authorized clients can access the vehicle network is easier.

Inadequate Logging and Monitoring

Challenge: How Can Organizations Monitor and Detect Security Incidents in Real-Time?

Real-time monitoring enhances the ability to swiftly detect abnormal activities, intrusions, or unauthorized access, allowing for immediate intervention and mitigation measures to maintain the security and integrity of the IoT ecosystem.

How Can It Be Solved?

Implement comprehensive logging, monitoring, and tracing systems to record, monitor, and track suspicious activities in real-time or near real-time. In a comprehensive security strategy, the continuous collection and analysis of logs, coupled with real-time monitoring and tracing capabilities, create a powerful defense against potential threats. The synergy of these systems allows administrators to identify deviations from standard patterns and trace the origins of variations back to specific internal processes or client interactions.

How Does HiveMQ Help with Inadequate Logging and Monitoring?

Logging

HiveMQ implements a powerful logging system that helps you monitor, diagnose, and troubleshoot your MQTT-based IoT applications. The audit logs capture critical events — including client connections, subscription changes, and published messages — offering visibility into the MQTT broker's activity. It provides a single, unified log for tracking audit-relevant data.

Access logs detail client interactions, providing insights into who accessed the broker and when. The access log provides a unified record for tracking security data. It includes details such as who accessed HiveMQ, the type of access, and when it was granted. With access logs, you can:

- Audit all accesses that the ESE grants retroactively.
- Configure your intrusion-prevention software (e.g. Fail2Ban) to use access logs to create firewall rules.
- Use chronological access logs (in the event of a data breach) to extract valuable information to aid in the post-mortem process.

HiveMQ also features:

- Application logs: Play a crucial role in detecting and responding to security incidents by providing a detailed record of client interactions and MQTT-related activities. Application logs also capture key events, error messages, and operational details within HiveMQ Broker.

- Trace recordings: Trace recordings go beyond typical application logs by providing detailed information about events, timing, and execution paths, enabling administrators to conduct in-depth analyses during troubleshooting or performance optimization.

- Client Event logging: These logs capture important events related to MQTT clients, including connection attempts, disconnections, subscription changes, and published messages. Client event logs offer administrators a focused view into the behavior of individual clients, detect anomalies, and track the lifecycle of client connections.

Integration with HiveMQ Control Center

The logs generated by HiveMQ can be accessed and analyzed through the HiveMQ Control Center. This integration gives administrators a centralized view of logs for efficient monitoring and troubleshooting. The Control Center also provides client insights like connected clients, including their status, connections, and subscriptions. This allows administrators to have visibility into the behavior of individual clients in real time. The Control Center provides real-time monitoring of MQTT topic usage, including subscription rates and message distribution. This enables administrators to understand the data flow and usage patterns within the MQTT broker.

Combined, these features enable a comprehensive approach to monitoring and managing MQTT environments.

Monitoring

Monitoring your MQTT brokers is vital, especially in clustered environments. HiveMQ supports several metrics to monitor critical aspects of the system, ensuring its stability and resilience. Some of these metrics are:

- Connection Rates: Monitoring the rates of client connections provides insights into the overall activity and demand of the MQTT broker.

- Message Rates: Tracking the message publication and consumption rates helps assess the data flow within the broker, ensuring efficient communication.

- Client Insights: Monitoring connected clients, their status, connections, and subscriptions allows administrators to see individual client behaviors in real-time.

- Topic Insights: Real-time monitoring of MQTT topic usage, including subscription rates and message distribution, helps understand data flow and usage patterns within the MQTT broker.

If you use HiveMQ in critical infrastructure, it is highly recommended to use an appropriate monitoring application such as Prometheus or InfluxDB. HiveMQ has a purpose-built Prometheus Monitoring Extension that allows HiveMQ to expose metrics to a Prometheus application. HiveMQ also has an InfluxDB Monitoring Extension that enables HiveMQ to connect to an instance of InfluxDB for time-series monitoring.

Tracing

Through its Enterprise Distributed Tracing Extension, HiveMQ Broker offers trace recordings and deep tracing capabilities. Trace recordings capture intricate details of MQTT broker internals, aiding in identifying security vulnerabilities and anomalies. The Distributed Tracing Extension enhances HiveMQ's capabilities, providing a detailed trace of messages across the entire IoT deployment. This precision empowers administrators to manage and secure IoT environments proactively.

Tracing, logging, and monitoring provide a comprehensive view of the message lifecycle and client interactions. This facilitates real-time detection and response to security incidents.

Conclusion

As IoT deployments proliferate and scale, IT architects, heads of digital transformation, and other professionals must clearly understand the risks associated with IoT deployments and have a set of well-tested strategies to prevent security breaches effectively. Ignoring the risks and not having a robust security posture for IoT deployments can lead to serious consequences that are too severe to overlook.

HiveMQ's comprehensive suite of security features, coupled with real-time monitoring capabilities, positions it as a powerful solution to fortify the security landscape of MQTT-based IoT deployments. As organizations navigate the complexities of IoT security, the HiveMQ platform stands as a reliable ally in safeguarding devices, data, and the integrity of the IoT ecosystem.

About HiveMQ

HiveMQ empowers businesses to transform with the most trusted MQTT platform. Designed to connect, communicate, and control IoT data under real-world stress, the HiveMQ MQTT Platform is the proven enterprise standard for Industry 4.0. Leading brands like Audi, BMW, Liberty Global, Mercedes-Benz, Siemens, and ZF choose HiveMQ to build smarter IIoT projects, modernize factories, and create better customer experiences.

Visit hivemq.com to learn more

HIVEMQ

www.hivemq.com