July 2023

PROVIDING AN AUTOMATED AUTHENTICATION TOKEN ALLOCATOR FOR PROXY SERVERS

Lakshmi Shashanka

Kumud Nawani

PROVIDING AN AUTOMATED AUTHENTICATION TOKEN ALLOCATOR FOR PROXY SERVERS

AUTHORS:
Lakshmi Shashanka
Kumud Nawani

ABSTRACT

Techniques are presented herein that support an automated authentication token allocator service running on any virtual machine (VM) or server, in the same network as any requesting entities, that can allocate the tokens that are to be used by an entity (such as a proxy server, devices, other services, etc.) that wishes to send a registration request to a management application (such as a network management application) so that it can be authenticated and then managed.

DETAILED DESCRIPTION

As an initial matter, it will be helpful to confirm the meaning of several of the terms that appear in the narrative that follows. For example, a switch may be characterized as an industrial protocol-supported switch. Such a switch may only include Ethernet ports, but may not support cellular-based access, and may not have direct access to the Internet. Further, an edge proxy server may be characterized as a virtual machine (VM) that runs on the edge of a manufacturing plant floor. A proxy server may forward or proxy traffic from a switch to the cloud.

Additionally, a network management application may be characterized as a management application that runs in a cloud. The network management application may provide for the ability to manage various switches, routers, proxies, and gateways for a deployment.

Figure 1, below, presents aspects of an exemplary access-restricted industrial network that illustrates one possible arrangement of the above-described components.

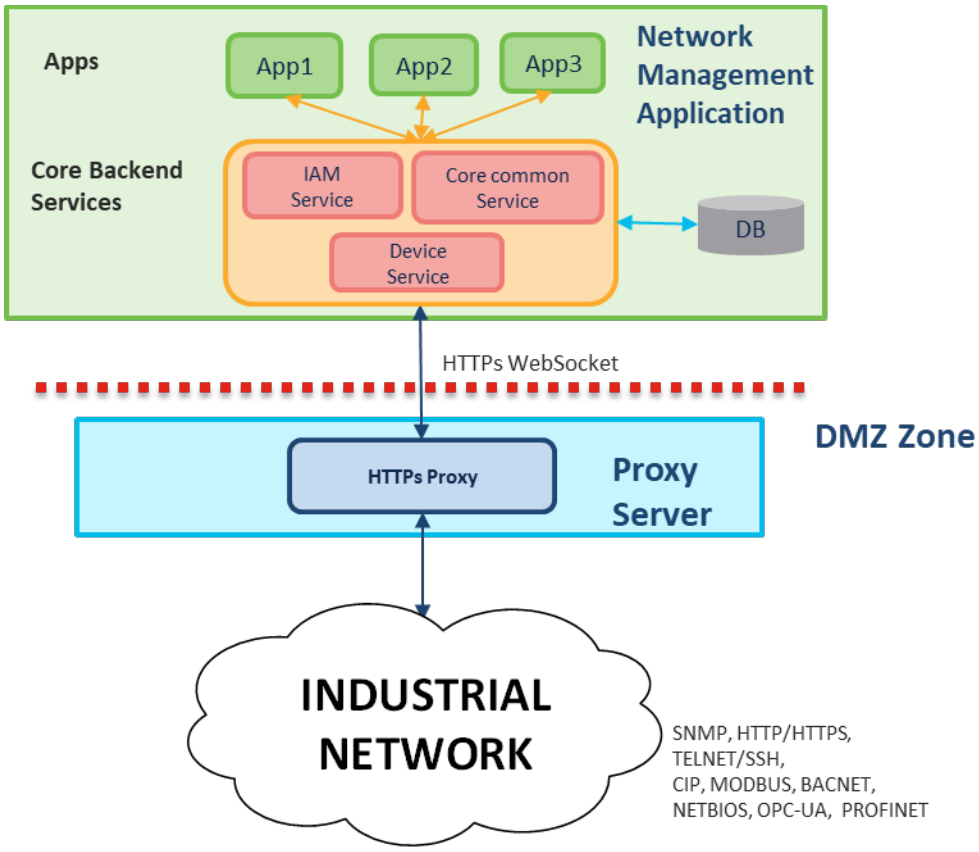

Figure 1: Access-Restricted Industrial Network with a Proxy Server

In the arrangement that is depicted in Figure 1, above, the elements that are shown as residing within the industrial network may employ any number of communication protocols or technologies including, for example, the Simple Network Management Protocol (SNMP), the Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS), Telnet or the Secure Shell Protocol (SSH), the Common Industrial Protocol (CIP), the Modbus protocol, the BACnet protocol, NetBIOS communication services, the OPC Foundation‘s Unified Architecture (OPC UA), and elements of the Profinet solution.

As illustrated in Figure 1, above, switches and routers that are deployed on a manufacturing floor are generally not accessible to entities that reside outside of such an industrial network. To manage such a restricted industrial network, a proxy sever may be deployed on a manufacturing floor to facilitate management by a network management application. Under such an arrangement, a mechanism is needed to allow the edge proxy servers (i.e., the proxy servers) to be authenticated and registered on the network

management application so that the proxy servers in the industrial network may be managed.

Critically, any solution of the above-described need must ensure that the network management application can authenticate that a sent request is from a genuine proxy server and not, for example, the result of a man-in-the-middle attack.

A number of existing technologies could be applied to try and meet the above-described need. For example, under a first approach, a certificate could be installed in a proxy server and then added as a trusted certificate in the network management application. However, since this approach would require the network management application become a certificate authority (CA) that can create certificates for a proxy server, this solution is not acceptable.

A second potential approach encompasses incorporating a proprietary agent in a proxy server and having the network management application operate as an automation server so that device registration may take place using an automation protocol. However, as a proxy is intended to be lightweight, adding an automation agent to a VM is inadvisable.

A third potential approach encompasses the use of tokens. Under this approach, a token could be generated by the network management application and then recorded in a configuration file on a proxy server. An agent in the proxy server could then send that token as part of each request and response to identify itself to the network management application. Although feasible, this approach is not preferred since it requires that an operational technology (OT) user manually update the configuration file in the proxy server. In general, any such manual intervention is not preferred; rather, an automated approach is desired.

Techniques are presented herein that address the above-described need by supporting an automated authentication token allocator (ATA) that may be employed with a network management application for industrial network deployment scenarios. Under the presented techniques, as will be described in detail below, an entry (comprising a unique identifier) for a proxy server may be added to the network management application after which the proxy server may be managed by the network management application.

In advance of the different activities of the presented techniques that will be described below, a number of preliminary setup steps may be completed by a system

administrator. First, such an administrator may update a token pool to allow for the use of the same by a customer organization (through, for example, an entry in a controller profile) following a purchase by the customer of network management application licenses for the customer's proxy servers. Additionally, a network management application user may add entries for those proxy servers in the network management application.

According to the techniques presented herein, an ATA may run or operate on any VM or server in the same network that an entity, such as a proxy server, resides. Such a component may allocate the tokens that are to be used by an entity (like a proxy server, devices, other services, etc.) that wishes to send a registration request to the network management application so that the entity can be managed. The ATA may consist of an automation agent that can connect to a home service and a token allocator (that uses the automation agent to access the pool of tokens that have been allocated to a customer or user by a system administrator). Figure 2, below, depicts elements of the arrangement that was described above.

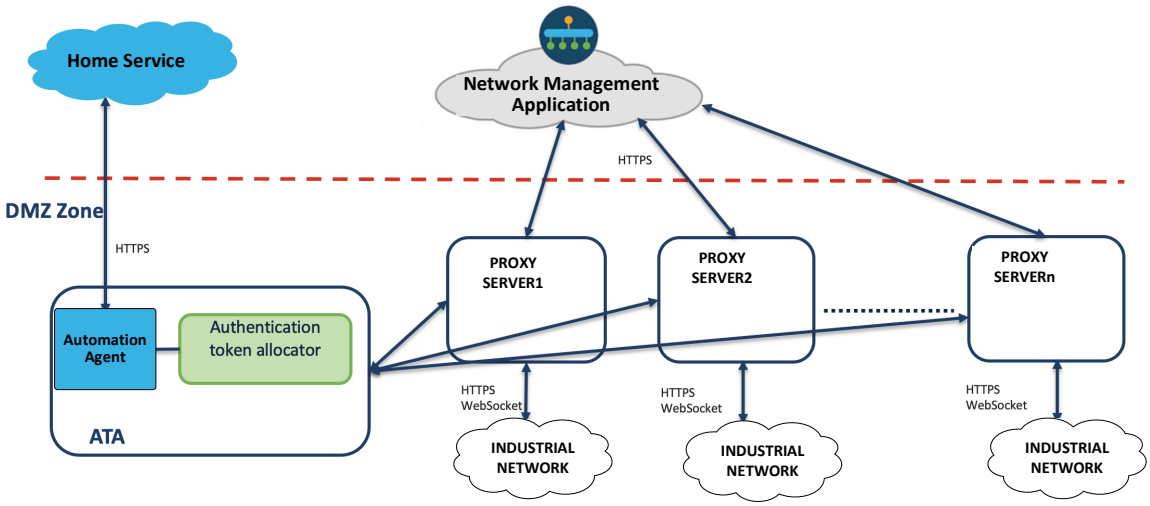

Figure 2: Authentication Token Allocator

When a proxy server boots up, it may send a GET representational state transfer (REST) application programming interface (API) request to the ATA to request a token. The ATA may communicate with a licensing account (through a home service), fetch the token pool that is allocated to the organization to which the proxy server belongs (based

on, for example, an entry in the controller profile), and then dynamically allocate one token from the pool to the requesting entity (i.e., the proxy server).

The proxy server may then send a registration request to the network management application, including the returned token, so that the network management application can authenticate the proxy server, complete the registration process, and begin managing the proxy server.

Figure 3, below, depicts elements of a proxy server registration workflow that is possible according to the techniques presented herein and which is reflective of the above discussion.

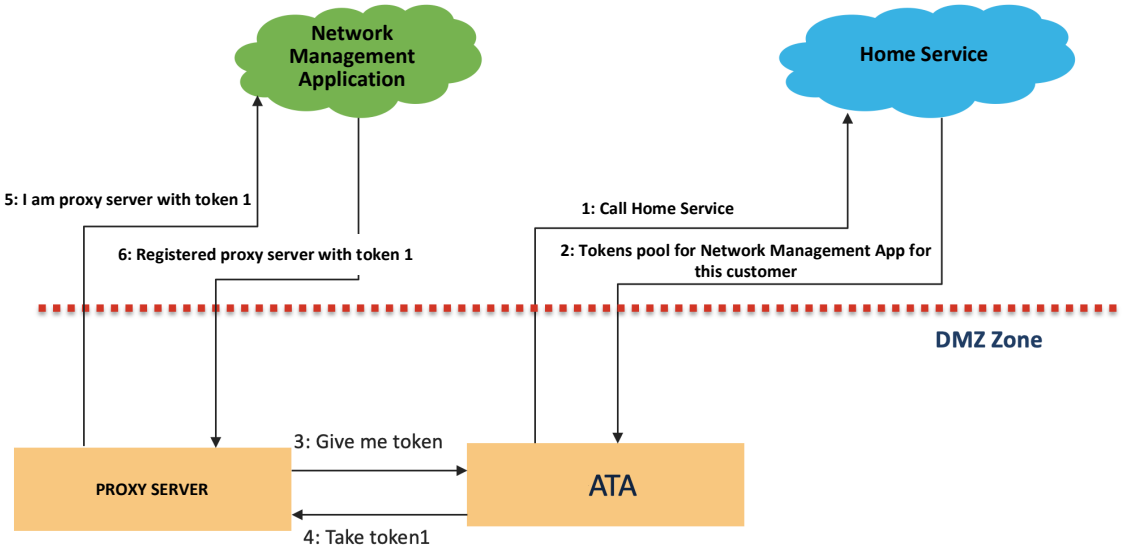

Figure 3: Proxy Server Registration Workflow

Use of the presented techniques offers a number of advantages. First, after a token is allocated an entity need not authenticate itself again with credentials and can gain access as long as the token is valid. Second, an additional agent software element or protocol stack is not needed to perform authentication within an entity. Third, the techniques offer the flexibility to use the same token pool for any entity software or hardware.

Fourth, the techniques offer a much simpler approach, compared to a complex secure unique device identifier (SUDI) certificate-based mechanism, for the allocation of a unique identifier for an entity. Fifth, the employed token-based authentication mechanism is completely automated in an industrial network, with no steps needing to be performed

by OT users. And sixth, an ATA may be installed on any Linux-based VM in an industrial network.

Additionally, the presented techniques encompass a number of security considerations. First, the communication between an ATA and a home service may be conducted using a proprietary protocol that is, among other things, secure. Second, the communication between an ATA and a proxy server is secure through the use of RESTCONF APIs over HTTPS. Third, while a token will remain valid for a lifetime, an explicit expiration date/time may be set for extra security where short-lived tokens are needed for limited access.

In summary, techniques have been presented herein that support an automated authentication token allocator service running on any VM or server, in the same network as any requesting entities, that can allocate the tokens that are to be used by an entity (such as a proxy server, devices, other services, etc.) that wishes to send a registration request to a network management application so that it can be authenticated and then managed.