Web-based Secure Remote Access to IoT Edge Devices

Secure remote access to web server of an edge device is one of the fundamental building blocks of the Internet of Things. macchina.io Remote Manager enables easy and secure remote access, even if the device is located behind a NAT router or a firewall and does not have a public IP address.

Web-based user interfaces are state-of-the-art in network-based embedded systems. These web-based user interfaces makes configuration, control and monitoring of a device from every PC, smart phone or tablet device running a web browser possible. Thanks to advanced web browsers, JavaScript and Ajax technologies, modern web-based user interfaces are powerful, visually attractive and easy to use. Since their only requirement is a HTTP(S) connection between the web browser and the web server running on the device, they are perfectly fitted for remote access scenarios. However, for this to work, the web browser must be able to create a network connection to the device's web server. This is only possible if the embedded device is located in the same network as the device running the web browser, if the networks containing the client and server are linked, or if the embedded device can be directly reached over the internet. Unfortunately, this is rarely the case in practice. IoT edge devices in the field are often connected to private networks behind NAT routers or firewalls. This is especially true for industrial IoT devices, which are typically located behind a NAT broadband router. Even devices connected to a mobile network such as 3G or LTE in most cases do not have public IP addresses and thus are not directly reachable. This means that while these devices can open connections to servers on the internet, it is not possible to access the device's web server from the outside, unless additional measures are taken.

WEB-BASED REMOTE ACCESS TO IOT EDGE DEVICES WITH REMOTE MANAGER

Port forwarding and Virtual Private Network (VPN) are well-known and established technologies for enabling internet-based remote access to computers and network devices behind NAT routers or firewalls. However, as detailed in the table at the end of this white paper, both technologies have severe drawbacks when being used with embedded systems. For this reason, Applied Informatics has created a new technology that is a great alternative to port forwarding and VPN.

macchina.io Remote Manager enables easy and secure remote access to the web server and other TCP-based services such as secure shell (SSH) or remote desktop (VNC) of a device, even if the device is located in a private or mobile network behind a NAT router or firewall. How this technology works will be explained in the following.

Application Scenarios

- Remote access to IoT gateways, edge computing devices, data loggers, metering and monitoring devices, e.g. in renewable energy, environmental monitoring, traffic, transportation and infrastructure, etc.
- Remote access to mobile devices for data acquisition, tracking, fleet management, etc.
- Remote support, maintenance and servicing of consumer electronics, home/building automation, HVAC devices, industrial equipment, etc.
- Remote access to IP network cameras and DVRs
- Remote access to security and access control systems

HOW REMOTE MANAGER WORKS

macchina.io Remote Manager is based on an extension of the well-known and proven HTTP protocol that drives the internet. The main difference to HTTP is who is setting up the network connection which is used for sending HTTP requests and receiving their responses. In HTTP, the client (web browser) is responsible for opening a connection to the device web server, over which it then sends the requests. In Remote Manager, it's the device that sets up the connection. Since the device does not know its clients, and would not even be able to create a direct network connection to each client (as clients are usually behind a NAT router or firewall), the device opens a connection to a special server called the reflector server. For this to work, the reflector server must be accessible over the internet. Once the connection between the device and the reflector server has been established, the reflector uses this connection to send ("tunnel") HTTP requests and other TCP-based protocols to the device. Where do this HTTP requests come from? The reflector also contains a normal HTTP server, which accepts requests from clients (web browsers). These requests are then simply forwarded to the device, using the device's tunnel

connection. Setting up the initial tunnel connection between the device and the reflector server is almost always possible as long as the device can access the internet. It even works through a HTTP proxy server. The tunnel connection uses the standard WebSocket protocol, which makes it firewall and proxy friendly.

REMOTE MANAGER IN PRACTICE

In a typical usage scenario, more than one device will be connected to a reflector server. Therefore, when the reflector receives a HTTP request from a client, it needs to find out to which device the request must be forwarded. There are two ways to make this work. The first one is via the URL sent from the client to the reflector (e.g., *http://dev1.my-devices.net*). This requires setting up a wildcard DNS record in the DNS server which resolves all requests for **.my-devices.net* to the reflector server *www.my-devices.net*. The reflector server can then use the Host header in the HTTP request together with an internal table to associate the request with a device. Alternatively, the reflector server could set a cookie in the client after logging in to the reflector server and selecting a target device. This cookie is then sent with every request from the client to the reflector and allows the reflector server to forward the request to the appropriate device. There are multiple options for running the reflector server. It can be deployed on a internet-facing server in a private datacenter (on-premises), or it can be ran on a virtual private server (VPS) provided by a cloud service provider such as Amazon (EC2), Azure, Rackspace or DigitalOcean. Running the reflector server can also be outsourced to a dedicated service provider.

SECURITY AND PRIVACY GUARANTEED

Since the reflector server only transparently forwards HTTP requests and TCP connections, but does not store any data passed through it (except for optional caching of images and style sheets in order to improve performance), macchina.io Remote Manager does not introduce any additional data security and privacy risks – especially if the reflector server is operated in a private data center. Of course, both the connection between the device and the reflector server, as well as the connection between the client (web browser) and the reflector server are encrypted with SSL or TLS. A single reflector server instance can easily handle many thousands of devices, with up to 200 or more simultaneous user sessions. A great advantage of this technology is that it is inherently secure. Since the device does not need to have any open ports to the internet, there is no danger of denial-of-service or other attacks against the device. Requests to the device can only be sent through the reflector server, and the reflector server requires proper authentication of the user before forwarding requests to the device. Also, devices must authenticate themselves against the reflector server when setting up the tunnel connection. Device authentication is done through a shared secret (password, or challenge-response) or certificate.

WORKS FOR WEB, VNC AND SSH

macchina.io Remote Manager is not just for accessing web pages. Virtually every TCP-based protocol can also be used over a Remote Manager tunnel connection, including web services based on REST, JSON-RPC or SOAP technologies, or even the SSH and VNC protocols. Remote Manager even includes a web-based VNC client. This makes it a great foundation for automated device management applications and remote support/maintenance portals.

EASY INTEGRATION AND CUSTOMIZATION

The software necessary for integrating Remote Manager into a device, as well as the reflector server is provided by Applied Informatics. For devices where the necessary modification of the firmware is not possible, a special low-cost gateway device can be used to connect the device to a reflector server. The gateway is located in the same local area network as the device, and forwards requests from the reflector server to the device's web server.

The reflector server can be integrated with customer applications via its REST API. The default web user interface of the reflector server can be customized to match customer-specific needs and visual style. The reflector server optionally supports LDAP for user authentication.

macchina.io Remote Manager is a great alternative to technologies like NAT port forwarding and virtual private networks to enable easy and secure remote access to the built-in web server of a device. The technology can be used without touching the existing network infrastructure. The necessary reflector server can be operated in "the cloud", and devices can be easily integrated, either by updating the firmware or by using a special gateway device.

GET STARTED WITH A FREE ACCOUNT

macchina.io Remote Manager can be used for free with up to five devices. For more information as well as tips for getting started, please visit the website at https://macchina.io.

Technology	Advantages	Disadvantages
macchina.io Remote Manager	> based on proven and proxy/firewall-friendly WebSocket protocol > can be used without changes to the existing network infrastructure > supports secure, encrypted (SSL/TLS) and authenticated connections > secure forwarding of most TCP-based protocols, not just HTTP, including SSH for remote shell and VNC for remote desktop access > the reflector server can be operated in the cloud > high scalability, up to ten thousands of devices per reflector server instance (multiple reflector servers can be clustered to increase capacity)	> macchina.io Remote Manager agent software must be integrated into device, or a gateway device must be used to integrate legacy devices > some TCP-based protocols cannot be forwarded (e.g., FTP)
Port Forwarding	> simple and widely supported by NAT routers > allows access to any TCP or UDP-based network service provided by the device	> NAT router configuration for port forwarding can be complex, especially if multiple devices must be accessible (every device needs a unique public port number) > a Dynamic DNS service is needed if the NAT router does not have a static public IP address > the device is directly exposed to the internet – very high risk and danger of denial-of-service or other attacks
Virtual Private Network	> the device is directly integrated into a remote network using a secure tunnel through the internet > secure, encrypted connection > proven, standardized and widely available technology	> VPNs may be blocked by network provider or legally restricted > necessary network and VPN server infrastructure is difficult to setup and to maintain, especially if lots of devices must be integrated > all clients must have access to VPN in order to access the devices – not suitable for end-user access > additional measures must be taken to isolate devices in the VPN from one another and to prevent users from accessing devices they should not access

CONTACT US FOR MORE INFORMATION

Applied Informatics Software Engineering GmbH
Maria Elend 143
9182 St. Jakob im Rosental
Austria

+43 4253 32596

info@macchina.io | https://macchina.io

macchina.io
empowering things