EdgeView FAQ

1) What is EdgeView

EdgeView is a tool to allow users to interact with the remote edge devices and applications. EdgeView is implemented as a Docker container. The EdgeView container on the remote device serves as a 'server' function for EdgeView, and the same container on the user laptop serves as a 'client' function. The EdgeView client and server hops through the Dispatcher to communicate to each other. For a more detail description of the EdgeView, see the EdgeView Architecture document. EVE has EdgeView support since release 8.5.0.

2) Where to get EdgeView

EdgeView is built as a Docker container, it can be pulled from docker registry with 'lfedge/eve-edgeview'. The source code is at EVE repository in pkg /edgeviw.

3) How do I start EdgeView

For the EVE implementation, the device needs to receive the configuration from the controller about the EdgeView. The API is described in this EdgeView Design page.

For the UI implementation of ZedControl, the user on the device page, clicks the button 'Start EdgeView' to enable the EdgeView on the device. After it is started, the 'Download Script' button can be used to download the EdgeView Client Script (the user may need to change the file to be executable. e.g. run 'chmod u+x <script file>').

For the ZCli implementation, by issuing 'zcli edge-node start-edgeview <device-name>' to enable the EdgeView on the device. An Edgeview Client Script will be downloaded onto the user's laptop.

4) What is EdgeView Security Mechanism

First of all, to enable EdgeView on an EVE device to allow users remote access into it, the session needs to be allowed and enabled on the controller side. EdgeView configuration is part of the EVE device configuration. The configuration also defines access policies for this particular session. See EdgeView Policies for details.

A JWT token is generated when the EdgeView session is enabled for the EVE device. The token is signed by the controller and verified by the EVE device when it receives the EdgeView configuration from the controller. The token has an expiration time which is defined by the controller for this session. When the token expires, the EdgeView session, which connects to the dispatcher, will be torn down.

The remote user needs to acquire the same JWT token in order to establish an EdgeView session into the device or applications for troubleshooting or management.

Both the device and the user's laptop connect to the dispatcher, defined in the JWT token, through HTTPs session with TLS encryption. All the messages inside the EdgeView session is either authenticated or encrypted bidirectionally with a random 'nonce' which is created when the JWT token is generated by the controller. Thus even if the dispatcher server is compromised, the EdgeView messages can not be modified or viewed.

5) Why not just use SSH

SSH works fine if the user laptop and the edge device are in the same network, either they are all on the Internet or all in a private VPN network. If the edge device is behind NAT, firewall, LTE or proxy server, and the user's laptop is not, then SSH will not work. Also in the case of the user's laptop and the device belonging to the same network, if multiple users want to access the device, they all need to share the private SSH key (or add multiple public keys onto the device) which sometimes is not desirable.

6) Why not just use SD-WAN

First, yes, when an edge device is behind the firewall, NAT or private LTE router, SD-WAN can be used to access that. The EVE device can be part of the SD-WAN just like any host or servers inside a company's VPN. This is in the IT domain of an enterprise. This assumes the enterprise already has the SD-WAN network and also the IT department allows the edge devices to be part of the VPN in the company.

There are many different SD-WAN solutions, different enterprises use different solutions and they have different IT policies and rules on the SD-WAN network. How to use the SD-WAN software to access the edge device and applications for troubleshooting will have to be achieved in case by case manner. The user can create a virtualized instance of the SD-WAN client as an App on the EVE device, the user's laptop has also to be part of the VPN. The correct routing needs to be set up in the SD-WAN App, the user can then access the other applications on the EVE device or the network connected to the EVE device.

Then, another solution can be to use the SD-WAN for EVE devices by the EVE controller provider independent of enterprises. The EVE controller provider manages the SD-WAN controllers and systems. The SD-WAN client runs as part of the EVE software. First of all, this needs to get enterprises IT permission to have a non-native SD-WAN into their remote locations; then to manage the SD-WAN controller itself, and make them scalable and HA is not a trivial task. There is also the challenge of security measures needed for managing multiple enterprises and synchronizing the device's SD-WAN status to the EVE controllers.

While the EdgeView solution is light weight, it does not need a controller for the operation. The user on the EVE device controller needs to authorize and start the session, the rest of the operation is between the user and the EVE device sharing a private token which only has a limited time to live. The EdgeView does not have all the capabilities of a normal SD-WAN, it has a set of commands to be used for EVE device troubleshooting, and it allows TCP access for applications and other servers on the remote network. Users do not need to configure and run routing protocols for EdgeView which normally is required by the SD-WAN clients.

7) Why not just use WireGuard or OpenVPN

The WireGuard and OpenVPN allows clients to communicate through their servers which reside in the cloud side. Normally all the endpoints share the same IP subnet in the VPN. All the endpoints of this VPN can talk to each other (if the server does not set limitations). The procedure to setup something for a user laptop to access the edge-node and its applications is like this:

- run wireguard, and generate private/public keys on device behind firewall
- run wireguard, and generate private/public keys on the user's laptop
- add entry on the wireguard server configuration file for the peers of the new device and the laptop, which include the public keys, internal VPN IP addresses, etc.
- setup routing on the device, the laptop and also on the wireguard server (if the traffic endpoints is not part of the VPN subnet)
- may need to open up a new firewall rule for this wireguard server (UDP packets) on the device side
- make sure the security, address allocation and routing among multiple sessions and multiple enterprises do not have issues
- make sure the server redundancy works

If one is going to automate the above list, it actually is an implementation of a SD-WAN.

OpenVPN is similar to WireGuard in terms of the scheme of client/server, different cryptographic mechanisms are used. Similar steps as above are needed.

Some firewalls drop outbound UDP packets unless they are DNS type (Wireguard is UDP), to utilize the edge node controller HTTPs endpoint is important.

While for EdgeView, on the device controller, the user just needs to click one button to start the EdgeView on the device, it also creates a EdgeView client script to be ready to run on the user's laptop. There is no need to program another server configuration for the IP addresses and public keys. There is no routing that needs to be set up, and also there are no VPN IP address allocation issues. Yes, EdgeView does not have the N-to-N communication capability as in a normal SD-WAN or a VPN, but it allows multiple users to access the device which is behind a firewall or a proxy server to do the troubleshooting of the device and the management of applications associated with the device in a secure way.

8) Does EdgeView use IP overlay

No. Unlike a normal VPN going through multiple domains (with Internet in the middle) using routing schemes, EdgeView has multiple intermediate nodes stitching the traffic bidirectionally. It does not need to use IP over IP scheme. The EdgeView message is carried in normal TCP packets without IP overlay.

9) Can the EdgeView Client and Device Communication bypass Dispatcher

No. The EdgeView channel between the client on the user's laptop and the remote device has to go through the Dispatcher. It currently does not support a dynamic shortcut for direct connection. The use cases of EdgeView is not a general full-mesh VPN solution which may try to optimize the latency.

10) If controller has 'Remote Console' for EVE App, is that equivalent to EdgeView

Yes or no. EdgeView TCP channel does offer the capability of allowing the users to use VNC client to connect to the EVE application's console port, but it also offers other access methods such as SSH, and it allows users to get to other TCP services provided by the applications. EdgeView allows the users to do debugging and troubleshooting on the targeted EVE device.

In the EdgeView scheme, the controller's role is to set up and start the EdgeView session; the controller does not get involved in the packet/data message switching part.

The decoupling of controllers from operation of EdgeView offers several benefits. First it allows simplification of controller's workflow; when debugging EdgeView operation itself, the only item needs to be checked is the EdgeView container which runs on the device side and on the user's laptop; when adding a new feature into EdgeView, normally only the EdgeView code needs to be touched, thus it is easy and fast to add features into EdgeView.

11) Why Dispatcher is needed, who controls it

If making an analogy between EdgeView and SD-WAN, EdgeView is a Hub and Spoke topology with the Dispatcher as the 'Hub' and the user's laptop and the EVE device are two 'Spokes'. This is true for any VPN with different remote sites having to cross the domains or Internet. The Dispatcher will connect to two different sites for the same EdgeView session to allow the user to access the EVE device and applications. The Dispatcher will stitch the messages from one side to the other based on a predetermined hash value generated by the EVE device controller.

Dispatcher can be placed anywhere, in the public cloud or private data center, as long as it can be reached from both ends of the EdgeView containers (the EVE device and the user laptop). It can be controlled by the same cloud management of the EVE device's controller or by the enterprises themselves. Since all the EdgeView messages through the Dispatcher are either authenticated or encrypted, it can not insert messages into the session or read from the session.

12) What is EdgeView TCP channel

From the user enabling EdgeView TCP channel point of view, it is by issuing the command 'tcp/<ip-address:port>/...', it will launch the TCP channel. See detail on TCP command.

The EdgeView TCP channel bridges the user's laptop TCP service to the remote device TCP service across different routing domains. The EdgeView on the user's laptop will set up TCP servers (listening on the pre-allocated TCP ports, 9001 and above). When the user launches some TCP client program, for example, 'ssh' client or web browser to point to the EdgeView local TCP service endpoint, this will be just like the user launching those client services on the remote device.

13) Does EdgeView work for devices behind NAT or Firewall

Yes. For firewalls, make sure the dispatcher IP address and port number is not blocked by the firewall rules.

14) Can Edgeview work if the device is behind a proxy server

Yes and no. EdgeView uses WebSocket (The Websocket Protocol) for bidirectional communication between the client and server. The HTTP protocol needs to be upgraded between the websocket client and server. If the proxy server is a 'pass-through' type for the HTTPs traffic from the device, in other words if the proxy server does not intercept the TLS, then the EdgeView will work through the proxy server. But if the proxy server is a 'MiTM' type or 'SSL-Bump' type, the proxy server needs to make a separate HTTPs connection to the DIspatcher and it may not request the 'Upgrade' service towards the Dispatcher, then the EdgeView will break since it can not establish the connection to the Dispatcher. This is mainly a proxy server software implementation issue. From an operational point of view, the proxy server can be configured by making exceptions for the WebSocket packets while keeping the 'MiTM' operation for the other HTTPs packets.

15) Can I SSH into the remote EVE device over EdgeView

Yes if the controller policy allows it. EVE software has the 'ConfigItem' configuration for installing user's SSH public key, the 'sshd' currently listens on '0.0.0.0:22'; but later on it can be changed to listen only on '127.0.0.1:22' and dynamically sets up a non-root user to be more secure. Assume the user's laptop has the SSH private key, the user sets up the EdgeView command 'tcp/localhost:22' in one terminal, and opens another terminal to enter the SSH session by issuing "ssh -i <my-ssh-private-key> root@localhost -p 9001".

16) How to use WireShark on remote EVE device interfaces

You can run WireShark application on your laptop and capture packets through it to remote EVE devices. Need to run later versions of WireShark, version 3+.

First SSH needs to be enabled on your remote EVE device (upload through the EVE configure items with your SSH public key).

Launch the EdgeView with command 'tcp/localhost:22', which normally maps the channel into port 9001 locally.

In WireShark user interface (taking example of macOS), in the 'Capture' section, there is a pull-down menu, to select 'External Capture'. The page should have a list underneath with a 'setup' icon and 'SSH remote capture: sshdump' line. Click on this 'setup' icon at the left of the line. The 'Wireshark - Interface Options: SSH remote capture: sshdump' window is popped up. Configure three items:

- Server: Remote SSH server address: 127.0.0.1; Remote SSH server port: 9001
- Authentication: Remote SSH server username: root; Path to SSH private Key: browse and select your private key file, or do nothing if the private key is your laptop's '~/.ssh/rsa_id' file.
- Capture: Remote interface: type in the name of the remote EVE device interface name, e.g. 'eth0', 'bn1' or 'nbu3x2'.

You can hit the 'Capture' button on the WireShark to start collect packets, optionally define the filters you need.

17) How to log into a remote application

Before the user tries to log into the application, some application related information needs to be gathered, for instance the VNC port number, application IP address and service port numbers. Edgeview 'tcp' command can be entered for different cases. For VNC, it will be 'tcp/localhost:<590x>' which the 'x' is the VNC display number for the application. Then launch a VNC client application on the laptop with "localhost:9001' as the VNC server endpoint; for SSH (assume the application has the SSH daemon running), the command will be 'tcp/<application-intf-ip>:22'. Then open another terminal window, and issue e. g. "ssh username@localhost -i 9001". In both examples, we assume the local port for TCP is 9001.

18) Can I use VNC or RDP

Yes. For using VNC to application console, see above section 'How to log into a remote application'. For RDP, enable the RDP service on the window application and find the application's interface IP address, using the EdgeView command 'tcp/<app-intf-ip>:3389' in one terminal, then launch the window RDP client to "localhost:9001" to connect.

19) Can web browser be used over EdgeView

Yes, the EdgeView TCP channel can be used to bridge the browser application on the user's laptop and the remote applications on the device side. For instance, an application with an interface IP address of 10.1.0.135 has a service on port 8080, EdgeView command can specify: 'tcp/10.1.0.135:8080' to launch the channel, and use the browser to point to url 'http://localhost:9001/<path-to-service>'.

20) Will EdgeView work for HTTPs or TLS services with remote applications

The normal EdgeView TCP relay will have problems supporting HTTPs or TLS protocols, since the source and destination IP addresses are changed and the Certificate content will not find a match. EdgeView supports the special TCP channel method using the proxy mechanism which can be used to support the HTTPs or TLS. EdgeView treats the user laptop and remote EVE device as a combined 'virtual proxy server'. The client application points to the proxy IP and port to the laptop (e.g. localhost:9001) and the proxy conversion is performed at the remote EVE device, since the device has access to the remote application's routing domain. To start EdgeView proxy, run the 'tcp/proxy' command on the laptop, then another client application (for instance a web browser) points to the laptop as its proxy server just as in a normal proxy service setting. For the details of proxy operation, see Proxy Command.

21) Does remote application need to be on EVE devices for EdgeView access

TBD

22) Why log-search if device log is already uploaded to the controller

Some logs are only present on the device and not uploaded to the controller side. For example, if the application on the device has the setting of 'not send logs'. Even if the logs are sent to the controller, the users of the enterprise may not have direct access to them. EdgeView offers the users some simple queries for the log entries on the device.

23) Is application port mapping still needed

If the device applications have the need of internal connection (not on the Internet) for machine-to-machine communication, then the port mapping is still needed.

24) How to get 'Show TechSupport' while the device fails to onboard

Yes, it is possible to get a compressed 'techsupport' file while the device has not onboarded yet. For the detailed steps, see Show TechSupport before Device Onboarding.

25) Does the EdgeView Client script run on MacOS and Window

Yes. The generated EdgeView client script will run on MacOS, assuming the docker client has been installed on the MacOS. It will run also on Windows OS if the Docker Desktop for Windows and WSL 2 is installed (e.g. with Ubuntu distro).

If the user laptop only runs WSL 1, then the EdgeView Client script needs to be simply converted into Window style script.

26) Is EdgeView Container built into EVE image

Yes, in current EVE OS releases. In future, EVE OS may decide to decouple the EVE image and some of the containers. The EdgeView container can then be dynamically downloaded into the EVE device when the EdgeView session is provisioned from the controller. In some cases it can have the chicken and egg situation, for example in order to troubleshoot the problem on the EVE device we need to use Edgeview, and due to those issues the EdgeView container can not be downloaded dynamically.

27) What does Multi-Instance EdgeView do

It allows multiple users to access the same remote EVE device or for different applications simultaneously. In the multi-instance case, the users share the same client script for EdgeView but supply an unique 'instance-id' when issuing the EdgeView commands.