An Analysis of IoT Protocols and the Security Therein

Harris McCall

Columbia Basin College

CSIA300

Matt Boehnke

December 1, 2023

The Internet of Things, frequently abbreviated as “IoT”, is growing exponentially each year. Thousands of new IoT Devices are connected to networks and the internet each year. In fact, the global IoT market is estimated to reach 2.465 trillion USD by 2026 (Moraes, 2023). IoT presently active in many different industries such as: Agricultural, Health Care, Environmental, Maritime, Military, or even City development (Gerodimos et al., 2023). If insecure devices are produced and sold to businesses, that creates vulnerabilities, or entry points for attackers, into their business network architecture. This means that standardized protocols at the Application layer are in high demand. A comprehensive assessment of the prevailing IoT communication protocols reveals an environment shaped by the necessity for interoperability, efficiency, and low overhead. This essay analyzes these protocols, with a particular focus on Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), and Extensible Messaging and Presence Protocol (XMPP), in the context of their architectural frameworks, operational mechanics, and inherently linked security features. It endeavors to underscore the importance of robust security measures, including encryption, authentication, and access control, which are paramount to safeguarding the growing IoT ecosystem from the escalating sophistication of cyber threats, thus ensuring the confidentiality, integrity, and availability of interconnected devices and the data they exchange.

Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), and Extensible Messaging and Presence Protocol (XMPP) are Application layer protocols that are particularly used for directing the flow of data (Iqbal et al., 2023). Because these protocols are on the same OSI layer, their purposes are extremely similar, though they interoperate with different sets of other protocols, both network and transport, for added security layers. Let’s get an overview of what that looks like.

MQTT is a lightweight, publish-subscribe messaging protocol invented in 1999 (Iqbal et al., 2023) for use in situations where efficiency and low bandwidth are crucial, especially in IoT and Machine-to-Machine (M2M) cases. Publish-subscribe is a concept that allows certain devices to "publish" information about a specific topic, while other devices can choose whether or not to "subscribe" and receive information from the publishing devices about chosen topics (AWS, n.d.). It primarily operates over TCP/IP and other ordered, lossless, bi-directional connections. With a focus on delivering secure and reliable communication, MQTT supports Quality of Service (QoS) levels and provides security features. Its publish-subscribe messaging pattern is well-suited for IoT environments, allowing devices to subscribe to specific topics and receive relevant data when published. Additionally, MQTT can be integrated with WebSockets, enabling seamless communication within web-based applications. These characteristics make MQTT a versatile protocol for various applications, especially those requiring low bandwidth, minimal latency, and dependable connectivity. (OASIS, 2019).

MQTT security is primarily dependent on the TLS/SSL to provide transport encryption, which helps to safeguard against eavesdropping. Additionally, the MQTT application provides client identifier and username/password verification, which can be utilized for device authentication. However, the primary disadvantage of MQTT security is the same use of TLS/SSL, which may not be suitable for constrained devices. As a more flexible, lightweight, and robust security mechanism is needed, researchers have proposed a secure MQTT (SMQTT) that extends security features using lightweight attribute-based encryption (ABE) over elliptic curves, allowing for broadcast encryption and fine-grained access control suitable for IoT applications (Panchiwala & Shah, 2020). Another limitation of the use of TLS/SSL is that it is slow over the TCP connection, and the UDP connection is unreliable. A potential solution to this is to use the QUIC protocol, which combines the speed expected from UDP with the reliability of TCP. (Iqbal et al., 2023)

CoAP is an IETF protocol, defined in RFC 7252. It is a specialized web transfer protocol designed for use with constrained nodes and networks, such as those with low power and limited resources present in IoT devices. It enables resource discovery along with service and resource discovery, making it possible for devices to query a server for the list of hosted resources. Designed for machine-to-machine (M2M) applications such as smart energy and building automation, CoAP allows devices to interact efficiently. Additionally, it supports proxying requests on behalf of another CoAP endpoint and caching responses for efficient request fulfillment. With a focus on reducing overhead, the protocol maintains low header overhead and parsing complexity, making it suitable for constrained environments. Moreover, CoAP is designed to easily interface with HTTP for integration with the existing web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments and M2M applications. (Shelby, Hartke, & Bormann, 2014)

Constrained Application Protocol (CoAP) security usually relies on Datagram Transport Layer Security (DTLS) to provide security features such as transport encryption and message integrity. DTLS is a variation of the Transport Layer Security (TLS) protocol specifically designed to secure datagram communications. By utilizing DTLS, CoAP ensures secure communication by providing encryption, data integrity, and endpoint authentication, making it suited for use in constrained environments typical of IoT devices (Panchiwala & Shah, 2020). IPSec is an acceptable alternative to DTLS, as it provides authentication and encryption at the IP level. Though, if neither DTLS or IPSec are used, the data flow from the IoT device is easy to access, and the IoT system is completely open to the internet and easy for an attacker to compromise (Meneghello et al., 2019).

XMPP is an open standard protocol that uses Extensible Markup Language (XML) the core of which is defined in IETF RFC 6120. It enables near-real-time exchange of structured and extensible data between network entities. It typically operates in a client-server architecture and uses XML streams to exchange data. The process involves setting up a TCP connection, opening an XML stream, optionally negotiating encryption, authenticating using SASL mechanisms, binding a resource, exchanging XML stanzas, and then closing the XML stream and the TCP connection. XMPP is designed for asynchronous communication and can handle messaging, network presence, and request-response interactions. (Saint-Andre, 2011)

XMPP utilizes several security measures, such as Transport Layer Security (TLS) for encryption, Simple Authentication and Security Layer (SASL) for authentication, and end-to-end encryption through extensions such as OMEMO and OpenPGP (Iqbal et al., 2023). XMPP security principles encompass several aspects to ensure secure communication. End-to-end security proposals for XMPP, such as OMEMO and OpenPGP for XMPP, prioritize authenticity, integrity, and encryption while offering forward secrecy. These solutions align with various communication patterns, including one-to-one messaging, group chat, and offline messaging functionalities. The comparative overview of these proposals addresses their compatibility with XMPP, emphasizing the need for authentication and protection from message tampering. Legacy solutions like XEP-0027 (Legacy OpenPGP) and OTR are discouraged due to security flaws or lack of standardization, pointing to the community's ongoing focus on refining and implementing robust end-to-end security measures within the XMPP ecosystem (XSF, 2018).

The Application Layer of IoT architecture confronts a spectrum of common security threats inherent to its diverse and expansive nature. Cross Site Scripting attacks, Malicious Code Attacks, and Cinderella Attacks are just a few concerning types of attacks that can affect IoT devices. Cross Site Scripting (XSS) Attacks occur when adversaries can inject malicious code scripts into trusted domains, allowing them to manipulate the contents of an application accessible to multiple users. This intrusion can result in the compromise of sensitive data or identity theft, thereby raising concerns about the confidentiality and integrity of the system. Similarly, Malicious Code Attacks present a significant challenge. These types of attacks involve the introduction of various malicious software elements such as Trojans, Viruses, Worms, or Backdoors, potentially leading to undesired system effects and compromising data integrity. Such infiltration undermines the trust and reliability of the IoT ecosystem, necessitating strong measures to identify and mitigate these attacks. (Gerodimos et al., 2023)

Cinderella Attacks are another notable security concern at the Application Layer. These attacks involve malicious entities manipulating the internal clock of the network, resulting in the false premature expiration of security software. This activity increases the network's vulnerabilities, posing a threat to the overall integrity of the IoT system. Moreover, the challenge of effectively handling big data within the network infrastructure presents an additional security concern. The large volume of data generated by IoT devices, coupled with limitations in the network's hardware to process this data efficiently, can lead to network disturbances and potential data losses. This poses security vulnerabilities, as sensitive information may be at risk if not handled appropriately. (Gerodimos et al., 2023)

When considering the security of an IoT device, physical security should also be in question. IoT devices are often considered weak at the Physical Access Level due to several factors. Many IoT devices are designed with minimal physical security features, lacking strong enclosures or tamper-resistant designs, making them vulnerable to physical breaches and attacks. Additionally, these devices are commonly deployed in environments like public spaces or outdoor locations, where they may be more susceptible to unauthorized physical access. Some IoT devices have limited or weak authentication mechanisms at the physical level, such as easily bypassed or default security settings, making it easier for unauthorized individuals to gain physical access. Moreover, many IoT devices may not include physical intrusion detection mechanisms, making it difficult to detect unauthorized attempts to access or tamper with the device physically. Due to resource constraints, manufacturers may prioritize other aspects of functionality over physical security features, leading to vulnerabilities at the physical access level. In some cases, ongoing physical security oversight and management of IoT devices in the field may be limited or overlooked, leaving them vulnerable to unauthorized physical access over time. Addressing these vulnerabilities requires a holistic approach, which may involve improving physical security design, implementing stronger authentication mechanisms, integrating physical intrusion detection mechanisms, and enhancing ongoing maintenance and oversight processes. (Panchiwala & Shah, 2020)

The application layer protocols on their own do not provide sufficient baseline security and depend on other protocols to improve the security of their data handling. To enhance the security of IoT protocols at the Application Layer and mitigate the range of security threats, it should be a priority to incorporate multiple layers of defense. The design and implementation should prioritize robust detection systems, employing strong authentication measures, and establish comprehensive access control mechanisms to manage and restrict network access efficiently. (Gerodimos et al., 2023)

Privacy by design must be a foundational principle to ensure data integrity and confidentiality, protecting against unauthorized access and data breaches, while also providing for user anonymity and secure, confidential information flow. The inclusion of lightweight cryptographic methods, secure hardware implementations, and diligent application and firmware update processes are key to reinforcing system resilience. Moreover, a system's capacity for self-organization helps maintain operational security amidst attacks or infrastructure failures. Successfully integrating these measures necessitates continuous refinement of security standards and practices, encapsulating software, hardware, and organizational protocols to uphold the integrity, confidentiality, and overall robustness of IoT applications, thus cultivating a more secure, trustworthy IoT ecosystem. (Meneghello et al., 2019)

Using robust measures of authentication, as mentioned, is crucial to enhance the security of IoT applications. Several methods of authentication are recognized as suitable for IoT devices, including Mutual Authentication, Role-Based Access Control, Certificate Based Authentication, Strong Passwords, and Two Factor Authentication. As a general rule, using two or more authentication methods together increases security. This is the concept behind Two Factor Authentication. Also called Multi-Factor Authentication, methods of authentication are broken down into: things you have, things you know, or things you are. Any combination of 2 or more of those things is considered Multi-Factor Authentication. Furthermore, anytime a password is required it should not remain the default password and it should be between 8-12 alphanumeric characters, including special characters like punctuation (Gerodimos et al., 2023). Neither MQTT, CoAP, or XMPP natively support Multi-Factor Authentication, however it can be enabled at the application layer using additional protocols such as OAuth or other plugins/systems.

Access Controls, whether Role-Based, Discretionary, or Attribute-Based, should be used wherever possible. Role-Based Access Control allows access based on a role the requester has previously been assigned while Discretionary Access Control allows for the administrator to allow access to specific resource requesters and Attribute-Based Access Controls grant access based on policies using the properties of the user, requested asset, and the environment (Panchiwala & Shah, 2020). MQTT allows for Attribute-Based Access Controls by allowing the broker to specify which clients have access to publish or subscribe to particular topics (OASIS, 2019).

Another method of authentication that is important to integrate into IoT systems is Mutual Authentication. Mutual authentication refers to a security process where both communicating parties authenticate each other's identities. In this process, not only does the client authenticate the server, but the server also authenticates the client. This ensures that both parties can trust each other's identities and validates that each is communicating with the intended and legitimate entity (Bonetto et al., 2012). Methods of mutual authentication commonly include the use of digital certificates, shared keys, and biometric credentials. Digital certificates validate the identity of devices and users within an IoT environment. Shared keys, also known as symmetric keys, involve both parties possessing and using the same key for authentication. Biometric credentials utilize unique biological characteristics such as fingerprints, iris scans, or facial recognition to authenticate individuals within the system. These methods help ensure that both parties in a communication exchange can verify each other's identity, enhancing the overall security of the IoT environment (Panchiwala & Shah, 2020). CoAP uses a form of mutual authentication called certificate-based authentication and can use

Pre-Shared Keys, Raw Public keys, or a Public Key Infrastructure to perform authentication (Shelby, Hartke, & Bormann, 2014).

An additional important practice to increase security in IoT devices is to segment networks to prevent intruders from making lateral movements across the network. An effective way to achieve this is through the implementation of Zero Trust Network Access (ZTNA). This approach provides user-to-app level segmentation, offering secure access to unmanaged devices and keeping unknown users off networks, thus curbing the risk of malware proliferation. ZTNA is designed to scale effectively across cloud environments, configure access policies, and maintain network performance, thereby taking a proactive stance to enhance security across IoT ecosystems. Furthermore, LAN micro-segmentation can be utilized to implement client isolation, exercise heightened control over trusted cloud and remote services interacting with IoT solutions and impose finer network and device controls. By enforcing inbound and outbound traffic monitoring and prohibiting third-party remote access, solid network segmentation practices can greatly bolster security across IoT environments. Organizations should also ensure that patches are promptly applied to fix vulnerabilities and bugs, maintaining close contact with vendors to incorporate security requirements into contracts. Additionally, performing a cyber risk assessment of existing solutions before integrating new systems and products into the cybersecurity portfolio enhances overall security posture. Lastly, a unified authorization layer based on an OAuth 2.0 profile with selected IoT communication protocols can safeguard IoT ecosystems and devices, further fortifying security measures. (Moraes, 2023)

In conclusion, the exploration of MQTT, CoAP, and XMPP has revealed distinct security features and challenges inherent to each protocol within the IoT realm. While MQTT excels with its lightweight publish/subscribe model, CoAP offers advantages in constrained environments, and XMPP provides exceptional real-time communication capabilities. Each protocol must be fortified with robust authentication, stringent access control measures, and a zero-trust network strategy to create a resilient defense against the evolving threat landscape. Physical security measures further serve to enhance the integrity of IoT devices, forming a comprehensive security posture. As IoT devices proliferate and their roles become increasingly critical, the amalgamation of these security principles will be imperative for safeguarding the interconnected fabric of our digital ecosystem. It is through diligent application of these layered security approaches that we can aspire to not just mitigate but stay ahead of potential risks, ensuring a secure and reliable future for IoT deployments.

References

AWS. (n.d.). *How does pub/sub messaging work?* Retrieved from aws.amazon.com: https://aws.amazon.com/what-is/pub-sub-messaging/

Bonetto, R., Lakkundi, V., Bui, N., & Olivereau, A. (2012, June). Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples. *World of Wireless, Mobile and Multimedia Networks*. doi:10.1109/WoWMoM.2012.6263790

Gerodimos, A., Maglaras, L., Ferrag, M. A., Ayres, N., & Kantzavelou, I. (2023). IoT: Communication protocols and security threats. *Internet of Things and Cyber-Physical Systems, 3*, 1-13.

Iqbal, F., Gohar, M., Alquhayz, H., Ko, S.-J., & Choi, J.-G. (2023). Performance evaluation of AMQP over QUIC in the internet-of-thing. *Journal of King Saud University*, 1-9.

Meneghello, F., Calore, M., Zucchetto, D., Polese, M., & Zanella, A. (2019). IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices. *IEEE Internet of Things Journal, 6*(5), 8182-8201. doi:10.1109/JIOT.2019.2935189

Moraes, M. T. (2023). The Importance of IoT Security: Understanding and Addressing Core Security Issues. *EC-Council*, 4-14.

OASIS. (2019). MQTT Version 5.0. *OASIS Standard*, 1-137.

Panchiwala, S., & Shah, M. (2020). A Comprehensive Study on Critical Security Issues and Challenges of the IoT World. *Journal of Data, Information, and Management*, 257-278. doi:10.1007/s42488-020-00030-2

Saint-Andre, P. (2011, March). Extensible Messaging and Presence Protocol (XMPP): Core. *Internet Engineering Task Force*, pp. 1-211.

Shelby, Z., Hartke, K., & Bormann, C. (2014, June). The Constrained Application Protocol (CoAP). *Internet Engineering Task Force*, pp. 1-112.

XSF. (2018, November 14). *XMPP E2E Security*. Retrieved from wiki.xmpp.org: https://wiki.xmpp.org/web/XMPP_E2E_Security