Secomea Security FAQ Whitepaper

Contents

Revision 1.2 – 2020-09-03

1. Secomea uses web-based security, so how does this generally work?

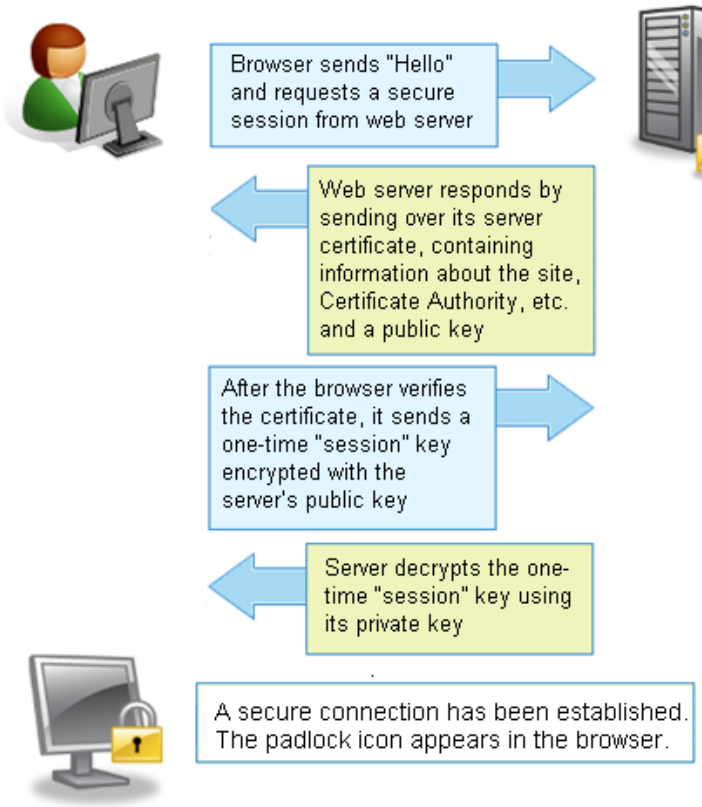
Yes, let's start with the basics. A Web client (your browser) typically does not know the specific web server beforehand, so it needs to establish **trust** in it.

You have all seen the term "HTTPS" in your browser, which is the secure version of the "Hyper Text protocol" (HTTP) used for communication with web sites. Browsers today will deem a plain HTTP server insecure, and today you will mainly see HTTP used for web server embedded in local devices, such as routers or webcams.

The "S" in HTTPS indicates use of so-called SSL (secure socket layer), or its successor TLS (Transport Layer Security). This is invented for assuring:

1. **Privacy** (encryption),
   This is typically AES
2. **Integrity** (must not be manipulated along the way).
   This is assured by an Authentication Algorithm. Typically, SHA256
3. **Identification** (assure that the server is the right one).
   This is done by an x.509 certificate

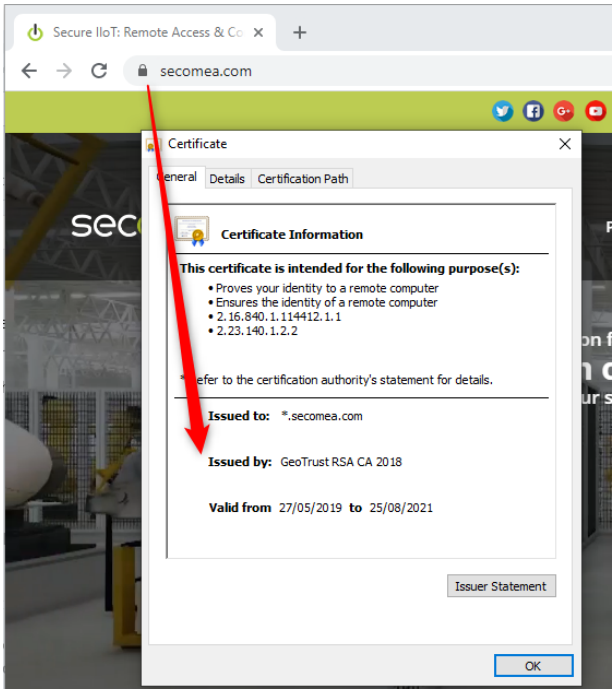
All this is obtained by a so-called **TLS handshake**. The TLS handshake works like this:



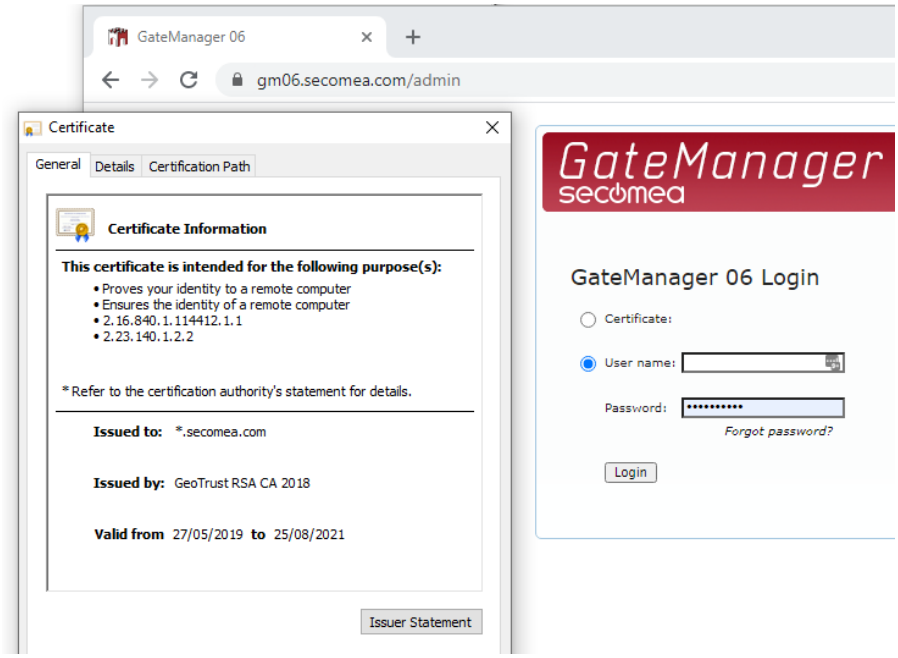
Very important here is the third step "After the browser verifies the certificate".

Modern browsers will check the certificate against the Certificate Authority (CA) that has issued the certificate for the web server, and which validates the authenticity of the server.

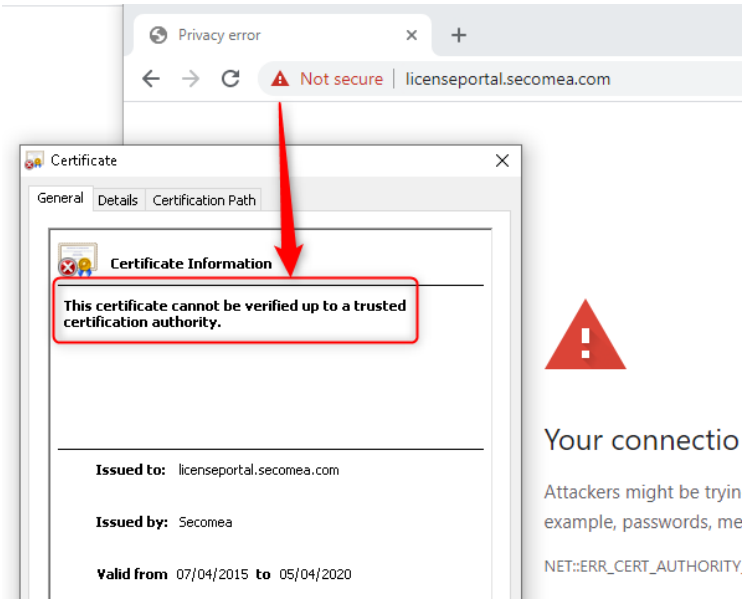
Here an example of the Secomea corporate website, where we see the certificate is issued by GeoTrust, and which the browser then validates the certificate against.



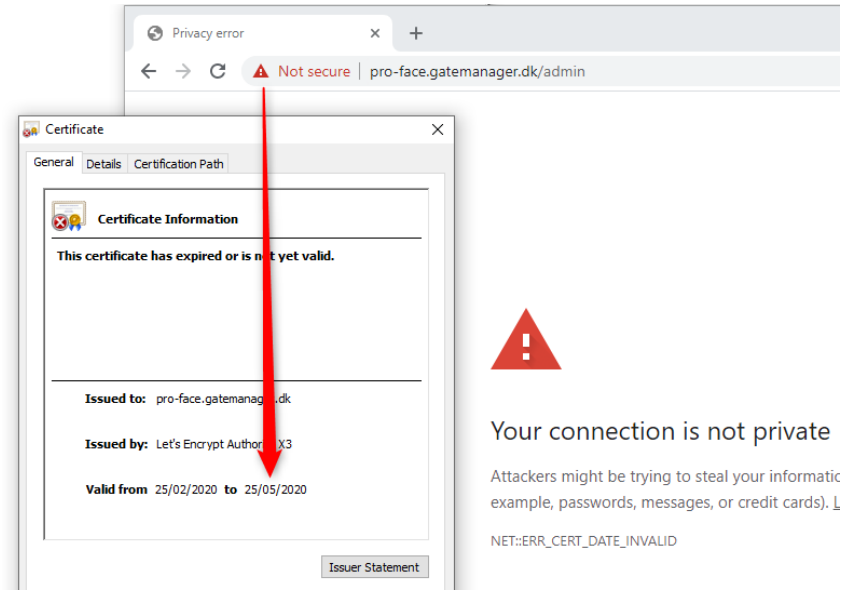
Or in this case with a GateManager



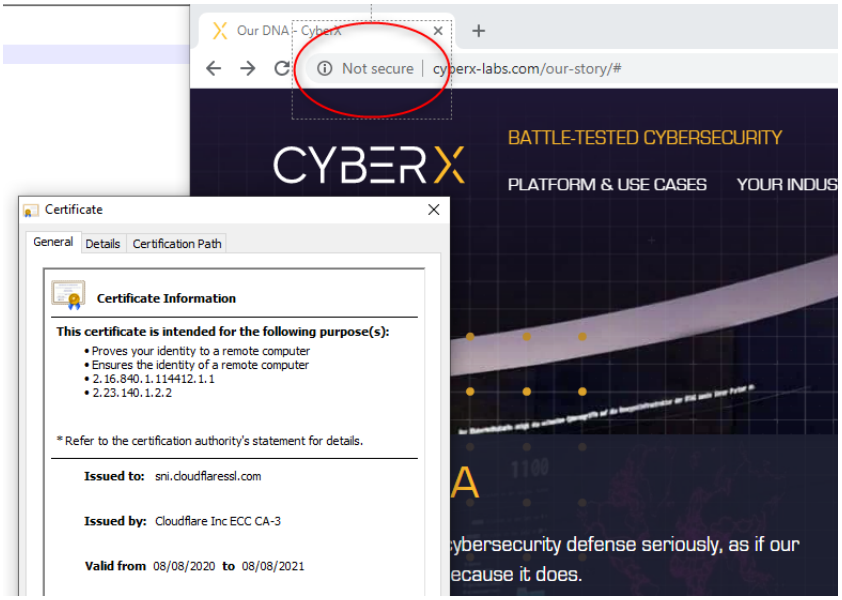
If the certificate cannot be looked up at a public CA, then the browser will deem the certificate invalid, like in this example:



There is also risk the certificate could be expired, like in this example:

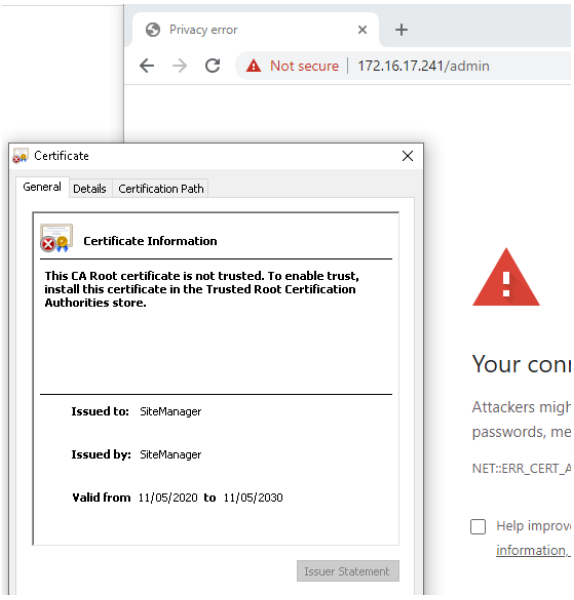


And you can even experience a server having a valid certificate, but where the server embeds elements of other sites that cannot be validated, which presumably is the case with CyberX here:



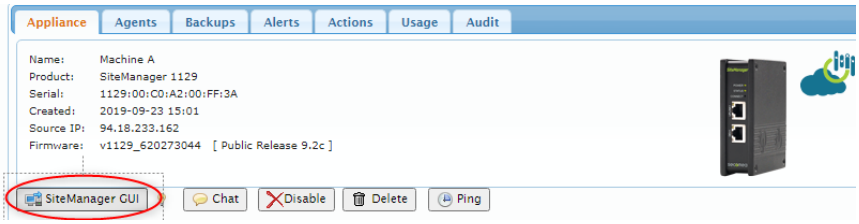
A hardware SiteManager also has an embedded HTTPS server for configuration. A SiteManager, however, does not have a DNS name associated that can be authenticated by a CA, but just has an IP address, either fixed or given by a local DHCP server.

Therefor a SiteManager will trigger the browser security warning if attempting to connect to it locally:

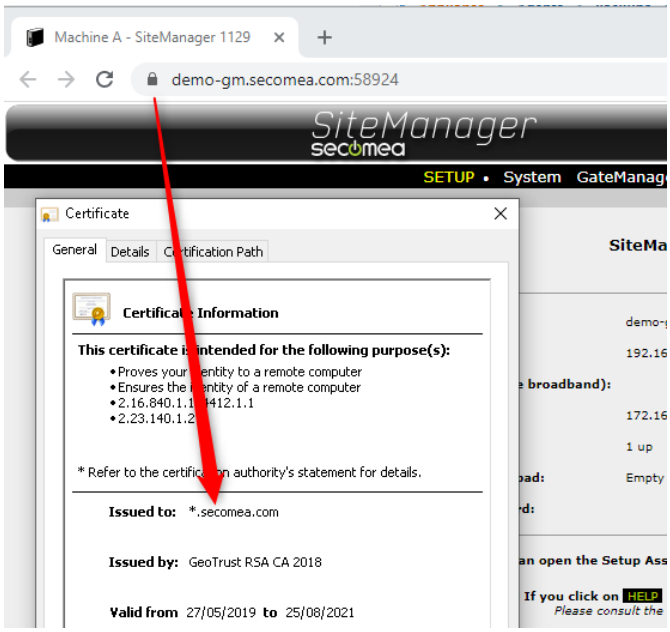


This applies to the vast majority of internal network devices that use a web browser for configuration. Many devices (including PLCs and HMIs) still just uses plain HTTP, which will not trigger the certificate warning, but exposes cleartext data on the network. Often AD applied policies even restrict users from connecting to HTTP only servers.

In most cases, however, the SiteManager is configured (e.g. via a USB memory stick) to connect to the GateManager while all further configuration is done from the GateManager via a logged in administrator by clicking the "SiteManager GUI" button.



The HTTPS session will then be managed by the GateManager that your browser already trusts



Once we have come this far, you can securely browse pages on the web server, or login to services on the web server.

2. How do LinkManager and GateManager user accounts trust the GateManager?

LinkManager and GateManager Administrator accounts work exactly like described above. The GateManager Portal is a HTTPS web server, and all client access is browser-based. As for any web server you must, as a user, assure that you are connecting to the intended GateManager by checking the DNS name and that your browser shows the Lock symbol indicating a CA validated certificate.

Once the secure TLS session is established, additional security is applied in the login process by:

1. **Authentication** - based on your login credentials.

A successful login brings you to the second step:

2. **Authorization** - based on the privileges of your account (the account role and what domains you have been granted access to by the GateManager administrator)

The LinkManager Windows adapter is just a passive component installed as a service on your machine and is fully controlled from your secure browser session with the GateManager server. A LinkManager does not have any connections to anything in idle mode. Connections are only initiated based on what your authorization grants you access to (see more details about LinkManager in a later section)

3. A SiteManager is not a client, so how can it trust the GateManager?

In fact, you can consider a SiteManager a client. It just does not have a user controlling it, so it uses a slightly different mechanism to compensate for this, although the ground principles of TLS are the same.

A SiteManager by default has trust in any GateManager at the first TLS connection. This is done by its preloaded public x.509 factory key issued by the same Secomea CA authority as the x.509 factory key on the GateManager. So, SiteManagers and GateManagers generally have trust in each other for the initial handshake.

This initial connection of a SiteManager is typically a manual portion of the SiteManager deployment, so the SiteManager should trust the GateManager that you have explicitly configured it to connect to, based on the entered GateManager IP address or DNS name.

However, the SiteManager still needs to ensure that the GateManager remains the same entity and prevent redirection of the connection to another server. Therefore, the first time the SiteManager connects with its factory certificate, it switches to using the unique TLS certificate installed on the GateManager (following its license activation based on its DNS and serial-number). This is pretty similar to the web browser scenario, although the certificate used is the activation certificate issued by the Secomea CA and not the web-certificate issued by an external CA. This is due to the SiteManager not being able, as a web browser can, to check the validity of the certificate at an external CA - it is not even allowed to connect to anything else on the Internet other than the GM. Therefor it simply binds itself to the unique activation certificate of the specific GateManager, and will not accept to connections to any GateManager without that unique certificate. Imagine your browser was locked to only connect to one single web server on the internet. To make the SiteManager connect to another GM, you must explicitly reconfigure the GateManager settings in the SiteManager, after which the certificate binding process starts over.

4. Where does the man-in-the-middle occur in all this?

The man-in-the-middle is in this browser context typically referred to as “phishing redirects”.

This was alluded to in the previous section. If an impostor takes over the identity of the original GateManager, by re-routing the connection, and simulates the GateManager IP address or DNS name, a successful connection would also require the unique TLS certificate of the server. This is nearly impossible, as the certificate is bound to the unique serial number of the GateManager that is created at installation and used for license activation of that server.

5. How does the GateManager trust the SiteManager?

The previously explained measure was primarily to protect the SiteManager from a hostile Server. So, the question is how to protect the Server from a hostile SiteManager “Client” which you cannot validate by login authentication as you can for a user with a browser.

By design, and for ease of deployment, you configure a SiteManager to connect to any GateManager publicly available on the internet and to a domain that exists on that GateManager, e.g. ROOT.AcmeInc. If that domain exists, the SiteManager will connect and attach in that domain.

Since a SiteManager is a passive component it cannot, by just connecting, send harmful data to the GateManager or its users, nor can it extract any information from the GateManager. To access data of any kind, a LinkManager must explicitly connect to a specific SiteManager or an agent on the SiteManager.

If a villain had intentionally directed a SiteManager towards your GateManager, your LinkManager users should still be authorized to connect to it, with good reason. If the villain's purpose is to attack the LinkManager Client, then the SiteManager would have to be connected locally at the villain's network to some infected server, represented by a configured agent.

GateManager administrators monitor their domains and would be suspicious of SiteManagers appearing for an unknown reason. You can also specify on the GateManager a security level which requires all SiteManagers attempting to connect be explicitly attached to a domain by the GateManager administrator.

In fact, SiteManager Embedded (SM-E) always must be bound manually to a domain as part of the license assignment process. It is doubtful that anyone would assign an activation license to an unknown SM-E.

6. When a LinkManager Windows adapter is connected, isn't it vulnerable to unintended connections on the PC?

This is a good question, and of course something that has been key to the design.

Firstly, a LinkManager connection always requires activation through a GateManager authentication and authorization process. The GateManager simply will not accept a LinkManager connection if it has not been explicitly allowed following:

- A LinkManager client login authentication;
- Authorization, and;
- Finally, the connect attempt to a SiteManager Agent.

The LinkManager user will additionally be prompted locally on the PC, by the LinkManager Windows adapter, for authorization to use this GateManager or LinkManager account with the LinkManager adapter.

Once authorized, and the LinkManager adapter has established an encrypted connection to the GateManager, the ports specified by the SiteManager Agent at the remote end are accessible by any application on the Windows user's PC.

7. What if I use natively unencrypted protocols towards the end-device?

The LinkManager adapter simply forwards all traffic "as is" between the local PC and the endpoint of the remote device, i.e. if the remote service is not encrypted, the traffic (internally to the LinkManager PC) is not encrypted either.

In this way, it is just like any other VPN tunnel service that you can operate on a PC. For example, if you had used Cisco AnyConnect to a remote network an accessed a PLC web server using the HTTP protocol, the traffic - internal to the PC - would also be (unencrypted) HTTP. Still, for both

LinkManager and AnyConnect, the traffic forwarded outside of the PC (between the PC and the remote VPN endpoint - SiteManager or AnyConnect server) is fully encrypted.

8. How to avoid Clients, SiteManagers and GateManager getting hacked on their public IP addresses?

GateManager and LinkManager user accounts, given they are based on a web browser, are always located behind firewalls, typically both a corporate one and the personal firewall on your PC.

As mentioned, the SiteManager essentially works as a web client, meaning that all traffic is outbound on port 443. SiteManagers typically are connected to a local intranet and therefore connect via the local internet firewall. When remote access is initiated, it is done via the encrypted outgoing connection established with the GateManager. Even if a SiteManager is connected directly to the internet via a 4G connection, there are no open ports on the SiteManager, and nothing that a hacker can work with (In fact, a 4G version of a SiteManager does not require any special subscription, and does NOT as some other solutions, require a SIM with a fixed IP address)

The GateManager is also located behind a suitable firewall, and like a typical web server, the only port the firewall requires to forward from the outside to the GateManager is port 443, in order for the previously described TLS handshake from GateManager and LinkManager Clients and SiteManagers to work.

The Secomea cloud based GateManagers are all behind monitored firewalls, and should you choose to migrate to a GateManager Own it would be protected by attacks by your corporate firewall.

Unlike most other remote access solutions, the Secomea solution is not relying on the internet. You could deploy the GateManager and subsequently clients and SiteManagers, in a totally closed WAN network. If granting access for remote access users from outside the WAN, you could apply a second layer of authentication (such as an RSA token) to access the WAN before the user can access the GateManager and through this be authorized to access remote devices via SiteManagers.

9. If the GateManager has access to all end devices, can't a hacker exploit these connections?

The GateManager does not hold any persistent connections to any end-device. It only holds service connections with the SiteManagers to check status and for fleet management purposes (firmware upgrade, configuration etc.).

Only when a LinkManager connection is requested by an authenticated and authorized user, will the GateManager open a proxy connection to the ports of the relevant SiteManager agents representing the service endpoints of the remote devices. Such proxy connection is end-to-end encrypted from the LinkManager to the SiteManager, so no other services or programs on the GateManager peek into or utilize this connection. The GateManager simply proxies the encrypted connection.

10. Secomea supports two factor authentication. Is this the x.509 certificate used?

Two factor authentication, typically referred to as 2FA, means a second layer of authentication and the two factors must be at least two of the following: Something you know (e.g. a password), something you have (e.g. a smartcard), or something you are (e.g. biometrics).

The Secomea solution historically used x.509 certificates for client login. Today a secure token in form of an encrypted digital file is used instead, although it is still referred to as a certificate. X.509 certificates are still used both in the TLS handshake process, and for SiteManager connections.

When using a digital certificate as second factor, it is assumed it is stored on a piece of hardware that only you are in possession of. This in combination with the password that only you should know, constitutes 2FA.

Some argue, with good reason, that since the certificate is not tied to the physical hardware (the PC), then it cannot be considered the second factor; i.e. you can load the certificate on several PCs. Secomea integrated this by design to increase flexibility, but admittedly it is not textbook 2FA. We could regard it as 1½FA, as you do have control of which PCs you copy it to. An optional feature on the GateManager allows you to lock a specific LinkManager to a specific PC, so if the certificate were copied to another PC, the GateManager would reject it despite the correct certificate.

To obtain an additional layer of authentication, you can enable SMS on the GateManager as an additional factor. In this case your phone represents the additional factor, i.e. the hardware that only you have in possession.

If you further combine this with the certificate, i.e. password+certificate+SMS, then we could argue for 2½ factor authentication.

11. There is a lot of talk about LDAP and AD. How does that fit in?

By default, the authentication and authorization process are self-contained in the GateManager. That means users must be created and administered in the GateManager portal. Many find this convenient as users may be external to the organization, and should not be managed by the corporate AD.

When talking AD (Active Directory) or LDAP (Lightweight Directory Access Protocol), we refer to managing authentication external to the application or service. That is, the login validation is done by the GateManager against a central user database that also controls access to other applications and services. The advantage is that when an employee leaves the company or should have privileges revoked or extended, it is done in the central user data base, without having to engage the GateManager.

Often the user database, in addition to user identification, also holds the policies for a specific application. This could be the user's authorization, such as their role in the application, and/or what rights the user has. In a GateManager context, LDAP authorization might be the user's classification as a LinkManager or Domain administrator, and what Domains they have access to.

The GateManager has an optional JSON API for controlling both user authentication and authorization externally. This is called CRM API and is available on private GateManagers.

In an upcoming 2020 release, the GateManager will be capable of authenticating users using LDAP (e.g. against an MS AD Server). This capability will later be extended with role and domain access authorization capabilities.

12. I've heard the term "Zero-trust". How does this apply to the Secomea solution?

Zero-trust has many definitions depending on context, but the most relevant here is:

"No security difference between Intranet and Internet"

This essentially means that the Private Network cannot be trusted, meaning that Virtual Private Networks (VPN) cannot be trusted either.

One element of zero trust is that all communication to all endpoints must be secured, and verification is required for everyone trying to get access to resources on the network. Further, you cannot rely on that the internal "perimeters", such as the Intranet, by definition are trusted.

Since the Secomea solution is designed to facilitate UDP/TCP real-time connections, it cannot by itself full fill requirements for verification to specific end-points, simply because it does not know if for example, a technician connecting from a PLC application to a physical PLC is allowed to do so. It also cannot control a potential native login process between the technician's application and the PLC. Neither can the Secomea solution ensure that the local communication between the SiteManager and the PLC is encrypted.

Which is why the Secomea solution is designed with this in mind. Some of measures that have been built in to adapt Secomea to a zero-trust context are:

1. The option to allow no external access to the SiteManager access gateway itself; i.e. it can solely be controlled by the local OT onsite, and central control by the GateManager administrator is removed.
2. Ability to deploy the GateManager as part of the internal OT/IT controlled infrastructure, and not as an external Cloud service.
3. Fine-grained access authorization; i.e. a user can access only specific end-devices, and only certain ports on the end-device.
4. Device access can be made in timeslots, or authorized by the operator locally (e.g. controlled by an input port on the SiteManager).
5. All access activity is logged for auditing, but can also be made available for real-time monitoring so suspicious activity can be intervened.
6. Ability to deploy the SiteManager as a software component directly on the end-device, and thereby eliminate potential unencrypted industrial protocols being exposed on the network.

13. What additional measures can you suggest to further elevate the security of the solution?

Different measures can be applied, but it really depends on the context and scenarios you want to address. Measures may increase control, restrict actions or access, but they may also reduce usability, so often there are trade-offs. Some examples of security elevation initiatives:

1. In your firewall: restrict the SiteManager to only access the GateManager (source/destination rule)
2. Place the SiteManager at the factory in a DMZ zone, and apply Deep Packet Inspection (DPI), on the traffic between the SiteManager and the factory devices, to monitor the type of traffic originating from LinkManager users.
3. Direct the real-time logging of the GateManager to a central syslog server such as SPLUNK and apply measures to identify cyberattacks.
4. Some of the measures are mentioned earlier, such as enforcing 2FA, requiring explicit approval of all new SiteManagers and restricting LinkManager access to specific PCs by using the GateManager feature of locking the LinkManager instance to a PC controlled by IT policies

Some of these topics are described in more detail on our online Knowledge Base; otherwise our support team is available to discuss specific concerns.

14. How do you ensure that new hacker methods do not put the solution at risk?

Any IoT solution can be subject to attacks. And any IoT solution will, at some point in time be subject to vulnerabilities; either by discovery of vulnerabilities in components or as a result of the way the solution is used. A vulnerability does not necessarily mean you will get attacks, and most vulnerabilities are discovered before hackers exploit them.

So, the trick is to fix or mitigate vulnerabilities before they become a problem to your business.

The Secomea solution is subject to regular security audits, which include penetration testing by external security experts. But even experienced security specialists cannot discover all potential vulnerabilities. To address this, Secomea has developed a Cybersecurity Advisory process, allowing anyone to report a vulnerability. Subsequently Secomea assesses the report, and if it is indeed a vulnerability, Secomea releases fixes with instructions on how to mitigate.

The findings, with fixes and mitigations, will be disclosed by CVE (Common Vulnerability Exposure) reports, so that the information is available in public CVE databases that you can subscribe to. Even vulnerabilities discovered internally in our testing and own audits will be disclosed as CVEs, so that the information is made available to the OT/IT departments of our customers.

15. You say that you are not using traditional VPN, but something called "RelayVPN"

We kept our most frequently asked question for last 😊

Some of the earlier generations of Secomea remote access also used traditional VPN, but our third-generation secure connection method introduced in 2010 is based on tunneling traffic via a proxy connection, rather than routing connections. We call it "RelayVPN" in order to highlight that it has the same end-to-end connection capabilities as traditional VPN.

RelayVPN is based on TLS connections based on x.509 (1024 bit keys) and AES256 encryption, just like SSL VPN (used by e.g. OpenVPN). So, security wise, in terms of key exchange and encryption, the Secomea solution and OpenVPN are very similar.

Both OpenVPN and RelayVPN transparently carry Layer3 traffic (UDP and TCP), and even Layer2 if appropriately implemented by network adapters on both ends (although not all OpenVPN solutions would support Layer2).

Proxying of connections has the advantage of eliminating potential issues of IP or subnet conflicts. Such challenges are typically handled by routed connections using NAT. However, handling very complex network scenarios is very difficult to accomplish with NAT and firewall rules alone. Secomea's RelayVPN eliminates most of the NAT issues one might have in many-to-many scenarios typical for industrial remote access. Additionally, Relay VPN addresses the multidimensional access need for fine-grained differentiated user access at device and port level.