SECURITY PROTOCOLS ATTRIBUTES IN M2M COMMUNICATION

Kadatskaja O., Zhuravka A., Chila Etel

V.V. Popovsky Dep.Engineering Infocommunication
HNURE, Ukraine

E-mail: olha kadatska@nure.ua

Abstract

The article is devoted to the study of M2M, Internet information services and protocols used in machine-to-machine communication.M2M services requirements with presents a real challenge as a consequence of small packet size of the M2M data which is transmitted as irregular bursts by a large number of devises. Constrained Application Protocol enables a seamless integration with most web applications through HTTP-CoAP proxies.Communication security is not directly addressed by CoAP protocol.Is proposed use HTTPS as a security protocol in M2M communication. In work describe setting of installing HTTPS service.

In M2M communications, the data integrity requirement should be satisfied since illegal alteration may cause serious consequences, especially in life-critical M2M applications. In the LTE/LTE-A networks the amount of uplink (UL) traffic is normally lower than the downlink (DL) traffic, but M2M applications may produce more traffic data in UL channels than the data over DL channels.

The evolution of the IoT requires an approach to security which is different and supports unforeseen changes, across a wide range of applications. IoT need to integrate various sensors, computers, communication equipment, which are using different communication protocols. Wireless protocols are used in three layers, which are PHY/MAC layer, network/communication layer and application layer. IoT PHY/MAC layers involve all the common wireless communication technology, such as IEEE 802.11 series, 802.15 series, HART (Highway Addressable Remote Transducer), etc. IEEE 802.15.4 standard specifies MAC/PHY part for long-range wireless personal area network (LR-WPAN). Zigbee, WirelessHART are based on IEEE802.15.4 by adding upper layers. As TCP/IP lay the foundation for the Internet, thus IoT communication network mainly employ TCP and UDP protocols.

Application layer use HTTP, CoAP EBHTTP,DLMS etc protocols. Application layer usually employ HTTP to provide web service, but HTTP has high computation complexity, low data rate and high energy consumption. The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks. CoAP is one of the latest application layer protocol developed by IETF for smart devices to connect to Internet. As many devices exist as components in vehicles and buildings with constrained resources, it leads a lot of variation in power computing, communication bandwidth etc. Thus lightweight protocol CoAP is intended to be used and considered as a replacement of HTTP for being an IoT application layer protocol. Machine-to-machine (M2M)communicationsare the ways enabling automated applications that provide connectivity among machines.

Secure communication over the Internet by HTTPS is an extension of the Hypertext Transfer Protocol (HTTP), use for secure communication over network. PowerShell as a tool of administration, audit and security of Internet services as well as HTTPS service. In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS) or, formerly, Secure Sockets Layer (SSL).The principal motivations for HTTPS are authentication of the accessed website, and protection of the privacy and integrity of the exchanged data while in transit. It protects against man-in-the-middle attacks, and the bidirectional encryption of communications between a client and server protects the communications against eavesdropping and tampering.

CoAP will complement HTTP and that CoAP will be used not only between constrained devices and between servers and devices in the constrained environment, but also between servers and devices across the Internet. An important requirement of the CoRE working group is to ensure a simple mapping between HTTP and CoAP so that the protocols can be proxied transparently. Thus proxies and/or gateways play a central role in the constrained environments architecture. These proxies have to be able to communicate between the Internet protocol stack and the constrained environments protocol stack and to translate between them as needed. The IoT PHY / MAC layers include all common wireless technologies. HTTP is used on an unrestricted network, while CoAP is used on a limited network. CoAP has reworked some of the HTTP functionality to accommodate the restrictions. CoAP is considered for use and is seen as a replacement for HTTP because it is an IoT application layer protocol.

CoAP is protocol which is designed for RESTful applications and uses HTTP semantics (and feeds into HTTP in the wider network) but with a much smaller footprint and a binary rather than a text-based exchange. CoAP is designed to be used over UDP and is designed not constrained in a local network but provide the fundamental basis of the web.

UDP-based protocol with its own set of vulnerabilities such as spoofing IP addresses and packets, both of which are highly effective for launching large-scale DDoS attacks.

When an attacker sends a small packet to the target device and receives a much larger packet in response, the DDoS attack packets are increased.

In addition, attackers can easily destroy a target using IP spoofing by replacing the "sender's IP address" with their target's IP address, which in turn results in large data packets being bombarded.

Although the developers have introduced security measures to prevent these problems, this has made the protocol "heavier", which has significantly weakened the advantage of CoAP to be a lightweight protocol. Because of this, most CoAP deployments operate in NoSec security mode, making them highly vulnerable to DDoS attacks.

There are two popular application level protocols in M2M communication: the Constrained Application Protocol (CoAP) and the Message Queue Telemetry Transport (MQTT). CoAP is a one-to-one protocol, whereas MQTT supports a one-to-many architecture. The two protocols are inspired by different communication paradigms. MQTT is a pub/sub protocol, whilst CoAP is a RESTful protocol with built-in support of a pub/sub mechanisms through the Observe option. With this option, a CoAP client registers its interest on a resource by issuing a GET request to the CoAP server. Whenever the state of a resource changes, or at regular time intervals, the server notifies the client [1]. This twofold nature of CoAP makes it a more flexible solution for application developers. Being it RESTful and using a subset of the HTTP verbs to manipulate resources, CoAP enables a seamless integration with most web applications through HTTP-CoAP proxies. In addition, the Observe feature allows the development of efficient applications. On the other hand, this comes at the price of a more complex integration with the web applications, since HTTP does not support the pub/sub model.

Transport protocols and communication security-MQTT relies on TCP, whereas CoAP relies on UDP. From the developer point of view it follows that MQTT and CoAP inherit
from TCP and UDP their advantages and disadvantages in terms of functionality and performance. Hence, we can expect MQTT to be more reliable than CoAP, at the price of a higher overhead. Communication security is not directly addressed by these protocols. However, encryption through the network can be achieved by using Secure Sockets Layer (SSL) for MQTT, or Datagram Transport Layer Security (DTLS) if CoAP is used. The most recent version of the MQTT specifications include a basic user security, which allows sending user name and password with a CONNECT packet.

All IoT support web sites must fully encrypt the user session, from the device to the backend services. Current best practices include HTTPS or HTTP Strict Transport Security (HSTS) by default. Devices should include mechanisms to reliably authenticate their backend services and supporting applications. In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS) or, formerly, Secure Sockets Layer (SSL).The principal motivations for HTTPS are authentication of the accessed website, and protection of the privacy and integrity of the exchanged data while in transit, it protects the communications against eavesdropping and tampering [2].

The authentication aspect of HTTPS requires a trusted third party to sign server-side digital certificates. HTTPS connections were usually found only on secured payment transaction services and other secured corporate information systems on the World Wide Web.The protocol becoming more prevalent. HTTPS is now used more often by web users than the original non-secure HTTP, primarily to protect page authenticity on all types of websites; secure accounts; and to keep user communications, identity, and web browsing private.

Algorithm of HTTPS setting and configuring (installing the Web Server Role via Power Shell) are proposed .Constrained Application Protocol is network-oriented protocol, using similar features to HTTP but also allows for low overhead, multicast, etc.This paper are presented some setting needed to accomplish the

M2M service requirements. Is proposed use HTTPS as a security protocol in M2M communication. In work describe setting of installing HTTPS service.

Conclusion

The paper considers the requirements for the transmission of data packets in M2M and data integrity, since illegal changes can entail serious consequences, especially in vital M2M applications.In this paper to is analyzed protocols in different layers IoT. Taking into account the advantages and disadvantages of the protocols CoAP and HTTP, it is proposed to use the HTTPS protocol for the integrity of data transmission in M2M communications. An algorithm for installing and configuring HTTPS for use in M2M is proposed. For configuration, PowerShell is used as a tool for administration, auditing and security of Internet services, as well as HTTPS. Algorithm of HTTPS setting and configuring (installing the Web Server Role via Power Shell) are proposed.

References:

1.S. Bandyopadhyay, A. Bhattacharyya, Lightweight Internet protocols for web enablement of sensors using constrained gateway devices, International Conference on Computing, Networking and Communications (ICNC), 2015.

2. K. Hartke, Observing Resources in CoAP, Internet-Draft, draft-ietf-coreobserve-07, 2017.