Opinnäytetyö YAMK

Insinööri (ylempi AMK), kyberturvallisuus

2024

Pepe Kivekäs

Kommunikaatiosertifikaattien elinkaaren hallinta

Tutkimus S7-1500 -ohjausjärjestelmän ja WinCC-valvomon välisestä yhteydestä

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Opinnäytetyö YAMK | Tiivistelmä

Turun ammattikorkeakoulu

Insinööri (ylempi AMK), kyberturvallisuus

2024 | 76 sivua

Pepe Kivekäs

Kommunikaatiosertifikaattien elinkaaren hallinta

Tutkimus S7-1500 -ohjausjärjestelmän ja WinCC-valvomon välisestä yhteydestä

Opinnäytetyössä tutkitaan TLS-suojatun yhteyden sertifikaattien voimassaoloajan hallintaa S7-1500 -ohjausjärjestelmän ja WinCC -valvomon välillä. Työ käsittelee digitaalisten varmenteiden voimassaoloaikaa ja sen merkitystä tuotannon ja talouden näkökulmasta, kun otetaan huomioon yllättävät kommunikaatiokatkot automaatiolaitteissa ja IT/OT-ohjelmistoissa.

Opinnäytetyössä kehitettiin ohjelmisto, joka integroidaan WinCC-valvomoon ja joka ilmoittaa varmenteen päättymisajankohdan mahdollistaen ennakollisen toiminnan sertifikaattien päivityksessä. Luodussa ohjelmistossa sertifikaatti ladataan automaattisesti logiikasta Ethernet/Profinet-verkon yli ja muunnetaan datatiedostoksi sisältäen tiedon jäljellä olevien päivien määrästä ennen varmenteen vanhenemista. Samainen informaatio luetaan WinCC-valvomon tietokannan muuttujaan.

Opinnäytetyössä tuotetut ratkaisut on implementoitu yleisimpiin WinCC -valvomotuotteisiin ja opinnäytetyö sisältää mallipohjat datatiedoston lataamiseen ja informaation esittämiseen operaattorille.

Opinnäytetyössä käsitellään myös EU:n NIS2- ja CRA-direktiivejä ja niiden merkitystä kyberturvallisuudessa.

Asiasanat:

WinCC, S7-1500, TLS, sertifikaatit, digitaaliset varmenteet, TIA Portal, kyberturvallisuus, SAT SDK

Master’s Thesis | Abstract

Turku University of Applied Sciences

Cybersecurity Degree Programme

2024 | 76 number of pages

Pepe Kivekäs

Communication certificate lifecycle management

A study of the connection between the S7-1500 control system and the WinCC SCADA products

The thesis investigates the management of certificate validity periods for TLS-protected connections between the S7-1500 control system and the WinCC SCADA. The work addresses the validity period of digital certificates and its significance from the perspective of production and economics, considering the unexpected communication disruptions in automation devices and IT/OT software.

In the thesis, software application was developed, which is integrated into the WinCC and which informs the expiration date of the certificate, enabling proactive operation in the update of certificates. The created application downloads the certificate from the S7-1500 CPU over the Ethernet/Profinet network and converts it into a local datafile that contains information about the number of days remaining before the certificate expires. This information is also transferred to WinCC.

The solutions produced in the thesis have been implemented into the most common WinCC SCADA products, and the thesis includes templates for presenting the information to the operator.

The thesis also discusses the EU's NIS2 and CRA directives and their importance in cybersecurity.

Keywords:

WinCC, S7-1500, TLS, digital certificates, TIA Portal, cybersecurity, SAT SDK

Sisältö

Liitteet

Kuvat

Taulukot

Käytetyt lyhenteet tai sanasto

Lyhenne	Lyhenteen selitys
API	Application Programming Interface
ACK	Acknowledge
CA	Certificate Authority
CI/CD	Continuous Integration / Continuous Development
CRA	Cyber Resilience Act
CVE	Public Security Vulnerability Database
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
GDPR	General Data Protection Regulation
HMI	Human-Machine Interface
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISA	International Society of Automation
MAC	Medium Access Control
NAT	Network Address Translation
NIS2	Network and Information Security Directive2
NTP	Network Time Protocol
OUC	Open User Communication
PC	Personal Computer

S7-1500	Simatic Automation Controller
SAT	Simatic Automation Tool
SBOM	Software Bill of Materials
SCADA	Supervisory Control and Data Acquisition
SDK	Software Development Kit
SIEM	Security Information and Event Management
SSH	Secure Shell
SSL	Secure Sockets Layer
TIA Portal	Totally Integrated Automation Portal
TIA Portal Openness	TIA Application Programming Interface
TLS	Transport Layer Security
UDP	User Datagram Protocol
VPN	Virtual Private Network
WinCC	Windows Control Center. Simatic SCADA

1 Johdanto

Opinnäytetyössä keskitytään S7-1500 -ohjausjärjestelmän ja WinCC -valvomon välisen TLS-suojatun yhteyden sertifikaatin voimassaoloajan analysointiin. Digitaalisia varmenteita käytetään kaikissa uusissa Simatic automaatiolaitteissa sekä IT/OT-ohjelmistoissa. Digitaalisilla varmenteilla on voimassaolon määrittävä aikaikkuna, jonka jälkeen ne täytyy uudistaa. Tämän huomioimatta jättäminen johtaa usein yllätyksenä tapahtuvaan kommunikoinnin loppumiseen laitteiden välillä ja näin ollen tyypillisesti myös tuotannollisiin ja taloudellisiin tappioihin. Jos käytetään TIA Portalin integroitua toimintoa sertifikaattien konfigurointiin niin nykyinen S7 diagnostiikka kertoo kyllä, että PG/HMI-sertifikaatin voimassaoloaika on mennyt umpeen, mutta tässä vaiheessa on jo liian myöhäistä toimia preaktiivisesti (Siemens. 2023f. 187).

Perustuen asiakaspalautteista kerättyyn informaation, on havaittu, että automaattisesti tapahtuva itse allekirjoitettujen sertifikaattien päivitys on harvinaista tuotannollisissa kohteissa, mikä puolestaan johtaa siihen, että varmenteen voimassaoloajan päivitys pitää tehdä manuaalisesti ollen usein liian haastava tehtävä laitoksen omalle käyttöhenkilökunnalle.

Opinnäytetyön tavoitteena oli luoda ratkaisuna mekanismi, jonka voisi integroida yleisesti käytettyyn WinCC-valvomoon ja kyseinen mekanismi kertoisi esimerkiksi 1kk ennakkoon varmenteen loppumisajan näyttäen informaation valvomossa visuaalisesti. Tällöin tuotantolaitoksen operaattorit ehtivät ottaa yhteyttä asiantuntijaan riittävän ajoissa, jotta varmistutaan että digitaalisen varmenteen päivitys tapahtuu oikein ilman tuotantokatkoksia.

Opinnäytetyötä valmistellessa tutustuin aiheesta kirjoitettuihin opinnäytetöihin TLS-tekniikasta sekä WinCC -arkkitehtuurien ympärille tehdyistä opinnäytetöistä. Nostona mainitsen Heikki Kallankarin opinnäytetyö vuodelta 2020 otsikolla ”TLS ja palvelimien turvallisuuden arviointi” sekä

Olivia Joki-Hollannin opinnäytetyö vuodelta 2021 otsikolla ”WinCC Unifiedin ja WinCC Professionalin tuoteominaisuuksin vertailu”. Edellä mainituista opinnäytetöistä sai ideoita ja ajatuksia, joiden pohjalta miten omaa toteutusta pystyi lähteä implementoimaan.

Opinnäytetyöni jakautuu kolmeen osuuteen.

Opinnäytetyöni alussa sivutaan EU Direktiivejä: NIS2 ja CRA ja niiden merkitystä kyberturvallisuuteen.

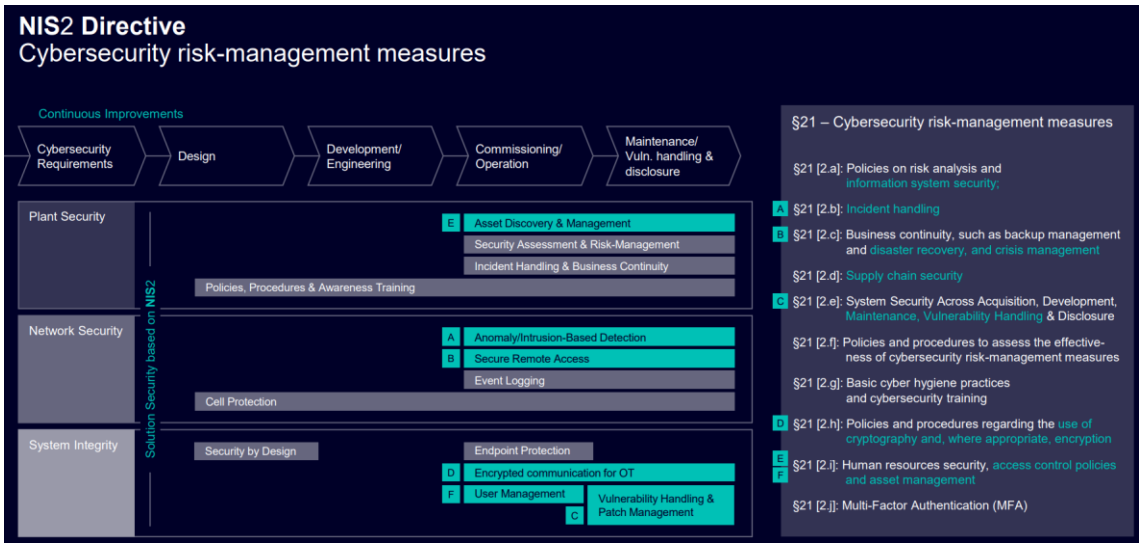
Luvussa ”Kryptografia” tutustutaan yleisesti käytettyyn TLS-salaukseen ja keskeisimpiin periaatteisiin.

Luvussa 12 kuvataan ohjelmointikirjastoja sekä uuden toteutuksen toimintaperiaate. Opinnäytetyöni loppuosuudessa analysoidaan sovellusohjelman rinnalla vaihtoehtoisia tapoja saada selville kommunikaatiovarmenteen aikaikkuna. Lopuksi käydään opinnäytetyön tulokset ja jatkokehitysideat läpi.

2 EU direktiivit NIS2 ja CRA

Euroopan unioni on ottanut käyttöön kaksi keskeistä direktiiviä, NIS2:n ja CRA:n, vastatakseen digitalisaation ja teknologisen kehityksen tuomiin kasvaviin kyberturvallisuusuhkiin. Nämä direktiivit pyrkivät parantamaan kyberturvallisuutta ja suojelemaan digitaalista infrastruktuuria Euroopassa, keskittyen erityisesti kriittisten infrastruktuurien ja automaatiojärjestelmien suojaukseen. NIS2 on suunnattu omaisuuden omistajille ja palveluntarjoajille, kun taas CRA kohdistuu tuotetoimittajiin ja OEM-valmistajiin.

NIS2 astui voimaan 16.1.2023 ja se muunnetaan kansalliseksi laiksi, joka astuu voimaan 17. lokakuuta 2024, kun taas CRA on vielä luonnosvaiheessa. Kun se hyväksytään, odotetaan 24 kuukauden siirtymäaikaa ennen sen voimaantuloa. Kuvassa 1 esitetään NIS2-vaatimukset artiklan 21 mukaisesti ja sen mukaisia ratkaisuja. Listatut ratkaisut ovat kolmessa kategoriassa: laitosturvallisuus, verkkoturvallisuus ja järjestelmien eheys (Kittler & Sulaiman. 2023. 24).



Kuva 1. NIS2-vaatimukset artiklan 21 mukaisesti ja sen mukaisia ratkaisuja (Kittler & Sulaiman. 2023. 24).

2.1 NIS2-direktiivin keskeiset uudistukset

NIS2 laajentaa aiemman NIS-direktiivin soveltamisalaa ja tiukentaa turvallisuusvaatimuksia (Kuva 1). Se sisältää uudistuksia, jotka kattavat laajemman joukon sektoreita, kuten energia, liikenne, pankki, digitaalinen infrastruktuuri, julkinen hallinto ja terveydenhuolto. Direktiivi asettaa tiukempia vaatimuksia riskienhallinnalle ja ilmoitusvelvollisuuksille, ja jäsenvaltioiden on asetettava seuraamuksia direktiivin noudattamatta jättämisestä. NIS2 koskettaa kaikkia 27 EU valtiota mukaan lukien Islanti, Norja ja Liechtenstein (Kittler & Sulaiman. 2023. 11). NIS2 on kansallisena lakina noudatettava viimeistään 17.10.2024.

2.2 CRA-direktiivin tavoitteet

CRA-direktiivi keskittyy EU-markkinoilla oleviin digitaalisten tuotteiden ja palveluiden, kuten ohjelmistojen ja älylaitteiden, kyberturvallisuuteen. Sen tavoitteena on parantaa tuotteiden ja palveluiden turvallisuutta niiden koko elinkaaren ajan, sisältäen suunnittelun, kehityksen ja ylläpidon. CRA edellyttää, että toimittajat integroivat kyberturvallisuuden osaksi tuotteidensa ja palveluidensa suunnittelua ja informoivat kuluttajia mahdollisista turvallisuuspuutteista (Kittler & Sulaiman. 2023. 37).

2.3 IEC 62443

IEC 62443 on ISA:n julkaisema kansainvälinen standardisarja, joka käsittelee kyberturvallisuutta automaatio- ja ohjausjärjestelmissä operatiivisen teknologian (OT) alueella. Se käyttää riskiperusteista lähestymistapaa määrittelemään kyberturvallisuuden näkökulmat OT:ssa. IEC 62443 jakaa kyberturvallisuuden aiheet sidosryhmien roolien mukaan, joihin kuuluvat:

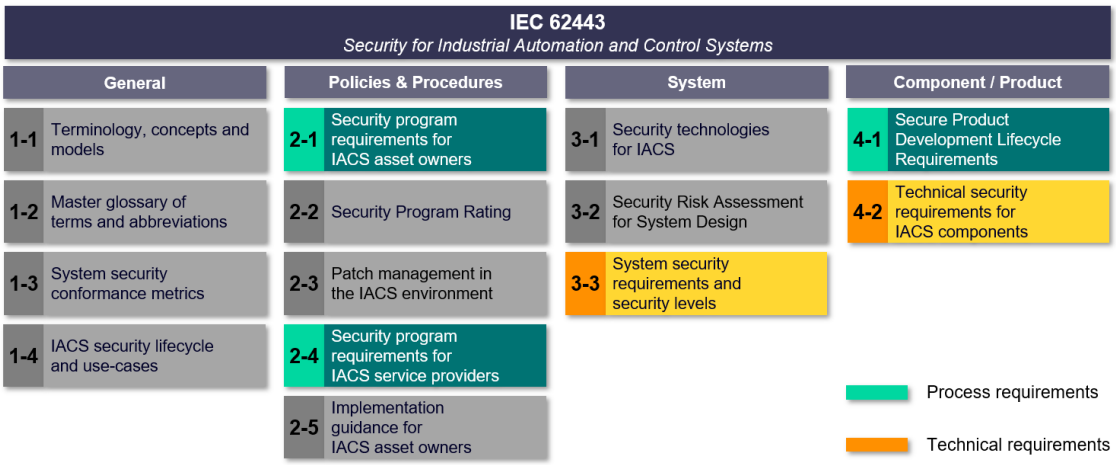
- käyttäjät
- palveluntarjoajat (palveluntarjoajat integraatiolle ja ylläpidolle)

- komponentti-/järjestelmävalmistajat

Vaikka standardi itsessään määrittelee, mitä näkökohtia tulee harkita kyberturvallisuudessa, se ei määrittele, mitä tehdä ja miten sitä tehdään. Tämä jää käyttäjän vastuulle.

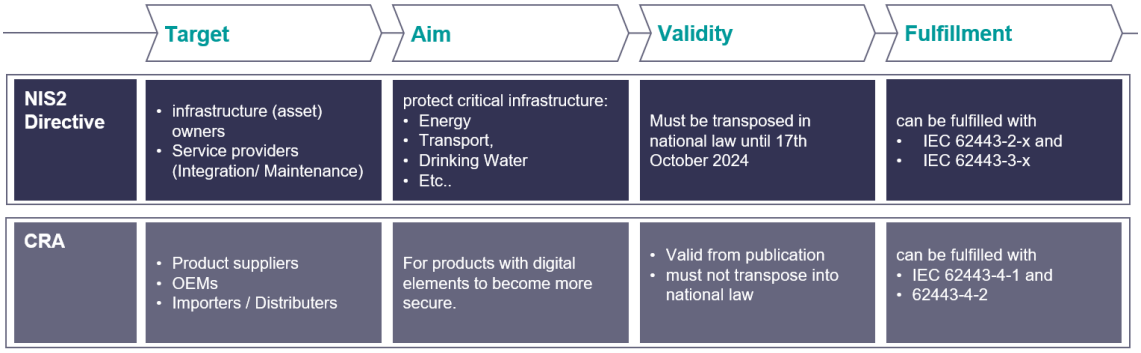
Kuten kuva 2 osoittaa, IEC 62443 voidaan jakaa neljään osaan (Kittler & Sulaiman. 2023. 41).

- Osa yksi keskittyy yleisiin aiheisiin, kuten määritelmiin ja sanastoon.
- Osa kaksi käsittelee politiikkoja ja menettelytapoja, kuten turvaohjelmavaatimuksia omaisuuden omistajille sekä palveluntarjoajille.
- Osa kolme keskittyy koko järjestelmään. Sieltä löytyvät erilaiset turvallisuustasot ja miten ne täytetään.
- Osa neljä keskittyy komponenttitasolle, 4-1 keskittyy komponentin turvallisen kehitysprosessin ja 4-2 komponentin turvallisuusvaatimusten käsittelemiseen.



Kuva 2. IEC 62443 standardin jako osa-alueisiin (Kittler & Sulaiman. 2023. 41).

NIS2 voidaan täyttää IEC62443:n normin osien 2 ja 3 avulla, kun taas CRA voidaan täyttää IEC62443:n osien 4-1 ja 4-2 avulla (Kittler & Sulaiman. 2023. 42).



Kuva 3. NIS2, CRA ja IEC62443 tiivistettynä (Kittler & Sulaiman. 2023. 42).

2.4 Automaatiojärjestelmien ja valvomoiden välisen kommunikaation kryptauksen merkitys

Automaatiojärjestelmät ovat elintärkeitä monenlaisille sektoreille, mukaan lukien energia, vesihuolto ja teollisuus, missä ne mahdollistavat prosessien sujuvan hallinnan. Valvomot toimivat näiden järjestelmien hermokeskuksina, halliten laajoja verkostoja ja varmistaen järjestelmien optimaalisen toiminnan. Tämän kriittisen kommunikaation suojaaminen on olennaisen tärkeää, sillä suojaamaton kommunikaatio tarjoaa mahdollisen hyökkäysvektorin kyberuhkille. Tässä yhteydessä NIS2 ja CRA direktiivit tuovat esiin kryptauksen keskeisen roolin, tarjoten kehyksen, jonka avulla voidaan parantaa kyberturvallisuutta ja suojata tärkeitä tietoja ulkopuolisten pääsyltä.

Koska NIS2-direktiivi keskittyy riskienhallintaan ja ilmoitusvelvollisuuksiin, se vaatii toimijoita toteuttamaan asianmukaisia teknisiä ja organisatorisia toimenpiteitä kyberuhkien torjumiseksi. Eräs tällainen toimenpide sisältää automaatiojärjestelmien ja valvomoiden välisen kommunikaation kryptauksen, joka on kriittinen toimenpide tietojen eheyden ja saatavuuden turvaamiseksi. Kryptauksen avulla voidaan estää ulkopuolisten pääsy herkkiin tietoihin ja varmistaa kommunikaation turvallisuus.

NIS2:n ja CRA:n toteutukseen liittyy teknisiä ja organisatorisia haasteita kuten muun muassa kryptauksen avainten hallinnan ja kryptausprotokollien valinta vaatien huolellista suunnittelua ja asiantuntemusta. On tärkeää varmistaa, että

kryptausmenetelmät eivät heikennä järjestelmän suorituskykyä tai käytettävyyttä erityisesti reaaliaikaista prosessointia vaativissa automaatiojärjestelmissä.

Yhteenvetona, NIS2 ja CRA direktiivit muodostavat vankan perustan automaatiojärjestelmien ja valvomoiden välisen kommunikaation turvallisuuden parantamiselle, korostaen kryptauksen kriittistä merkitystä kyberuhkien torjunnassa. Direktiivien täytäntöönpano vaatii organisaatioilta merkittäviä toimenpiteitä, mutta tarjoaa samalla mahdollisuuden vahvistaa luottamusta digitaalisiin järjestelmiin ja suojata yhteiskuntaa laajenevien kyberuhkien keskellä. Kryptauksen tehokas käyttö on keskeinen osa tätä prosessia, varmistaen kriittisen kommunikaation turvallisuuden ja integroiden kyberturvallisuuden osaksi organisaatioiden toimintaa.

NIS2 ja CRA direktiivien täytäntöönpano edellyttää yrityksiltä merkittäviä ponnisteluja, mutta ne ovat välttämättömiä askelia kohti turvallisempaa ja kestävämpää digitaalista tulevaisuutta. Direktiivien onnistunut implementointi parantaa merkittävästi automaatiojärjestelmien turvallisuutta, edistää yhteistyötä ja tietojenvaihtoa, ja integroi kyberturvallisuuden osaksi tuotteita ja palveluita.

3 Kryptografia

Nykypäivän digitaalisessa maailmassa, jossa tietoliikenne ja tiedonvaihto ovat jatkuvasti lisääntyneet ja monimutkaistuneet, tietoturvan merkitys on noussut entistä keskeisempään asemaan. Sertifikaatit ja niiden elinkaari, asymmetrinen ja symmetrinen salaus, TLS sekä Public Key -infrastruktuuri ovat olennaisia elementtejä, jotka muodostavat yhdessä vahvan perustan digitaalisen maailman tietoturvamekanismeille (Carlisle Adams, Steve Lloyd. 2023. 8). Nämä tekniikat eivät ainoastaan suojaa tietoa luvattomalta käytöltä ja varmistavat tiedon eheyden, vaan myös takaavat kommunikaation luottamuksellisuuden ja osapuolten aitouden digitaalisessa ympäristössä (Siemens. 2023f. 13).

Sertifikaatit ja niiden elinkaari ovat keskeisessä roolissa digitaalisten identiteettien todentamisessa. Digitaaliset sertifikaatit, jotka myönnetään luotettavien tahojen toimesta, kuten sertifikaatin myöntäjien (CA) kautta, varmistavat verkkosivustojen ja muiden palveluiden aitouden. Sertifikaattien elinkaari – myöntämisestä uusimiseen ja vanhentumiseen – on hallittu prosessi, joka takaa, että vain voimassa olevat ja aidoiksi todistetut sertifikaatit ovat käytössä, mikä estää vanhentuneiden tai vaarantuneiden sertifikaattien käytön (IETF RFC 5280. 2008. 10-11).

Asymmetrinen ja symmetrinen salaus muodostavat kaksi perustaa tietojen salaukselle, jossa asymmetrinen salaus käyttää kahta erillistä avainta (julkinen ja yksityinen) tiedon salaukseen ja purkuun, kun taas symmetrinen salaus käyttää samaa avainta molempiin toimintoihin. Asymmetrinen salaus on olennainen osa digitaalisten allekirjoitusten ja sertifikaattien todentamisessa, kun taas symmetrinen salaus tarjoaa nopean ja tehokkaan tavan salata suuria datamääriä. Yhdessä nämä salausmenetelmät mahdollistavat sekä tiedon turvallisen siirron että osapuolten aitouden varmistamisen (Siemens. 2023f. 48).

TLS-protokolla on keskeinen tekniikka, joka käyttää sekä asymmetrista että symmetristä salausta turvatakseen internetin yli tapahtuvan datan siirron. TLS suojaa yhteyksiä esimerkiksi verkkopankkipalveluissa, sähköpostiviestinnässä ja verkkokaupoissa, varmistaen, että siirretty tieto on sekä salattua että

muuttumatonta matkansa aikana. TLS käyttää digitaalisia sertifikaatteja osapuolten todentamiseen, taaten näin yhteyden aitouden (Stephen A. Thomas. 2000. 53).

Public Key -infrastruktuuri (PKI) on järjestelmä, joka yhdistää digitaaliset sertifikaatit, niiden myöntämisen, jakelun ja vahvistamisen yhdeksi kokonaisuudeksi, mahdollistaen näin luotettavan digitaalisen identiteetin hallinnan. PKI on perusta, joka mahdollistaa edellä mainittujen teknologioiden yhteistoiminnan, tarjoten kattavan ratkaisun digitaalisen viestinnän tietoturvaan (Siemens. 2023f. 48).

Näiden teknologioiden yhdistelmä luo turvallisen ympäristön digitaaliselle viestinnälle, jossa tietoturvan kolme peruspilaria – luottamuksellisuus, eheys ja saatavuus – on taattu. Tämän seurauksena, niiden ymmärtäminen ja oikea käyttö ovat keskeisiä taitoja kaikille, jotka toimivat digitaalisessa maailmassa, olipa kyse sitten yksilöistä tai organisaatioista.

3.1 Sertifikaatit ja niiden elinkaari

Digitaaliset sertifikaatit ovat keskeinen tekijä nykyaikaisessa digitaalisessa todentamisessa, mahdollistaen luotettavan kommunikaation osapuolten välillä internetissä. Niiden ymmärtäminen ja tehokas hallinta ovat tärkeitä taitoja kaikille, jotka työskentelevät tietotekniikan ja kyberturvallisuuden parissa. Tässä yhteydessä käsittelemme digitaalisten sertifikaattien roolia, toimintaa, elinkaarta ja niihin liittyviä hallinnollisia haasteita.

3.1.1 Määritelmä ja tarkoitus

Digitaaliset sertifikaatit ovat sähköisiä todisteita, jotka linkittävät organisaation tai yksilön digitaalisen identiteetin julkiseen avaimen kryptografiseen avainpariin. Ne ovat digitaalisen todentamisen perusta, joka mahdollistaa

luotettavan ja turvallisen viestinnän kahden osapuolen välillä internetissä. Sertifikaatit varmistavat, että julkisen avaimen omistaja on se, kuka väittää olevansa, ja ne ovat keskeisiä SSL/TLS-protokollassa, joka suojaa muun muassa verkkosivustoja (Stephen A. Thomas. 2000. 92).

Sertifikaattien avulla voidaan varmistaa viestinnän luottamuksellisuus ja eheys, sillä ne mahdollistavat tiedon salauksen ja digitaalisten allekirjoitusten käytön. Tämä on tärkeää esimerkiksi verkkopankkipalveluissa, sähköisessä kaupankäynnissä ja muissa verkkopalveluissa, joissa käyttäjien on voitava luottaa siihen, että heidän yksityisyytensä ja tietonsa ovat suojattuja.

Digitaalisen sertifikaatin elinkaari

Digitaalisen sertifikaatin elinkaari koostuu useista vaiheista (Sectigo. 2023):

- **Myöntäminen:** Luotettu kolmas osapuoli, sertifikaatin myöntäjä (CA) todentaa hakijan identiteetin ja myöntää sertifikaatin, joka sisältää julkisen avaimen ja hakijan identiteetin.
- **Käyttö:** Sertifikaattia käytetään digitaalisessa todentamisessa, salauksessa ja digitaalisissa allekirjoituksissa.
- **Uusiminen:** Sertifikaatit ovat voimassa rajoitetun ajan. Ennen vanhenemistaan ne on uusittava, mikä yleensä vaatii hakijan identiteetin uudelleentodentamisen.
- **Mitätöinti:** Jos sertifikaatti on vaarantunut tai sen tietoja on muutettava, se voidaan mitätöidä ennen sen voimassaoloajan päättymistä.

3.1.2 Hallinta ja haasteet

Sertifikaattien hallinta suurissa organisaatioissa voi olla haastavaa. Hallinnollisiin haasteisiin kuuluu sertifikaattien elinkaaren ylläpito, mukaan lukien niiden uusiminen, varmuuskopiointi ja mitätöinti, sekä sertifikaattien ja niiden yksityisten avainten turvallinen säilyttäminen.

Sertifikaattien seuranta on tärkeää, mutta samalla usein teknisesti haastavaa. Organisaation on kyettävä seuraamaan kaikkien sertifikaattiensa voimassaoloaikoja ja uusimaan ne ajoissa.

Sertifikaattien ja niihin kuuluvien yksityisten avainten suojaamatta jättäminen voi johtaa tietoturvariskeihin.

Haasteena voi myös olla automaation puute. Manuaalinen sertifikaattien hallinta on työlästä ja altistaa virheille. Automaattiset työkalut ja prosessit voivat auttaa hallitsemaan sertifikaatteja tehokkaammin.

Tehokas sertifikaattien hallinta edellyttää prosessien ja työkalujen kehittämistä sertifikaattien elinkaaren hallitsemiseksi automaattisesti ja turvallisesti. Tämä voi sisältää sertifikaattien hallintajärjestelmiä, jotka automatisoivat uusimisen ja mitätöinnin prosesseja, sekä turvallisia säilytysratkaisuja yksityisten avainten suojaamiseksi.

Digitaalisten sertifikaattien ymmärtäminen ja hallinta on olennainen osa nykyaikaisen tietoturvan perustaa. Niiden avulla voidaan varmistaa digitaalisen maailman luottamuksellisuus, eheys ja osapuolten aitouden todentaminen.

3.2 Asymmetrinen ja symmetrinen salaus

Salaus on keskeinen osa digitaalista tietoturvaa, suojaten tietoja luvattomalta käytöltä. Salaustekniikat jaetaan kahteen pääluokkaan: symmetriseen ja asymmetriseen salaukseen. Kumpikin menetelmä tarjoaa omat etunsa ja haasteensa, ja niitä käytetään eri tilanteissa riippuen tarvittavasta turvallisuustasosta ja käyttötarkoituksesta (Dionisie Gitlan. 2024a).

3.2.1 Symmetrinen salaus

Symmetrinen salaus on vanhin ja yksinkertaisin salausmenetelmä, jossa sekä tiedon salaamiseen että purkamiseen käytetään samaa avainta. Tämän

menetelmän etuna on sen nopeus, mikä tekee siitä tehokkaan suurten datamäärien käsittelyssä (Dionisie Gitlan. 2024a).

3.2.2 Symmetrisen salauksen toimintaperiaate

Symmetrisessä salauksessa lähettäjä ja vastaanottaja jakavat saman salausavaimen. Lähettäjä käyttää tätä avainta salatakseen viestin, ja vastaanottaja käyttää samaa avainta viestin purkamiseen (NIST SP 800-57. 2020a. 28).

Symmetrisen salauksen suurin etu on nopeus. Symmetrinen salaus on nopeampi kuin asymmetrinen salaus, mikä tekee siitä sopivan suurten datamäärien käsittelyyn. Toinen mainittava arvoinen asia on yksinkertaisuus. Algoritmit ovat yksinkertaisempia, mikä helpottaa implementointia (Dionisie Gitlan. 2024a).

Tämän menetelmän haasteena puolestaan on avaimen turvallinen siirto osapuolten välillä. Suurissa järjestelmissä, joissa on monta käyttäjää, avainhallinta voi muodostua monimutkaiseksi (Dionisie Gitlan. 2024a).

Esimerkkejä algoritmeista:

- AES (Advanced Encryption Standard)
- DES (Data Encryption Standard)
- 3DES (Triple DES)

3.2.3 Asymmetrinen salaus

Toisin kuin symmetrinen salaus, asymmetrinen salaus käyttää kahta avainta: julkista avainta tiedon salaamiseen ja yksityistä avainta sen purkamiseen. Julkinen avain on vapaasti jaettavissa, mutta yksityinen avain pidetään salassa (NIST SP 800-57. 2020b. 15).

3.2.4 Asymmetrisen salauksen toimintaperiaate

Asymmetrisessa salauksessa jokaisella käyttäjällä on avainpari: julkinen avain, joka jaetaan vapaasti, ja yksityinen avain, joka on vain avaimen omistajan tiedossa. Jos esimerkiksi Pepe haluaa lähettää salatun viestin Mikaelille, hän salaa viestin Mikaelin julkisella avaimella. Vain Mikaelin yksityinen avain voi purkaa viestin, mikä takaa viestin turvallisuuden (Siemens. 2023f. 52-53).

Asymmetrisen salauksen tärkein etu on turvallisuus. Avaimen jakaminen ei ole ongelma, koska julkista avainta voidaan jakaa turvallisesti. Toinen merkittävä etu on mahdollisuus käyttää digitaalisia allekirjoituksia, jotka varmistavat viestin alkuperän ja eheyden. Haasteina puolestaan on nopeus. Asymmetrinen salaus on hitaampi kuin symmetrinen salaus, mikä voi olla haitta suurien datamäärien käsittelyssä (Dionisie Gitlan. 2024a).

Toinen hankaloittava tekijä on algoritmien monimutkaisuus, mikä voi tehdä niiden toteutuksesta haastavampaa.

Esimerkkejä algoritmeista:

- RSA (Rivest-Shamir-Adleman)
- ECC (Elliptic Curve Cryptography)
- DSA (Digital Signature Algorithm)

3.3 Sovellusalueet

Symmetristä salausta käytetään yleisesti tilanteissa, joissa datamäärien on oltava suuria ja salausprosessin nopea, kuten tiedostojen salauksessa levylle tai videon striimauksessa. Asymmetrista salausta käytetään usein silloin, kun tarvitaan vahvaa todentamista ja turvallista avainten vaihtoa, kuten sähköisessä asioinnissa, digitaalisissa allekirjoituksissa ja SSL/TLS-yhteyksissä, jotka suojaavat verkkosivustoja (Stephen A. Thomas. 2000. 92).

Yhdessä nämä salausmenetelmät tarjoavat kattavan ratkaisun digitaalisen maailman tietoturvaan, mahdollistaen sekä tehokkaan datan salauksen että turvallisen avainten hallinnan ja vaihdon.

4 TLS

Transport Layer Security (TLS) on kryptografisten protokollien sarja, joka mahdollistaa turvallisen viestinnän verkkojen välillä ja on internetissä hyvin laajalti käytössä. TLS on SSL-protokollan seuraaja ja se on suunniteltu estämään tietojen vakoilu, tiedon muuttaminen ja viestien väärinkäyttö (Stephen A. Thomas. 2000. 1).

TLS on tärkeä osa nykyaikaista internetin tietoturvaa, sillä se mahdollistaa luottamuksellisen ja muuttumattoman tiedonsiirron muutoin epäluotettavien verkkojen yli suojaten muun muassa verkkopankkitoimintaa, sähköpostiviestintää ja sosiaalisen median sivustoja. TLS:n käyttö takaa, että käyttäjien tiedot pysyvät suojattuina ja että vain oikeutetut osapuolet voivat tarkastella niitä.

4.1 Toimintaperiaate

TLS:n toiminta koostuu kolmesta päävaiheesta: kättely, salauksen neuvottelu ja tiedonsiirto (Dionisie Gitlan. 2024b).

Kättelyvaihe aloittaa yhteyden ja määrittää protokollan version, valitsee salausalgoritmit, vaihtaa avainten vaihtotiedot ja varmistaa osapuolten aitouden käyttäen asymmetrista salausta avainten vaihtoon ja todentamiseen sekä digitaalisia sertifikaatteja palvelimen ja asiakkaan aitouden vahvistamiseen.

Salauksen neuvotteluvaiheessa osapuolet sopivat käytettävistä salausalgoritmeista, jotka tukevat useita eri algoritmeja perustuen yhteensopivuuteen ja turvallisuusvaatimuksiin. Tässä vaiheessa luodaan symmetrinen salausavain, jota käytetään istunnon aikana tiedon salaamiseen ja purkamiseen.

Tiedonsiirron vaihe alkaa, kun kättely on suoritettu ja salausalgoritmit on sovittu, ja siinä kaikki lähetettävä ja vastaanotettava data salataan neuvotellulla

symmetrisellä salausmenetelmällä. Tämä takaa siirrettävän tiedon luottamuksellisuuden ja eheyden.

4.2 TLS ja sertifikaatit

TLS:n ja digitaalisten sertifikaattien yhteistyö on keskeinen osa protokollan toimintaa ja osapuolten aitouden varmistamista. TLS käyttää digitaalisia sertifikaatteja, jotka myöntää luotettu sertifikaatin myöntäjä (CA), varmistaakseen palvelimen ja mahdollisesti asiakkaan aitouden kättelyvaiheen aikana. Sertifikaatti sisältää palvelimen julkisen avaimen ja mahdollistaa asiakkaan varmistaa, että se on yhteydessä oikeaan palvelimeen ja että avainten vaihto on turvallista.

Kun asiakasohjelma (esimerkiksi verkkoselain) yhdistää TLS:llä suojattuun palvelimeen, palvelin esittää ensin digitaalisen sertifikaattinsa osana kättelyprosessia. Asiakasohjelma tarkistaa sertifikaatin aitouden varmistaakseen, että sertifikaatti on voimassa, se ei ole vanhentunut, se on myönnetty sivustolle, johon yhteyttä ollaan muodostamassa, ja että se on luotetun CA:n allekirjoittama. Tämän varmistuksen jälkeen voidaan turvallisesti jatkaa avainten vaihtoa ja salauksen neuvottelua, tietäen, että yhteys on aito (Hazem Sulaiman. 2022. 37).

TLS:n ja digitaalisten sertifikaattien yhteiskäyttö tarjoaa vahvan perustan turvalliselle viestinnälle internetissä, suojaten käyttäjätietoja ja varmistaen osapuolten aitouden. Tämä prosessi minimoi riskin, että tietoja voitaisiin kaapata tai että käyttäjät yhdistäisivät vahingossa haitallisiin palvelimiin.

5 Public Key (julkinen avain)

Julkisen avaimen kryptografia, tunnetaan myös nimellä asymmetrinen salaus, on salausmenetelmä, jossa käytetään kahta erillistä avainta: julkista avainta ja yksityistä avainta. Nämä avaimet muodostavat avainparin, jossa julkista avainta voidaan jakaa vapaasti, mutta yksityinen avain pidetään salassa. Julkisen avaimen rooli digitaalisessa todentamisessa ja salauksessa on kriittinen, sillä se mahdollistaa turvallisen tiedonsiirron ja varmennetun kommunikaation kahden osapuolen välillä (NIST SP 800-57. 2020b. 88).

5.1 Määritelmä ja käyttötarkoitus

Julkiset avaimet ovat olennainen osa digitaalista todentamista ja salattua viestintää. Ne toimivat osana asymmetrista salausjärjestelmää, jossa kuka tahansa voi käyttää julkista avainta salatakseen viestin tai tarkastaakseen digitaalisen allekirjoituksen. Vain vastaavan yksityisen avaimen haltija voi purkaa kyseisen viestin salauksen tai luoda digitaalisen allekirjoituksen, joka voidaan todentaa julkisella avaimella. Tämä mahdollistaa kahden päätoiminnon: luottamuksellisen tiedonsiirron ja digitaalisen allekirjoituksen.

Digitaaliset allekirjoitukset

Yksi julkisen avaimen kryptografian yleisimmistä sovelluksista on digitaalinen allekirjoitus. Tässä prosessissa lähettäjä luo viestistä tiivisteen (hash), jonka hän allekirjoittaa yksityisellä avaimellaan. Vastaanottaja voi sitten käyttää lähettäjän julkista avainta varmistaakseen allekirjoituksen. Jos tiiviste vastaa viestin sisältöä, vastaanottaja voi olla varma viestin eheydestä ja lähettäjän identiteetistä. Digitaaliset allekirjoitukset ovat keskeisiä sähköisessä kaupankäynnissä, sähköisessä äänestyksessä ja digitaalisissa sopimuksissa (Baivab Kumar Jena. 2024).

Salauksen purkaminen

Julkista avainta voidaan käyttää myös viestien salaamiseen siten, että vain tietty vastaanottaja, jolla on vastaava yksityinen avain, voi purkaa viestin salauksen. Tämä on erityisen hyödyllistä esimerkiksi sähköpostin salauksessa, jossa lähettäjä voi salata viestin vastaanottajan julkisella avaimella, varmistaen, että vain vastaanottaja pystyy lukemaan viestin (Siemens. 2023f. 48).

SSL/TLS yhteydet

Julkisia avaimia käytetään laajalti SSL/TLS-protokolloissa, jotka suojaavat verkkoviestintää. Kun webselain yhdistää TLS:llä suojattuun verkkosivustoon, se käyttää sivuston julkista avainta salatakseen yhteyden aikana vaihdettavat symmetrisen salauksen avaimet. Tämä varmistaa, että vain verkkosivusto pystyy purkamaan symmetrisen avaimen ja sitä kautta koko istunnon ajan vaihdettavan datan.

Julkisen avaimen infrastruktuuri (PKI)

PKI-järjestelmissä julkiset avaimet ovat keskeisiä sertifikaattien myöntämisessä ja hallinnassa. Sertifikaatit, jotka ovat digitaalisia asiakirjoja vahvistamassa avaimen omistajuuden, sisältävät julkisen avaimen ja ovat allekirjoitettu luotettavan tahon, kuten sertifikaatin myöntäjän (CA), toimesta. Tämä mahdollistaa osapuolten aitouden varmistamisen ja luotettavan kommunikaation internetissä (Stephen A. Thomas. 2000. 55).

Julkisen avaimen käyttö näissä sovelluksissa osoittaa sen monipuolisuuden ja merkityksen nykyaikaisessa digitaalisessa yhteiskunnassa. Se on avain turvallisen ja luotettavan digitaalisen identiteetin, viestinnän ja transaktioiden mahdollistamisessa.

5.2 Yhteenveto

Digitaaliset sertifikaatit, asymmetrinen ja symmetrinen salaus sekä TLS-protokolla muodostavat yhdessä vankan perustan nykyaikaiselle tietoturvalle ja turvalliselle digitaaliselle viestinnälle. Digitaaliset sertifikaatit ja niiden hallinta mahdollistavat luotettavien digitaalisten identiteettien luomisen, varmistaen, että kommunikaatio tapahtuu aitojen osapuolten välillä. Asymmetrinen salaus, johon kuuluu julkisen ja yksityisen avaimen käyttö, mahdollistaa turvallisen tiedonvaihdon ja digitaalisten allekirjoitusten luomisen, tarjoten sekä salauksen että todentamisen mekanismit. Symmetrinen salaus puolestaan tarjoaa nopean ja tehokkaan tavan suurten datamäärien salaamiseen, kun avaimet on jo turvallisesti vaihdettu.

TLS-protokolla, joka käyttää sekä asymmetristä että symmetristä salausta, on keskeinen turvallisen internet-viestinnän mahdollistaja, suojaten esimerkiksi verkkopankkitoimintaa ja sähköistä kaupankäyntiä. Yhdessä nämä tekniikat takaavat, että digitaalisessa maailmassa voidaan viestiä turvallisesti, varmistaen tiedon luottamuksellisuuden, eheyden ja saatavuuden. Ne ovat nykyaikaisen tietoturvan kulmakiviä, jotka mahdollistavat luotettavan ja suojatun digitaalisen ympäristön niin yksilöille kuin organisaatioillekin.

6 Simatic S7-1500 -ohjausjärjestelmä

S7-1500 on Siemensin kehittämä teollisuusautomaation ohjelmoitava logiikkaohjain (PLC), joka on suunniteltu monimutkaisten automaatiotehtävien hallintaan.

S7-1500:n valinta opinnäytetyön tarkasteluun perustui sen hyvin laajaan käyttöön teollisuudessa (Mordor Intelligence. 2024).

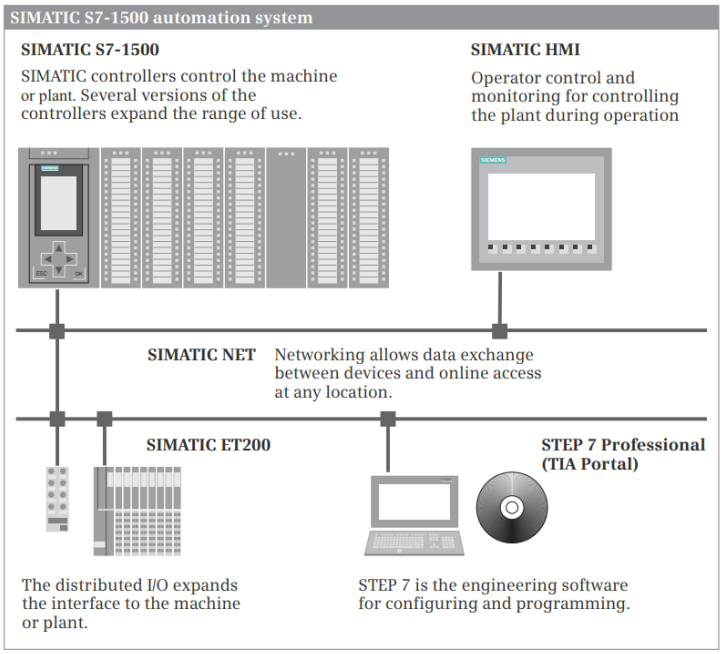
S7-1500 eroaa vanhemmista S7-300 ja S7-400-sarjoista tarjoamalla paremman suorituskyvyn, modernimmat turvallisuusominaisuudet ja laajemmat kommunikaatiomahdollisuudet, mukaan lukien sisäänrakennettu tuki Profinet-verkolle ja on suunniteltu käytettäväksi TIA Portal -ohjelmointiympäristössä (Kuva 4), mahdollistaen intuitiivisemman ohjelmoinnin ja järjestelmän hallinnan.

Vastaavia järjestelmiä markkinoilla ovat esimerkiksi Rockwell Automationin Allen-Bradley ControlLogix, Mitsubishi Electricin iQ-R-sarja ja Omronin Sysmac-sarja.

S7-1500 sisältää modulaarisia ohjainkortteja, joita on helppo laajentaa applikaation tarpeita ajatellen. S7-1500-tuoteperheen ohella on olemassa myös S7-1200, ET200SP ja S7-1500 Software Controller -tuotteet, jotka perustuvat samaan keskeiseen teknologiaan kuin S7-1500 mitä tulee digitaalisten varmenteiden osalta (Hans Berger, 2014, 23-25).

S7-1500 -ohjausjärjestelmää käytetään laajasti erilaisissa teollisuuden sovelluksissa, kuten:

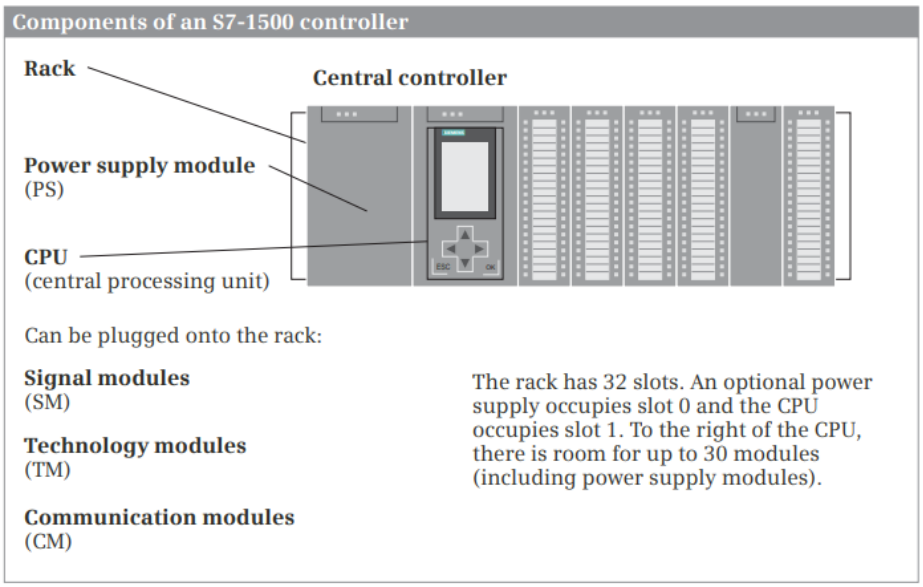
- Prosessinohjaus: Kemian teollisuus, elintarviketuotanto, vesi- ja jätevesikäsittely.
- Koneenohjaus: Pakkauskoneet, muovauskoneet, metallintyöstökoneet.
- Tuotantolinjojen automatisointi: Autoteollisuus, elektroniikkateollisuus, kuljetinjärjestelmät.



Kuva 4. SIMATIC S7-1500 automaatiojärjestelmän komponentit (Hans Berger. 2014).

6.1 Modulaarisuus ja laajennettavuus

Modulaarinen rakenne (Kuva 5) on yksi S7-1500:n keskeisistä ominaisuuksista. Järjestelmä koostuu useista moduuleista, kuten keskusyksiköistä (CPU), signaali- ja kommunikaatiomoduuleista sekä teknologiamoduuleista, jotka voidaan yhdistää tarpeen mukaan. Tämä mahdollistaa järjestelmän mukauttamisen ja laajentamisen sovelluksen määrittelyn mukaisesti.



Kuva 5. Modulaarinen rakenne (Hans Berger. 2014. 24).

6.2 Integroitu turvateknologia

Turvallisuus on keskeinen osa S7-1500 -ohjausjärjestelmää sisältäen integroidun turvateknologian, joka mahdollistaa sekä standardi- että turvasignaalien käsittelyn samassa järjestelmässä. Integroitu turvateknologia vähentää erillisten turvajärjestelmien tarvetta ja yksinkertaistaa järjestelmän rakennetta.

6.3 Avoin kommunikointi

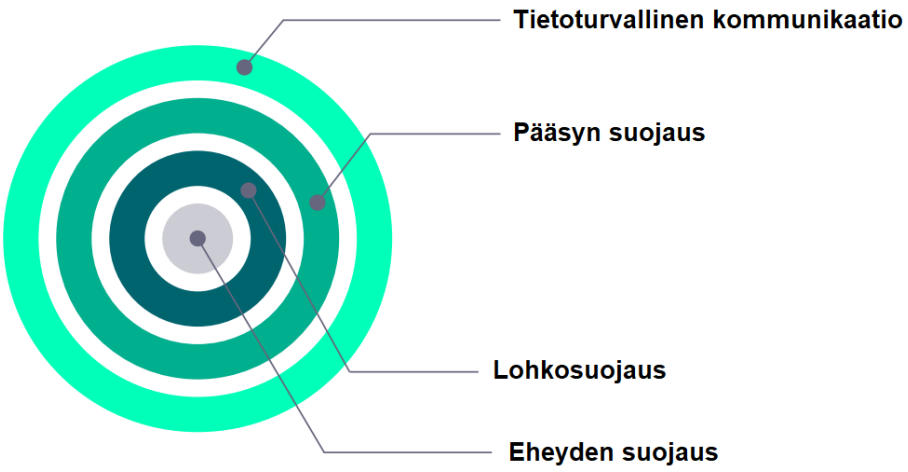
S7-1500 tukee laajaa valikoimaa kommunikaatiostandardeja ja -protokollia, mukaan lukien PROFINET, PROFIBUS, OPC UA, MODBUS TCP ja MQTT tehden siitä helposti yhdistettävän muihin järjestelmiin ja sovelluksiin mahdollistaen monipuoliset kommunikointiratkaisut (Siemens. 2023b).

7 Tietoturvamekanismit S7-1500 CPU:ssa

Siemens SIMATIC S7-1500 -sarjan ohjaimet ovat laajalti käytössä (Mordor Intelligence. 2024) teollisuusautomaatiossa ja ne tarjoavat monipuolisia toimintoja prosessien ohjaukseen ja seurantaan. Tänä päivänä ja myös jatkossa tietoturva on kriittinen osa modernia teollisuusautomaatiota, sillä se suojaa järjestelmiä luvattomalta käytöltä ja tietovuodoilta.

S7-CPU:n tietoturvamekanismit voidaan jakaa 4 vaiheeseen (Kuva 6):

- Tietoturvallinen kommunikaatio
    - Turvallinen PG/HMI-viestintä
    - Turvallinen avoin käyttäjäviestintä
    - Turvallinen OPC UA -viestintä
- Pääsyn suojaus
    - Paikan päällä tapahtuvan pääsyn rajoitus
    - Projektipääsyn suojaus
    - Online-pääsyn ja toiminnon rajoitus
- Lohkosuojaus
    - Osaamisen suojaus
    - Kopiosuojaus
    - Kirjoitussuojaus
- Eheyden suojaus
    - Arkaluontoisten PLC-konfiguraatiotietojen suojaus
    - Firmwaren allekirjoitus



Kuva 6. Tietoturvamekanismit S7-CPU:ssa (Siemens. 2023a. 10).

7.1 TLS ja S7-1500

TLS on protokolla, joka suojaa verkkoviestintää salauksen avulla. Se estää ulkopuolisten pääsyn siirrettyyn dataan ja varmistaa, että tiedot ovat muuttumattomia ja peräisin todennetusta lähteestä. S7-1500 -ohjaimissa TLS -protokollaa käytetään varmistamaan, että ohjaimen ja esimerkiksi ohjelmointi- tai valvontalaitteiden välillä siirtyvä tieto on suojattua. Tämä on erityisen oleellista, kun ohjaimia käytetään kriittisissä prosesseissa, joissa tiedon eheys ja luottamuksellisuus ovat ensiarvoisen tärkeitä.

TIA Portal V17 ja S7-1500 -ohjaimien firmware 2.9-versiosta lähtien TLS 1.3 toimii vakiona nativisti osana S7-1500 -ohjaimen tietoturvaa (Siemens. 2022a. 13). Tuettu TLS versio on 1.3 ja se on taaksepäin yhteensopiva TLS 1.2 version kanssa (Taulukko 1). TLS 1.3 sisältää lukuisia parannuksia verrattuna TLS 1.2 -versioon ja sen käyttö on suositeltavaa (Eric Rescorla. 2018).

Taulukko 1. TLS-tuki suhteessa S7-CPU-versioon (Siemens. 2023f. 51).

S71500 CPU versio	Tuettu TLS versio
V3.0	TLS 1.2, TLS 1.3
V2.9	TLS 1.2, TLS 1.3
V2.8 … V2.0	TLS 1.2

S7-1500:ssa on 64 resurssia avaimille. Luotettu juurisertifikaatti vaatii yhden resurssin (julkinen avain). Laitesertifikaatti vaatii 2 resurssia (yksityinen + julkinen avain) (Hazem Sulaiman. 2022. 53). Riippuen CPU:n mallista kommunikaation resurssit jakautuvat kiinteiden ja dynaamisesti varattujen resurssien kesken (Kuva 7).

S7-1500 käyttää sertifikaatteja oheisiin kommunikaatiopalveluihin.

- PG/HMI-kommunikaatio logiikan ja valvomon välillä
- Turvallinen ja suojattu avoin OUC-sovelluslohkoilla toteutettu TCP/IP -pohjainen viestintä
- Web-palvelin
- OPC UA (asiakas/palvelin)
- VPN-tunnelit
- Salatut sähköpostit (CP 1543-1)

Connection resources

	Station resources			Module resources	Module resources
	Reserved		Dynamic	PLC_1 [CPU 1516...	CM 1542-1_1 [C...
Maximum number of resources:	10		182	128	64
	Maximum	Configured	Configured	Configured	Configured
PG communication:	4	-	-	-	-
HMI communication:	4	0	0	0	0
S7 communication:	0	-	13	6	7
OUC:	0	-	0	0	0
Web communication:	2	-	-	-	-
Other communication:	-	-	0	0	0
Total resources used:		0	13	6	7
Available resources:		10	169	122	57

Kuva 7. TIA Portal diagnostiikkanäkymä, jossa selviää käytetyt ja varatut yhteydet (Hazem Sulaiman. 2022. 53).

7.2 TLS:n toimintaperiaate S7-1500 -ohjaimessa

Yhteyden muodostamisen alussa S7-1500 -ohjain ja toinen laite suorittavat TLS-kättelyn, jossa sovitaan käytettävästä TLS-versiosta, salausalgoritmeista ja avainvaihtomenetelmistä (William Stallings. 2017. 550). Samalla varmistetaan osapuolten aitous käyttäen digitaalisia sertifikaatteja. Kättelyvaiheessa neuvotellaan myös siitä, että tiedonsiirtoa varten käytetään symmetristä avainta, joka on istuntokohtainen ja joka luodaan turvallisesti käyttäen asymmetrista

salausta avainten vaihtoon. Kun kättelyvaihe on saatu päätökseen ja symmetrinen avain on vaihdettu, kaikki viestintä näiden osapuolten välillä salataan tällä avaimella, mikä varmistaa, että tiedot pysyvät luottamuksellisina ja muuttumattomina siirron aikana.

7.2.1 Alkuyhteysasetusten määrittäminen S7-1500 CPU:lle - Valmisteluvaihe

Kuva 8 selittää alkuyhteysasetusten määrittämisen ohjelmointilaitteesta tai HMI-paneelista CPU:lle. Tätä kutsutaan "valmisteluvaiheeksi".

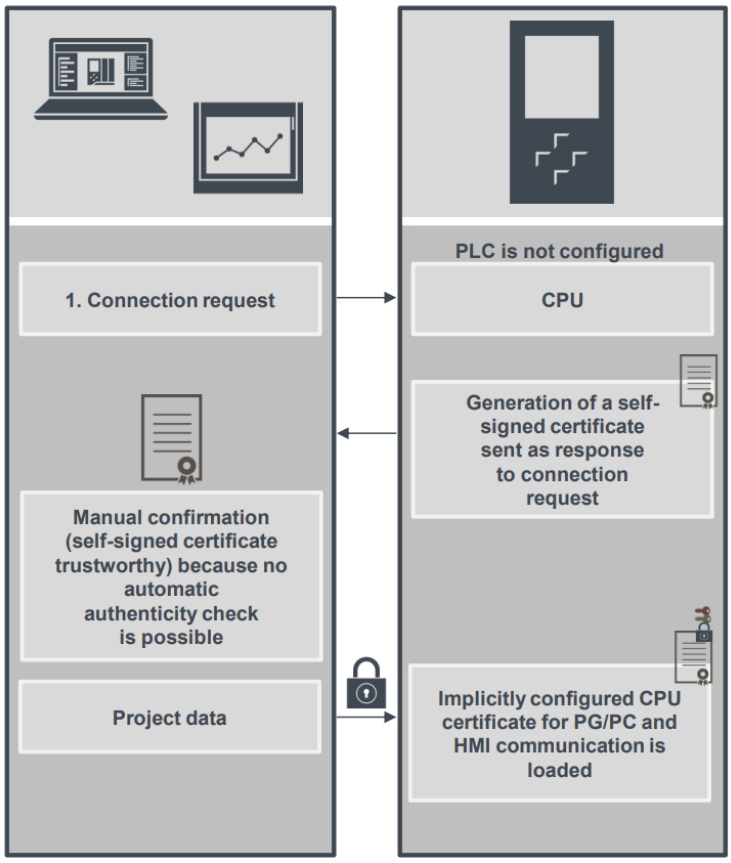
Alkuyhteysasetusten määrittäminen S7-CPU:lle lataamista varten on suojattu TLS-protokollalla tietoturvallisen PG/HMI-viestinnän mallissa. Tässä yhteydenmuodostuksen vaiheessa CPU käyttää valmistajan laitetodistusta (jos olemassa) tai itse allekirjoitettua todistusta. S7-CPU:n käyttö on rajoitettua tässä vaiheessa. Tässä vaiheessa S7-CPU odottaa salasanaan perustuvan avaintiedon toimittamista. Toisin sanoen se odottaa salasanaa arkaluonteisille PLC-konfiguraatiotiedoille. Tätä vaihetta kutsutaan "valmisteluvaiheeksi". Prosessin aikana CPU ilmoittaa olevansa valmisteluvaiheessa vastaavalla viestillä diagnostiikkapuskurissa.

7.2.2 Valmisteluvaiheen päättäminen

Salasana arkaluonteisille PLC-konfiguraatiotiedoille ja/tai salasanasta generoitua avaintietoa ei tallenneta tietokoneella olevaan projektiin TIA Portalissa. Siksi salasanaa pyydetään sarjassa dialogeja alkulatauksen aikana tai kun uusi projekti ladataan. Kyselydialogien jälkeen se sitten siirretään S7-1500 CPU:lle. Vain edellä mainittujen vaiheiden jälkeen S7-1500 CPU pystyy käyttämään suojattuja PLC-konfiguraatiotietoja ja tämä päättää valmisteluvaiheen ja CPU voi siirtyä RUN-tilaan (Siemens. 2020a. 13).

7.2.3 PG/HMI-viestinnän käynnistys

Kun S7-1500 CPU:n ohjelma on kokonaisuudessaan ladattu ja se on saanut CPU-sertifikaatin turvalliseen PG/HMI-kommunikointiin, ohjelmointilaite tai valvomo voi suorittaa uudelleen yhteyden muodostamisen ja tällä kertaa ladatun CPU-sertifikaatin perusteella.

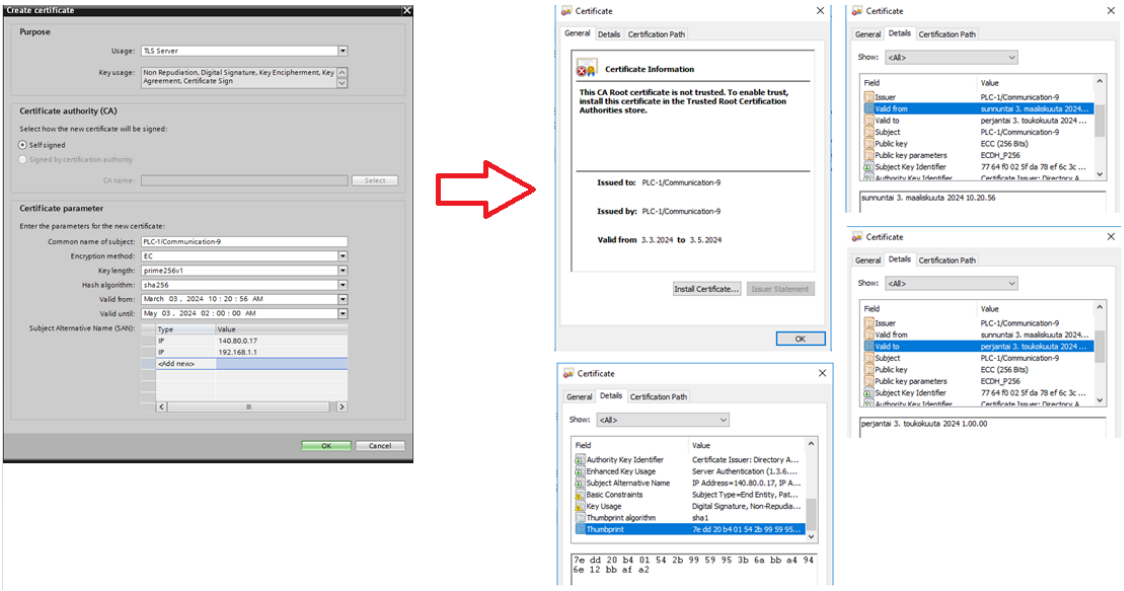


Kuva 8. TLS-yhteyden alustuksen valmisteluvaihe (Siemens. 2020a. 14).

7.3 Sertifikaatit S7-1500 -ohjaimessa

Sertifikaatit, jotka sisältävät julkisen avaimen ja ovat luotettavan tahon, kuten sertifikaatin myöntäjän (CA), allekirjoittamia, käytetään osapuolten aitouden todistamiseen. S7-1500 -ohjaimen sertifikaatin voi luoda ja asentaa ohjaimen turvallisuusasetuksiin, jolloin ohjain voi todentaa itsensä turvallisesti kommunikoidessaan muiden laitteiden kanssa ja päinvastoin. Itse allekirjoitetun

sertifikaatin määrityksissä on monta kenttää (Taulukko 2), joista ”Lifetime” ja ”SAN” ovat oleellisimpia. Jos logiikkaohjaimen IP-osoitetta muutetaan sertifikaatin luomisen jälkeen niin SAN-kentän informaatio ei pidä enää paikkaansa ja tällöin sertifikaatin tarjoama luottamus ei ole enää voimassa (Kuva 9).



Kuva 9. TIA Portal -ympäristössä luotu itse allekirjoitettu sertifikaatti, joka on mapattu S7-1500 CPU:n IP-osoitteisiin. Nämä näkyvät ”Subject Alternative Name” -kentässä sertifikaatissa.

Taulukko 2. Sertifikaatin olennaiset kentät (Hazem Sulaiman. 2022. 48).

Kentän nimi	**Mahdollinen arvo parametrille**
Type of certificate	Self-signed CA certicate
Common Name of Subject	
Encryption Method	RSA EC
Encryption Parameter	Key lenght for RSA secp256k1 secp384r1 prime256v1

Hash Algorithm	SHA1 SHA128 SHA384 SHA512
Lifetime - Validity	
Usage	OPC UA Client/Server TLS Client/Server WebServer
Subject Alternative Name (SAN)	URI IP DNS Email

7.4 Turvallisuushaasteet ja ratkaisut

Vaikka TLS tarjoaa vahvan suojan verkkoviestinnälle, on tärkeää ymmärtää ja hallita siihen liittyviä turvallisuushaasteita. Ohjausjärjestelmien ja OT-verkon muiden laitteiden välisen kommunikaation suojaamiseksi on varmistettava, että käytössä on ajantasaiset sertifikaatit ja TLS-versiot. Lisäksi on suositeltavaa käyttää vahvoja salausalgoritmeja ja säännöllisesti päivittää ohjelmistoja ja laitetason firmware-paketteja uusimpiin versioihin, jotka sisältävät viimeisimmät tietoturvapäivitykset.

TLS:n käyttö Siemens SIMATIC S7-1500 -ohjaimissa on esimerkki siitä, miten moderni tietoturva integroidaan teollisuusautomaatioon, suojaten kriittisiä prosesseja ja varmistaen turvallisen tiedonsiirron. Tämän kaltaiset turvatoimet ovat välttämättömiä teollisuusympäristöissä, joissa tietoturvariskit voivat johtaa merkittäviin taloudellisiin menetyksiin ja turvallisuusuhkiin.

7.5 TLS-sertifikaatin vanhentuminen

Kun TLS-sertifikaatti vanhenee Siemens SIMATIC S7-1500 -ohjaimen ja HMI tai SCADA -järjestelmän välillä, se voi johtaa useisiin ongelmiin ja haasteisiin kommunikaatiossa ja järjestelmän toiminnassa. TLS-sertifikaattien tarkoitus on varmistaa turvallinen ja luotettava yhteys kahden laitteen tai järjestelmän välillä,

ja niiden voimassaoloajan päättyminen voi aiheuttaa merkittäviä tietoturvariskejä ja operatiivisia häiriöitä.

Jos yhteydenmuodostus epäonnistuu, seurauksena on yhteyden katkeaminen. Vanhentunut sertifikaatti voi estää turvallisen yhteyden muodostamisen S7-1500 -ohjaimen ja HMI/SCADA-järjestelmän välillä. Useimmissa tapauksissa, kun HMI tai SCADA-järjestelmä yrittää muodostaa yhteyttä ohjaimeen, se tarkistaa sertifikaatin voimassaolon. Jos sertifikaatti on vanhentunut, yhteys hylätään turvallisuussyistä. Lisäksi S7-1500 -järjestelmä on suunniteltu ilmoittamaan turvallisuuteen liittyvistä ongelmista, kuten vanhentuneista sertifikaateista. Käyttäjille, järjestelmänvalvojille ja huoltohenkilöille näytetään varoituksia tai hälytyksiä, jotka kertovat sertifikaatin vanhenemisesta. Ongelmana on, että hälytys annetaan vasta kun sertifikaatti on jo vanhentunut, mikä vaatii välitöntä uusimista.

GSD Push -tekniikassa ennakointitoimintoon voi vaikuttaa asettamalla prosentuaalisen parametriarvon. Tällöin kun vanhenemassa oleva sertifikaatti on saavuttanut varoitusrajan niin tällöin aktivoituu järjestelmähälytys sekä ”maint” -valo ohjausjärjestelmässä, kunnes vanhenemassa oleva sertifikaatti on uusittu. GDS-toiminto ei toistaiseksi ole tuettuna tietoturvallisen PG/HMI-sertifikaattien kanssa (Siemens. 2022b. 345), mutta OPC OA -pohjaisten varmenteiden kanssa se on hyödyllinen.

7.6 Turvallisuusriskit

Vaikka yhteydenmuodostus estyy vanhentuneen sertifikaatin vuoksi, se itsessään on turvatoimi. Yleisesti vanhentuneita sertifikaatteja ei tule sivuuttaa tai kiertää, sillä ne voivat olla tietoturvariski ja voivat antaa hyökkääjille mahdollisuuden suorittaa "man-in-the-middle" -hyökkäyksiä, jos yhteyksiä ei varmisteta asianmukaisesti.

Operatiivisiin häiriöihin kuuluu tuotantoprosessien keskeytyminen. Kun S7-1500 -ohjaimen ja HMI/SCADA-järjestelmän välinen yhteys katkeaa, se voi keskeyttää teollisuusprosessien seurannan ja ohjauksen. Tämä puolestaan tyypillisesti johtaa tuotannon viivästyksiin, tehokkuuden laskuun ja jopa laitteistovaurioihin pahimmillaan. Manuaalinen puuttuminen tarvitaan vanhentuneen sertifikaatin aiheuttaman yhteysongelman korjaamiseen. Järjestelmänvalvojan, kunnossapidon edustajan tai teknisen tuen on uusittava tai vaihdettava vanhentunut sertifikaatti ja varmistettava, että kaikki järjestelmät luottavat uuteen sertifikaattiin, mikä voi olla aikaa vievää.

7.7 Ratkaisut ja ennaltaehkäisy

Sertifikaattien seuranta on prosessi, jossa aktiivisesti tarkkaillaan sertifikaattien voimassaoloaikoja ja huolehditaan niiden uusimisesta ennen niiden vanhenemista. Prosessi voidaan toteuttaa esimerkiksi käyttäen sertifikaattien hallintaan suunniteltuja työkaluja tai kehittämällä automatisoituja skriptejä, jotka varoittavat käyttäjää lähestyvistä vanhentumisajoista.

Toisena ratkaisuna on automaattinen uusiminen, joka on käytössä tietyissä järjestelmissä. Tällöin sertifikaatit uudistetaan ilman manuaalisia toimenpiteitä vähentäen inhimillisen virheen mahdollisuutta ja samalla lisäten järjestelmän luotettavuutta.

Yhteenvetona todettakoon, että vanhentuneen TLS-sertifikaatin hallinta on kriittinen osa teollisuusautomaatiojärjestelmän tietoturvaa, ja se vaatii jatkuvaa huomiota ja ylläpitoa varmistaakseen, että S7-1500 -ohjaimen ja HMI/SCADA-järjestelmän välinen viestintä pysyy turvallisena ja luotettavana.

8 WinCC

Siemensin WinCC brändi sisältää useita HMI- ja SCADA-järjestelmiä ja on yksi johtavista ja suosituimmista teollisuusautomaation ohjelmistoista maailmassa (Yu Jordan Zhao. 2023).

Kaikki WinCC -tuotteet sisältävät toimintoja kuten reaaliaikaisen prosessidatan näyttämisen, tiedon analysoinnin, hälytysjärjestelmän, raportointimahdollisuudet, käyttäjähallinnan sekä historian tallennuksen tietokantaan. Laajemmissa WinCC -tuotteissa on mukana palvelimien redundanssi, asiakas/palvelin -toiminnot sekä lukuisia muita optioita.

Tässä opinnäytetyössä tarkastellaan neljää keskeistä WinCC-tuotetta ja niiden käyttötarkoituksia ja eroja toisiinsa: WinCC 7/8, TIA Portal WinCC Runtime Professional, TIA Portal WinCC Unified ja WinCC Open Architecture (Kuva 10).

8.1 WinCC 7 ja 8

WinCC 7 ja WinCC 8 ovat vakiintuneita ja silti moderneja valvomo- ja visualisointiohjelmistoja. Niiden vahvuus piilee laajassa yhteensopivuudessa Siemensin vanhempien ohjausjärjestelmien kanssa sekä tuessa kolmannen osapuolten laitteille ja protokollille mahdollistaen integraatiot erilaisiin järjestelmiin. WinCC 7 ja WinCC 8 ovat keskeinen osa Siemens SIMATIC HMI -tuoteperhettä ja kyseinen tuoteperhe on ollut olemassa 90-luvun puolelta lähtien erottuen muista WinCC-tuotteista pitkällä markkinoilla olollaan ja laajalla käyttöönotollaan teollisuudessa. Mainittakoon, että WinCC 7/8-ohjelmistokomponenttia käytetään hyväksi myös PCS7-prosessiautomaatiojärjestelmässä. S7-1500:n TLS on tuettuna versiosta WinCC 7.5 SP2 update4 lähtien (Siemens. 2023a. 36).

8.2 TIA Portal WinCC Runtime Professional

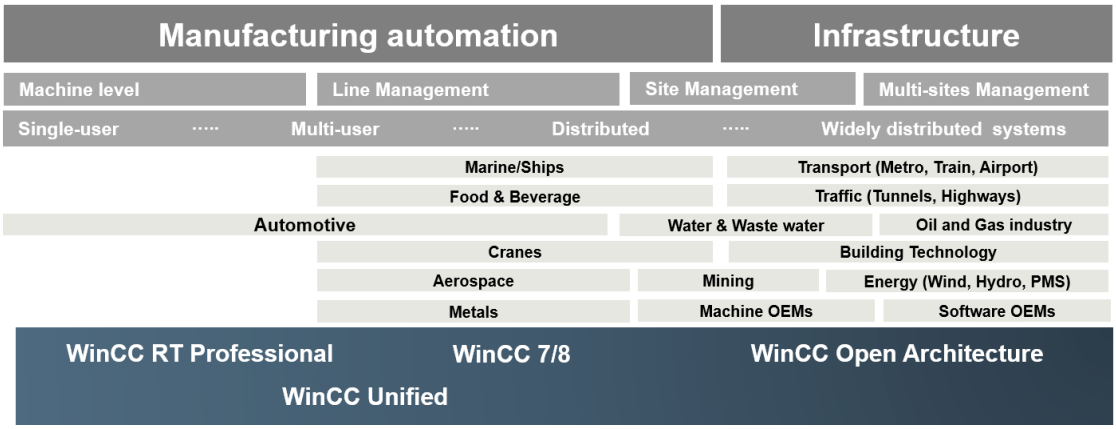
TIA Portal WinCC RT Runtime Professional on suunniteltu toimimaan osana Siemensin TIA Portal -kehitysympäristöä, tarjoten yhteisen projektitietokannan eri Simatic S7 ohjainperheiden kanssa. Tämän tuotteen erityisvahvuus on sen kyvyssä yhdistää PLC-ohjelmointi, HMI-suunnittelu ja muut automaation kehitystehtävät yhdeksi integroiduksi kokonaisuudeksi. Tuotteen ajonaikaisen osuuden toiminallisuus pohjautuu WinCC 7/8-versioon sisältäen muun muassa samanarvoiset skriptausominaisuudet sekä tietokantahistorian (Siemens. 2020b).

8.3 TIA Portal WinCC Unified

TIA Portal WinCC Unified on täysin uusi visualisointijärjestelmä, joka skaalautuu pienistä käyttäjäpaneelista aina SCADA-järjestelmään asti. Vaikka sillä ei vielä ole samanlaista toiminnallisuutta kuin olemassa olevilla muilla SCADA-järjestelmillä, sen kehitys on nopeassa edistymisvaiheessa. Valinta WinCC Unifiedin ja WinCC Professionalin välillä riippuu sovelluksesta, ja tehdasautomaatiossa Siemens suosittelee WinCC Unified -alustaa. WinCC Unifiedin kohokohta on sen moderni web-teknologiaan perustuva HTML5-käyttöliittymä, joka mahdollistaa modernin selainpohjaisen käyttäjäkokemuksen (Siemens. 2020a).

8.4 WinCC Open Architecture

WinCC OA:n merkittävin ero muihin WinCC-tuotteisiin on sen avoimen arkkitehtuurin ansiosta saavutettava mukautuvuus ja laajennettavuus. Se on suunniteltu alusta alkaen tukemaan suuria ja räätälöityjä projekteja, joissa vaaditaan erityisiä toiminnallisuuksia ja integraatioita. Tyypillistä käyttökohteita ovat infrastruktuuriltaan laajat projektit kuten muun muassa moottoritiet ja tunnelisovellukset (Siemens. 2022c. 4)



Kuva 10. WinCC -tuotteiden positiointi eri teollisuusautomaation sekä infrasektorien välillä (Siemens. 2022c. 4).

9 Syslog

Syslog on laajalti käytetty standardi viestien lokiin kirjaamiseen erityyppisistä järjestelmistä, laitteista ja sovelluksista. Sen alkuperä juontaa juurensa UNIX-järjestelmiin, mutta nykyään Syslogia käytetään monenlaisissa ympäristöissä ja alustoilla, mukaan lukien teollisen automaation järjestelmät. Syslogin avulla voidaan keskitetysti kerätä, tallentaa, analysoida ja hallinnoida lokitietoja, mikä on olennaisen tärkeää järjestelmien ja verkkoinfrastruktuurin vianmäärityksessä, suorituskyvyn seurannassa ja turvallisuuden valvonnassa (Arfan Sharif. 2023).

9.1 Syslogin periaate

Syslog perustuu yksinkertaiseen arkkitehtuuriin, jossa lähettävä taho (tuottaja) lähettää lokiviestejä Syslog-palvelimelle (kerääjä). Lokiviestit sisältävät yleensä tietoa tapahtumista, kuten järjestelmän varoitukset, virheet ja muut operatiiviset tiedot. Viestit luokitellaan tärkeysasteen mukaan, joka vaihtelee debug-tason viesteistä kriittisiin hälytyksiin. Tämä mahdollistaa järjestelmänvalvojien ja automaatioasiantuntijoiden keskittyä relevanteimpiin tapahtumiin ja nopeuttaa vianetsintää (Arfan Sharif. 2023).

9.2 Syslogin merkitys automaatiossa

Teollisen automaation järjestelmissä Syslog toimii työkaluna järjestelmän kunnon ja prosessien seurannassa. Syslog tallentaa tietoja laitteiden toiminnasta, kommunikaatiovirheistä ja prosessien poikkeavuuksista, mikä on tärkeää laitteiston ja ohjelmistojen sujuvan toiminnan varmistamiseksi. Syslogin avulla voidaan seurata ja analysoida prosessien suorituskykyä sekä havaita ja diagnosoida vikatilanteita, jolloin varhaisten varoitusten ja virheilmoitusten seuranta auttaa tunnistamaan potentiaaliset ongelmat ennen niiden kriittiseksi muuttumista.

Lisäksi Syslogin avulla voidaan tarkkailla tietoturvallisuuteen liittyviä tapahtumia, kuten luvattomien käyttöyritysten määrää, jolloin sen käyttäminen auttaa tunnistamaan turvallisuusuhkia ja helpottaa toteuttamaan korjaavia toimenpiteitä, mikä tärkeätä huomioida koska nykyään monet toimialat edellyttävät tiettyjen tapahtumien kirjaamista ja säilyttämistä osana sääntelyvaatimuksia.

Syslog on hyvä työkalu teollisuuden automaatiojärjestelmien ja IT-infrastruktuurin hallinnassa. Sen standardoitu formaatti ja laaja tuki eri alustoilla ja laitteissa tekevät siitä arvokkaan resurssin järjestelmien seurantaan, vianmääritykseen ja turvallisuuden valvontaan.

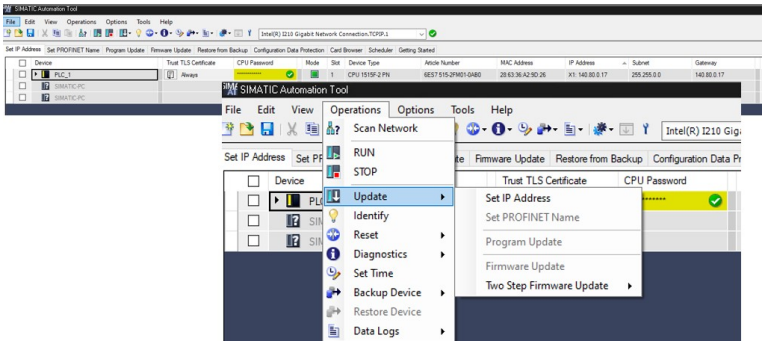
Tässä opinnäytetyössä luodun sovelluksen generoima data olisi suositeltavaa tallentaa Syslog- tai SIEM-järjestelmään, mikä olisi potentiaalinen idea applikaation jatkokehitykselle. Lisäksi on suositeltavaa aktivoida S7-1500 CPU:n ominaisuuksista toiminto, jonka avulla käyttäjä- ja tietoturvaan liittyvät tapahtumat lähetetään automaattisesti suoraan Syslog-palvelimelle.

10 Simatic Automation Tool SDK

Tässä opinnäytetyössä tutustuttiin syvällisesti Siemens Simatic Automation Tool (SAT) Software Development Kitin (SDK) ominaisuuksiin, toimintaperiaatteisiin ja mahdollisiin käyttökohteisiin. Simatic Automation Tool on Siemens AG:n kehittämä ohjelmisto, joka mahdollistaa automaatiolaitteiden, kuten ohjelmoitavien logiikkaohjainten (PLC), HMI-paneelien ja muiden komponenttien konfiguroinnin, päivityksen, diagnostiikan ja hallinnan. Simatic Automation Toolin avulla käyttäjät voivat suorittaa nämä toimenpiteet ilman syvällistä ohjelmointitaitoa ja SAT:n käyttöliittymä on muutoinkin suunniteltu intuitiiviseksi (Kuva 11), jotta käyttäjät voivat nopeasti suorittaa tarvittavat huoltotoimenpiteet (Siemens. 2024b. 9).

Yleisesti käytetty käyttötarkoitus SAT:n avulla on OT-verkon skannaus, jolloin saadaan listaus käytetyistä Simatic automaatiokomponenteista ja niiden versioista firmware-tason ja nämä puolestaan voidaan tallentaa muun muassa SBIM-listana SIEM-järjestelmään. Muita tyypillisiä toimintoja ovat IP-osoitteiden määrittäminen, laiteohjelmiston päivitykset ja yhteyksien testaus.

Ohjelmasta on olemassa myös Linux -ympäristöön tarkoitettu variaatio nimeltään ”Simatic Transfer Tool” (Siemens. 2021). Sen avulla muun muassa CI/CD-kaltaisen mekanismin implementointi olisi mahdollista S7-1500:n yhteydessä. Simatic Transfer Tool-työkalu lisäksi mahdollistaa logiikan ja operointipaneelien sovellusohjelman hallitun päivityksen muun muassa bash -komentotulkin kautta.



Kuva 11.Simatic Automation Tool-ohjelman graafinen käyttöliittymä.

10.1 Simatic Automation Tool SDK:n rooli ja ominaisuudet

Vaikka Simatic Automation Tool itsessään tarjoaa laajan valikoiman toimintoja, sen SDK (Software Development Kit) -osuus laajentaa näitä mahdollisuuksia tarjoamalla ohjelmointirajapinnan (C# API), joka mahdollistaa ohjelmistokehittäjien luoda räätälöityjä ja kustomoituja sovelluksia. SDK:n avulla voidaan toteuttaa esimerkiksi:

- Automatisoida toistuvia tehtäviä ja prosesseja, kuten varmuuskopiointi.
- Integroida Simatic Automation Tool muihin järjestelmiin, kuten tietokantoihin tai yrityksen resurssienhallintajärjestelmiin.
- Kehittää erikoistuneita diagnostiikka- ja valvontatyökaluja.

SAT SDK sisältää kirjastoja, dokumentaatiota ja kehitystyökaluja mahdollistaen laajan yhteensopivuuden ja joustavuuden sovelluskehityksessä. Alustana voi olla Windows tai Linux.

Tällä hetkellä Siemens Simatic Automation Tool SDK ei ole yleisesti tunnettu tai erikseen dokumentoitu tuote Siemensin virallisissa resursseissa samalla tavalla kuin TIA Portal Openness -ohjelmointirajapinta, mutta mielestäni etenkin kyseisen paketin SDK osuus mahdollistaa täysin uusia prosesseja ja lähestymistapoja IT/OT sovelluskehittäjille ja automaation kunnossapidon ylläpidolle. Tässä opinnäytetyössä hyödynnettiin SDK-paketin tarjoamia rajapintoja selvittämään S7-1500:n kommunikaatiosertifikaatin elinkaaren voimassaoloajan aikaikkunan. Simatic Automation Tool SDK ei sisällä graafista valmisohjelmaa vaan se osuus kuuluu osuuteen ilman SDK-kirjastoa.

SAT SDK:lla luotu räätälöity sovellus ei tarvitse erikseen kaupallista lisenssiä ja sitä voidaan vapaasti jakaa eteenpäin.

10.2 Simatic Automation Tool perustoiminnot

Opinnäytetyössä käytetyn SAT SDK-ohjelmiston versio oli V5.0 ja Windows-variaatio ohjelmistosta on asennettavissa Windows10 ja Windows11 (64-bit) ja

näiden Windows Server puolen vastaaviin versioihin. SDK-version suhteen Windowsin lisäksi Linux -käyttöjärjestelmissä on tuettuna Debian 11, Ubuntu 20, 22 sekä SIMATIC Industrial OS -ympäristöt (Siemens. 2023d. 4).

Ohjelmiston tilauskoodi on 6ES78531AE050AG8 ja siitä on saatavilla myös Trial -versio kokeilukäyttöön ja sisältää oheisia keskeisiä toimintoja (Siemens. 2023c).

- Varmuuskopiointi ja palautus S7-CPU:lle
- Muistin nollaus
- Palautus tehdasasetuksiin
- Muistikortin alustus
- Ajan asettaminen
- Tietojen lokien lukeminen ja poistaminen
- Muistikorttien luominen
- Suodatettu skannaus
- Nopea ping

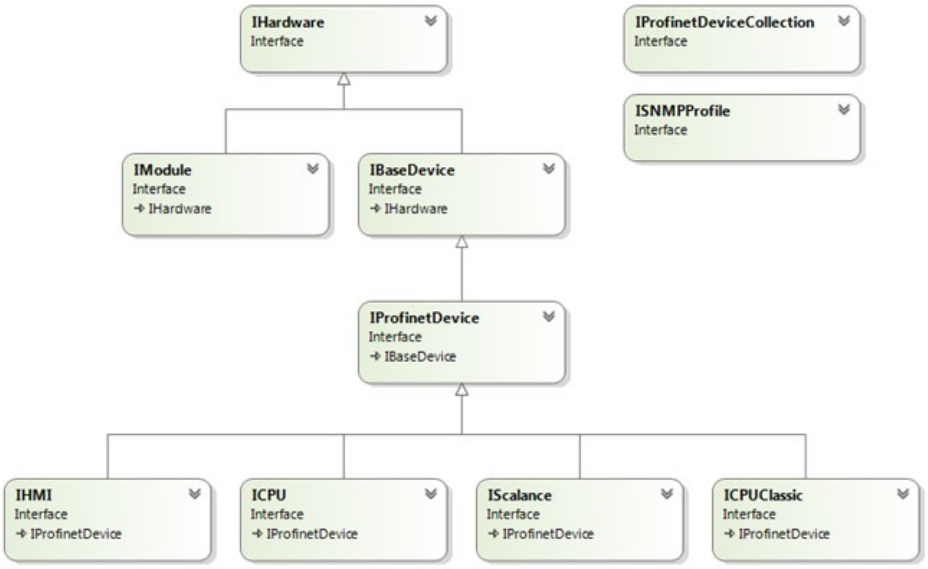
10.3 SAT SDK API -luokat.

Simatic Automation Tool SDK:n sisältämä API tarjoaa luokkia, rajapintoja ja metodeja PROFINET-verkon SIMATIC-laitteiden kanssa kommunikointia varten ja sen sisältämä NET-luokka ”Network” edustaa PROFINET-verkkoa kokonaisuudessaan (Kuva 12). Tämä luokka suorittaa toimintoja käyttäen ohjelmointilaitteeseen asennettua verkkokorttia (NIC). Network-luokkaa käytetään saatavilla olevien verkkoliityntäkorttien etsimiseen ja PROFINET-verkkoon liitetyn verkkoliitynnän valitsemiseen.

PROFINET-verkon yksittäiset laitteet edustetaan rajapinnoilla. Kukin rajapintaluokka tarjoaa ominaisuuksia ja metodeja edustamalleen verkkolaitteelle. Kukin verkon laitteisto edustetaan parhaiten seuraavilla rajapinnoilla:

- IProfinetDevice – Kaikki PROFINET-verkossa suoraan saavutettavissa olevat laitteet voidaan edustaa tällä rajapinnalla, koska kaikki laitteet ovat periytettyjä tästä luokasta.
- ICPU edustaa suoraan verkkoon kytkettyjä S7-1X00 CPU:ita. CPU:ille on tuettu erityistoiminnallisuutta.
- ICPUClassic edustaa klassisia S7-300 ja S7-400 CPU:ita, jotka ovat suoraan kytkettyjä verkkoon.
- IHMI edustaa suoraan verkkoon kytkettyjä SIMATIC HMI:itä. HMI:ille on tuettu erityistoiminnallisuutta.
- IBaseDevice -rajapintaa käytetään edustamaan laitteita, jotka eivät ole suoraan kytkettyjä PROFINET-verkkoon, mutta ovat saavutettavissa toisen laitteen kautta. Esimerkiksi PROFIBUS-orjasema, joka on kytketty verkon prosessiasemaan, edustetaan IBaseDevice-rajapintana.
- IModule -rajapintaa käytetään yksittäisten I/O-moduulien edustamiseen, jotka on kytketty prosessiasemaan, PROFINET-laitteeseen tai PROFIBUS-asemaan.
- IHardware on kaikkien muiden rajapintojen perusluokka. Tämä rajapinta tarjoaa pääsyn ominaisuuksiin, jotka ovat yhteisiä kaikille verkon tunnistamille laitteille.
- IScalance -rajapinta, joka edustaa SCALANCE-laitteita. Rajapinnat on ryhmitelty kokoelmiin, jotka edustavat laitteiden ryhmiä. Kokoelmat on tarjottu tukemaan iteraatiota, suodatusta ja etsintää.
- IProfinetDeviceCollection on kokoelma kaikista verkossa suoraan saavutettavista laitteista.
- IModuleCollection on kokoelma, joka voi edustaa CPU:seen tai IM:ään kytkettyjä moduuleja.
- IHardwareCollection kokoelma edustaa CPU:ta ja kaikkia sen moduuleja.
- IScanErrorCollection kokoelma edustaa kaikkien laitteiden verkkoskannauksen operaatiosta palautettujen virheiden joukkoa.

(Siemens. 2023e. 39)



Kuva 12. Luokkakaavio näyttää perimissuhteen rajapintaluokkien välillä (Siemens. 2023e. 40).

10.4 Päätelmät ja tulevaisuuden näkymät

Simatic Automation Tool SDK tarjoaa merkittäviä ja mielenkiintoisia mahdollisuuksia automaatiojärjestelmien kunnossapidon tarpeiden räätälöintiin ja integraatioon, mikä oikein käytettynä luonnollisesti parantaa teollisuuden prosessien tehokkuutta, luotettavuutta ja joustavuutta. Tulevaisuudessa SDK:n rooli voi kasvaa entisestään, kun digitalisaatio syventävät ohjelmistojen ja automaation integraatiota.

EU:n direktiivit NIS2 ja CRA johtavat siihen, että OT-verkon komponenteista on ylläpidettävä laitelistaa, jolloin mahdollisten haavoittuvuuksien paikkaaminen on helpompaa ja suoraviivaisempaa ja tämä haaste puolestaan onnistuu osittain SAT-ohjelman avulla.

11 Microsoft .Net 6.0

.NET 6, joka yleisesti tunnetaan vain .NET 6:na (ilman "Framework" tai "Core" lisänimeä), on Microsoftin kehittämä ohjelmistokehitysalusta. Se on osa .NET:n uudempaa sukupolvea, joka yhdistää .NET Coren, .NET Frameworkin ja Mono/Xamarinin parhaat puolet yhdeksi yhtenäiseksi alustaksi. .NET 6 julkaistiin marraskuussa 2021, ja se on ensimmäinen LTS (Long Term Support) versio uuden yhtenäisen .NET-alustan puitteissa, mikä tarkoittaa, että Microsoft tukee sitä usean vuoden ajan (Microsoft .NET Support Policy. 2024).

.NET 6:n tärkein ominaisuus on sen yleiskäyttöisyys ja laaja-alainen tuki erilaisille ohjelmointitarpeille, mukaan lukien verkkosovellukset, mobiilisovellukset, pilvipalvelut, työpöytäsovellukset ja IoT (Internet of Things) -laitteet ja on suunniteltu toimimaan useilla eri alustoilla mukaan lukien Windows, Linux ja macOS. Usean käyttöjärjestelmän tuki antaa kehittäjille mahdollisuuden kohdistaa sovelluksensa monenlaisiin laitteisiin ja ekosysteemeihin.

.NET 6:n suunnittelussa on keskitytty suorituskyvyn ja skaalautuvuuden parantamiseen, mikä on etenkin pilvipohjaisten ja suurta datamäärää käsittelevien sovellusten kannalta edullista. Kehittäjille tarjotaan moderni kehityskokemus uusimpien ohjelmointimallien, kuten asynkronisen ohjelmoinnin ja tuen uusimpien ohjelmointikielten, kuten C# 10, kautta. Lisäksi .NET 6 hyötyy .NET:n laajasta kirjastosta ja sen ympärille rakentuneesta vahvasta ekosysteemistä, mikä avaa laajat kehitysmahdollisuudet. Koska .NET 6 on LTS (Long Term Support) -versio, se saa Microsoftilta virallista tukea ja päivityksiä useiden vuosien ajan, mikä tekee siitä luotettavan vaihtoehdon erityisesti yrityssovellusten kehittämiseen (Microsoft .NET Support Policy. 2024).

11.1 Kehitysympäristöt ja työkalut

.NET 6-sovellusten kehitykseen voidaan käyttää useita eri työkaluja, kuten Visual Studio 2022, Visual Studio Code ja JetBrains Rider (JetBrains. 2024).

Visual Studio 2022 on integroitu kehitysympäristö (IDE), joka tarjoaa laajan tuen .NET 6-sovellusten suunnittelulle, kehitykselle, testaukselle ja julkaisulle. Visual Studio Code on kevyempi editori, joka sopii erinomaisesti ristiin alustaiseen kehitykseen ja tukee laajasti .NET 6-kehitystä laajennusten kautta.

.NET 6 edustaa merkittävää askelta eteenpäin Microsoftin .NET-alustan kehityksessä, tarjoten kehittäjille yhtenäisen ja monipuolisen alustan erilaisten sovellusten kehittämiseen. Sen yleiskäyttöisyys, suorituskyky ja tuki ristiin alustaiselle kehitykselle tekevät siitä houkuttelevan vaihtoehdon monenlaisiin projekteihin, alkaen pilvipohjaisista palveluista mobiili- ja työpöytäsovelluksiin.

12 Sertifikaattien valvontasovellus

Opinnäytetyön tutkimusosio jakautui kollegoiden kanssa tapahtuneeseen kyselyyn, kirjattuihin asiakastukipyyntöihin liittyen muihin vastaaviin tilanteisin, joissa esimerkiksi OPC UA -kommunikaatioon liittyvä varmenne oli laitoksella vanhentunut ja aiheuttanut huomattavat tuotannolliset ja taloudelliset tappiot sekä tosiasiaan, että toistaiseksi ei ole saatavilla helppoa tapaa monitoroida dynaamisesti varmenteiden käyttöaikaa, joten tavoitteena oli löytää tähän uusi näkökulma.

Kollegoiden kyselyt tapahtuivat suullisesti puhelimessa sekä MS Teamsissa (Microsoft Teams. 2024) samalla kun asiaan viittaavista ongelmista keskusteltiin yleisellä tasolla. Haastateltavia oli kolme henkilöä. Palautteen pohjalta muodostui ideoita sekä toiminnallisia parannuksia kehitettyyn ohjelmistoon.

Kysymykset olivat seuraavia.

- Mikä on tärkein sertifikaatti, jota tulee valvoa?
- Mitä muuta ohjelman pitäisi monitoroida ja tallentaa ajan lisäksi?
- Millä tavoin ohjelman kannattaisi huomioida Access Level -suojaussalasanan käyttö.

Yhteneväisiä vastauksia olivat: PG/HMI-sertifikaatti sekä tieto siitä, jos sertifikaatti on muuttunut laitteessa.

Toivottu ohjelmiston lopputulos tulisi olla sellainen, että datatieto voidaan esittää valvomossa visuaalisesti operaattoreille ilman, että jokaiseen WinCC -variaatioon pitäisi erikseen tehdä maksullisilla ODK tai API-rajapinnoilla syvällisiä muokkauksia eli linkki tulisi olla yksinkertainen.

Kerätyn informaation pohjalta opinnäytetyössä luotiin Simatic Automation Tool SDK:n avulla ”PKIMonitor” -niminen applikaatio, joka toimii middleware-ohjelmana sijaiten ohjausjärjestelmän ja WinCC:n välillä.

”PKIMonitor” -ohjelmisto voidaan käynnistää komentoriviltä taikka suoritettava prosessina ilman käyttöliittymää joko manuaalisesti käynnistettynä taikka ajastetusti esimerkiksi kerran päivässä. Ohjelmiston suoritus voidaan myös tarvittaessa toteuttaa Windows-palveluna. Ohjelmisto voi sijaita samalla tietokoneella WinCC-valvomon kanssa taikka erillisellä serverillä, jolla on pääsy OT-verkossa sijaitseviin S7-1500-logiikkaohjaimiin. Ohjelmisto luotiin Visual Studio 2022 Community -versiolla käyttäen hyödyksi .Net 6.0-kirjastoa, SAT SDK:ta ja niiden sisältämiä luokkia.

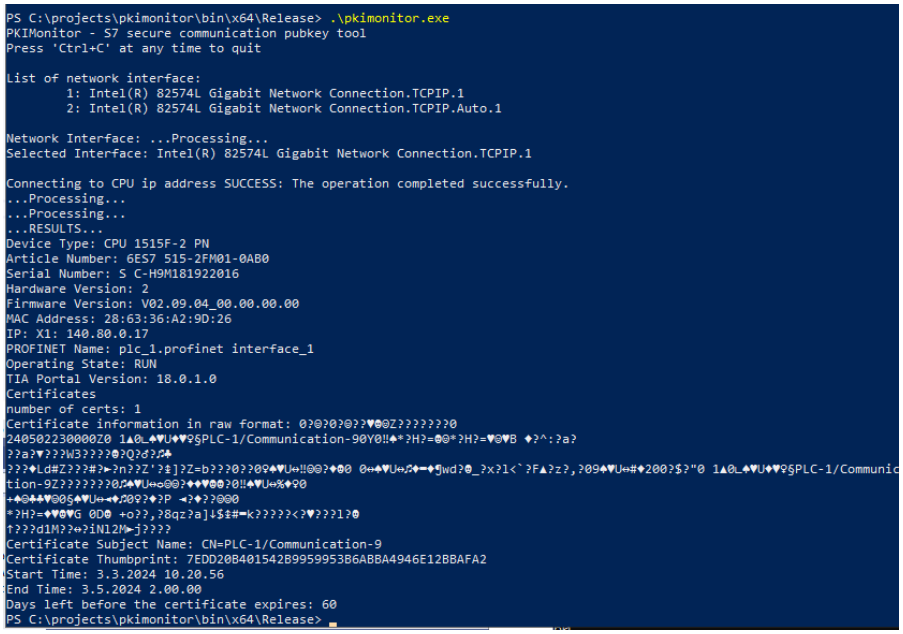
Taustana toiminnalle on vakioitu toimintaperiaate TIA Portal V17 -kehitysympäristön ja S7-1500 -ohjaimien firmware 2.9-versiosta lähtien, jossa TLS toimii osana S7-1500 -ohjaimen tietoturvaa. Tällöin lisätessä S7-1500 CPU-mallin TIA Portal Step7 Professional -projektiin niin samalla pakollisena toimenpiteenä on luoda PG/HMI-kommunikaatiolle digitaalinen sertifikaatti. Oletuksena sertifikaatin validi voimassaolon aikaikkuna on vuoteen 2037 alkaen hetkestä, jolloin se luodaan projektiin, mikä on tässä yhteydessä tietokoneen sen hetkinen kellonaika. Ajonaikaisesti HMI/WinCC:n kellonaika täytyy olla tämän aikaikkunan sisäpuolella ja näin ollen kellonaikojen tahdistus S7-CPU:n ja HMI/WinCC:n välillä on pakollinen tai ainakin hyvin suositeltu toimenpide kaikissa TLS-suojaukseen pohjautuvissa applikaatioissa. Kellonajan tahdistamiseen voidaan käyttää muun muassa NTP-tekniikkaa.

OPC UA ja WebServer (https) käyttävät eri sertifikaatteja PG/HMI ja niiden luominen on optionaalista.

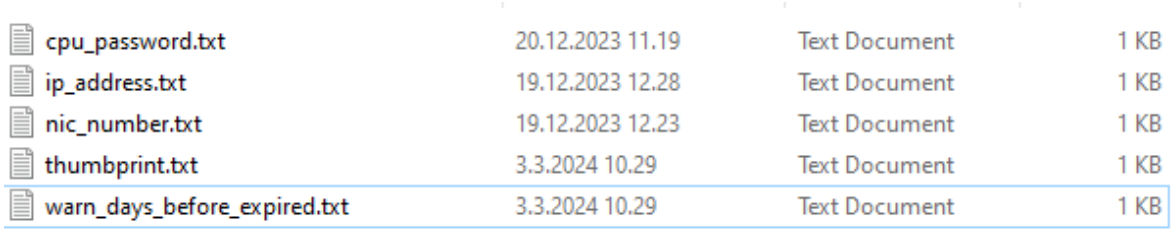
12.1 Valvontasovelluksen suoritus vaiheittain

PKIMonitor -ohjelmisto ilmoittaa (Kuva 13) onnistuneen yhteyden luonnin jälkeen S7-1500, S7-1200, ET200SP -ohjaimen tunnistetiedot. Ensimmäisessä sovelluksen koontiversiossa valtaosaa näistä ei tallenneta tiedostoon ja ne ovat vain informatiivisia tietoja.

- Laitetyyppi ja malli
- Tilauskoodi
- Yksilöllinen sarjanumero
- Hardwaren versio
- Firmwaren versio
- MAC-osoite
- IP-osoite
- Profinet-verkon osalta laitteen nimi
- Toimintatila, RUN/STOP
- Vanhin TIA Portal Step7 -versio, jolla logiikkaohjelmaa voidaan käsitellä
- Aktiivisten PG/HMI-kommunikaatiosertifikaattien määrä
- PG/HMI-kommunikaatiosertifikaatin sisältö kryptatussa formaatissa.
- PG/HMI-kommunikaatiosertifikaatin sisältämä "Subject Name" -kenttä
- PG/HMI-kommunikaatiosertifikaatin sisältämä "Thumbprint" -kenttä
- PG/HMI-kommunikaatiosertifikaatin sisältämä "Start Time" -kenttä
- PG/HMI-kommunikaatiosertifikaatin sisältämä "End Time" -kenttä
- Laskennallinen arvo, jossa "End Time" kentän arvosta on vähennetty "Start Time" -kentän arvo, jolloin saadaan selville, kuinka monta päivää on jäljellä siihen, että sertifikaatin voimassaoloaika on validi.



Kuva 13. PKIMonitor-ohjelman tuloste, kun yhteys on luotu onnistuneesti.



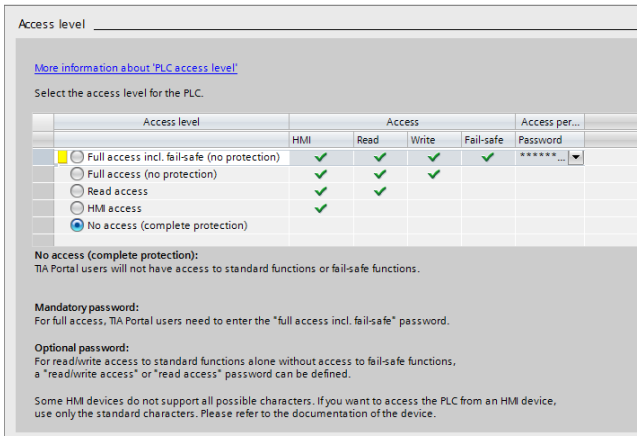
Kuva 14. PKIMonitor -sovelluksen kontekstitiedostot

Ohjelmiston (Kuva 14) käyttämät parametritiedostot (1) sekä S7 online analyysin perusteella luodut datatiedostot (2).

- cpu_password.txt (1)
- ip_address.txt (1)
- nic_number.txt (1)
- thumbprint.txt (2)
- warn_days_before_expired.txt (2)

”Cpu_password.txt” -tiedosto sisältää S7-1500 CPU:n ”Access Level” -salasanan (Kuva 15) kryptatussa muodossa. Salasana konfiguroidaan ja

määritellään TIA Portal Step7 Professional -ohjelmiston CPU:n suojausasetuksissa.



Kuva 15. Access Level -suojaustason määritykset.

PKIMonitor-ohjelmalle salasana syötetään komentoriviargumenttina, jolloin ohjelmisto käynnistyy erityistilassa ja tallentaa sen kryptatussa muodossa tiedostoon. Kun PKIMonitor-ohjelmisto suorittaa varsinaisen diagnostiikan niin tällöin ohjelmisto lukee salasanan tiedostosta, purkaa salauksen auki ja käyttää samaa salasanaa suojatun PG/HMI-yhteyden muodostamiseen (Kuva 18 ja 19).

”ip_address.txt” -tiedosto sisältää S7-1500 CPU:n ip-osoitteen. Tämä parametri määritellään etukäteen.

”nic_number.txt” -tiedosto indeksinumeron verkkokorteista, joita voidaan käyttää yhteyden muodostamiseen. TCPIP.1 on hyvin suositeltava valinta tässä yhteydessä. Tämä parametri määritellään etukäteen.

”thumbprint.txt” -tiedosto (Kuva 16) sisältää HMI/PG-sertifikaatin fingerprint -osuuden. Tämä on hyvin hyödyllinen informaatio, joka kertoo muun muassa onko laitteen sovellusprojekti päivittynyt toisella versiolla, jolloin myös sertifikaatti on muuttunut ja tästä voidaan päätellä, että ulkopuolinen taho on muuttanut sovellusta.

```
PS C:\projects\pkimonitor\bin\x64\Release\options> cat .\thumbprint.txt
7EDD20B401542B9959953B6ABBA4946E12BBAFA2
```

Kuva 16. ” thumbprint.txt” -tiedoston sisältämä sertifikaatin thumbprint -kenttä.

”warn_days_before_expired.txt” -tiedosto (Kuva 17) sisältää päivien määrän ennen kuin sertifikaatti on vanhentunut. Tämä informaatio voidaan lukea eri WinCC SCADA -sovelluksiin niiden omien skriptausmoottorien avulla.

```
PS C:\projects\pkimonitor\bin\x64\Release\options> cat .\warn_days_before_expired.txt
60
PS C:\projects\pkimonitor\bin\x64\Release\options>
```

Kuva 17. Näkymä lopputuloksesta.

```
public static string EncryptToBase64(string originalText)
{
    var userBytes = Encoding.UTF8.GetBytes(originalText); // UTF8 saves Space
    var userHash = MD5.Create().ComputeHash(userBytes);
    SymmetricAlgorithm crypt = Aes.Create(); // (Default: AES-CCM (Counter with CBC-MAC))
    crypt.Key = MD5.Create().ComputeHash(Encoding.UTF8.GetBytes("MySecretHashPassWord")); // MD5: 128 Bit Hash
    crypt.IV = new byte[16]; // by Default. IV[] to 0.. is OK simple crypt
    using var memoryStream = new MemoryStream();
    using var cryptoStream = new CryptoStream(memoryStream, crypt.CreateEncryptor(), CryptoStreamMode.Write);
    cryptoStream.Write(userBytes, 0, userBytes.Length); // User Data
    cryptoStream.Write(userHash, 0, userHash.Length); // Add HASH
    cryptoStream.FlushFinalBlock();
    var resultString = Convert.ToBase64String(memoryStream.ToArray());
    return resultString;
}
```

Kuva 18. Funktio, jota käytetään kryptaamaan Access Level -salasana. Kuvassa oleva hash-salausavain ”MySecretHashPassWord” on esimerkki eikä ole sama lopullisessa versiossa.

```
public string DecryptFromBase64(string encryptedText)
{
    var encryptedBytes = Convert.FromBase64String(encryptedText);
    SymmetricAlgorithm crypt = Aes.Create();
    crypt.Key = MD5.Create().ComputeHash(Encoding.UTF8.GetBytes("MySecretHashPassWord"));
    crypt.IV = new byte[16];
    using var memoryStream = new MemoryStream();
    using var cryptoStream = new CryptoStream(memoryStream, crypt.CreateDecryptor(), CryptoStreamMode.Write);
    cryptoStream.Write(encryptedBytes, 0, encryptedBytes.Length);
    cryptoStream.FlushFinalBlock();
    var allBytes = memoryStream.ToArray();
    var userLen = allBytes.Length - 16;
    if (userLen < 0) throw new Exception("Invalid Len");   // No Hash?
    var userHash = new byte[16];
    Array.Copy(allBytes, userLen, userHash, 0, 16); // Get the 2 Hashes
    var decryptHash = MD5.Create().ComputeHash(allBytes, 0, userLen);
    if (userHash.SequenceEqual(decryptHash) == false) throw new Exception("Invalid Hash");
    var resultString = Encoding.UTF8.GetString(allBytes, 0, userLen);
    return resultString;
}
```

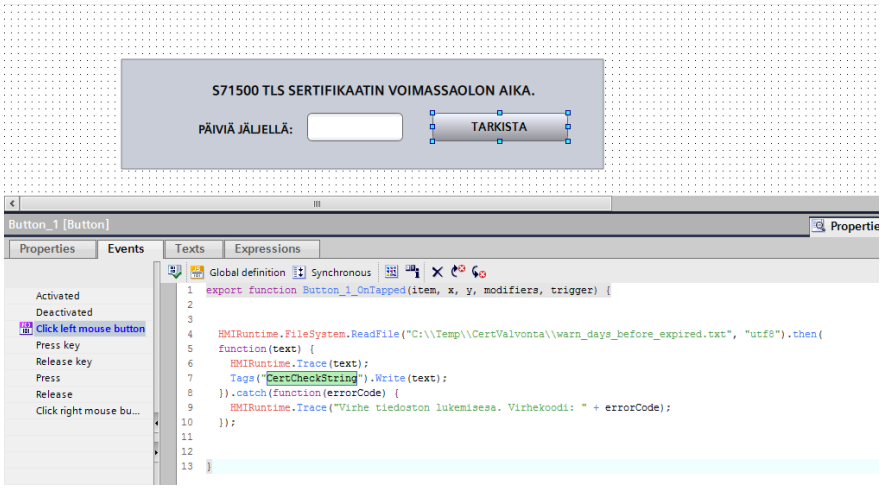
Kuva 19. Funktio, jota käytetään purkamaan Access Level -salasana. Kuvassa oleva hash-salausavain ”MySecretHashPassWord” on esimerkki eikä ole sama lopullisessa versiossa.

13 Sovellusesimerkit eri WinCC -variaatiolle

Jokainen WinCC -tuote poikkeaa toisistaan, mutta kaikille on yhteistä, että niissä on mahdollisuus skriptien avulla lukea paikallisia teksti- ja binääritiedostoja. Tätä ominaisuuksista hyödynnettiin luettaessa datatiedoston sisältöä ja tallennettaessa se WinCC projektin hmi tag -muuttujatietokantaan. WinCC Unified:ssa skriptauskielenä toimii javascript. WinCC 7 ja 8:ssa on mahdollisuus käyttää VBS- ja C-skriptiä ja WinCC OA:ssa skriptauskielenä toimii ”CTRL script”, joka noudattaa C-kielen syntaksia.

13.1 TIA Portal WinCC Unified

Kuva 20 sisältää WinCC Unifiedin implementaatioesimerkin siitä, kuinka javascriptin avulla sertifikaatin ”päiviä jäljellä” -tiedon voi lukea PKIMonitor -ohjelman luomasta datatiedostosta ja kuinka se tallennetaan WString -datatyyppiä olevaan HMI Tag -muuttujaan. Esimerkissä on luotu kuvasivulle button-objekti, jonka ”click left mouse button” -tapahtumassa suoritetaan varsinainen scripti.



Kuva 20. WinCC Unified implementaatio.

13.2 WinCC 7/8 ja TIA Portal WinCC Runtime Professional

Kuva 21 sisältää mallitoteutuksen siitä, kuinka VBS:n avulla voidaan lukea tiedoston sisältö järjestelmään ja tallentaa se HMI Tag -muuttujatietokantaan myöhempää visualisointia varten.

```
Sub OnClick(ByVal Item)
    On Error Resume Next
    Dim objFSO, objTextFile
    Dim strFilePath, strLine
    ' Polku tekstitiedostoon
    strFilePath = "c:\temp\warn_days_before_expired.txt"

    ' Luo FileSystemObject-olio
    Set objFSO = CreateObject("Scripting.FileSystemObject")

    ' Tarkista, löytyykö tiedosto
    If objFSO.FileExists(strFilePath) Then
        ' Avaa tiedosto lukemista varten
        Set objTextFile = objFSO.OpenTextFile(strFilePath, 1)
         'Lue tiedosto
        strLine = objTextFile.ReadLine
        'Näytä informaatio debug -viestiruudussa
        HMIRuntime.Trace "Käyttöaikaa jäljellä päiviä: " & strLine
        'kopioi informaatio valvomon hmi tag tietokantamuuttujaan
        HMIRuntime.Tags("warn_days_before_expired").Write strLine
        ' Sulje tiedosto
        objTextFile.Close
    Else
        HMIRuntime.Trace  "Tiedostoa ei löytynyt: " & strFilePath
    End If
    ' Siivotaann muuttujat
    Set objTextFile = Nothing
    Set objFSO = Nothing

    'Virhetilanteessa tulostetaan debug-objektiin tapahtuman kuvaus
    If err.number <> 0 Then
        HMIRuntime.Trace  "Virheen kuvaus: " & err.description
    End If

End Sub
```

Näkymä ajonaikaisesta WinCC-kuvasivusta.

HAE SERTIFIKAATIN KÄYTTÖAIKA PÄIVISSÄ

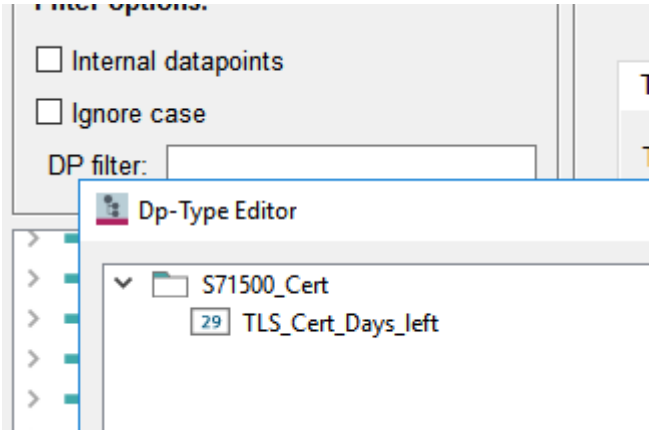
60

Käyttöaikaa jäljellä päiviä: 60

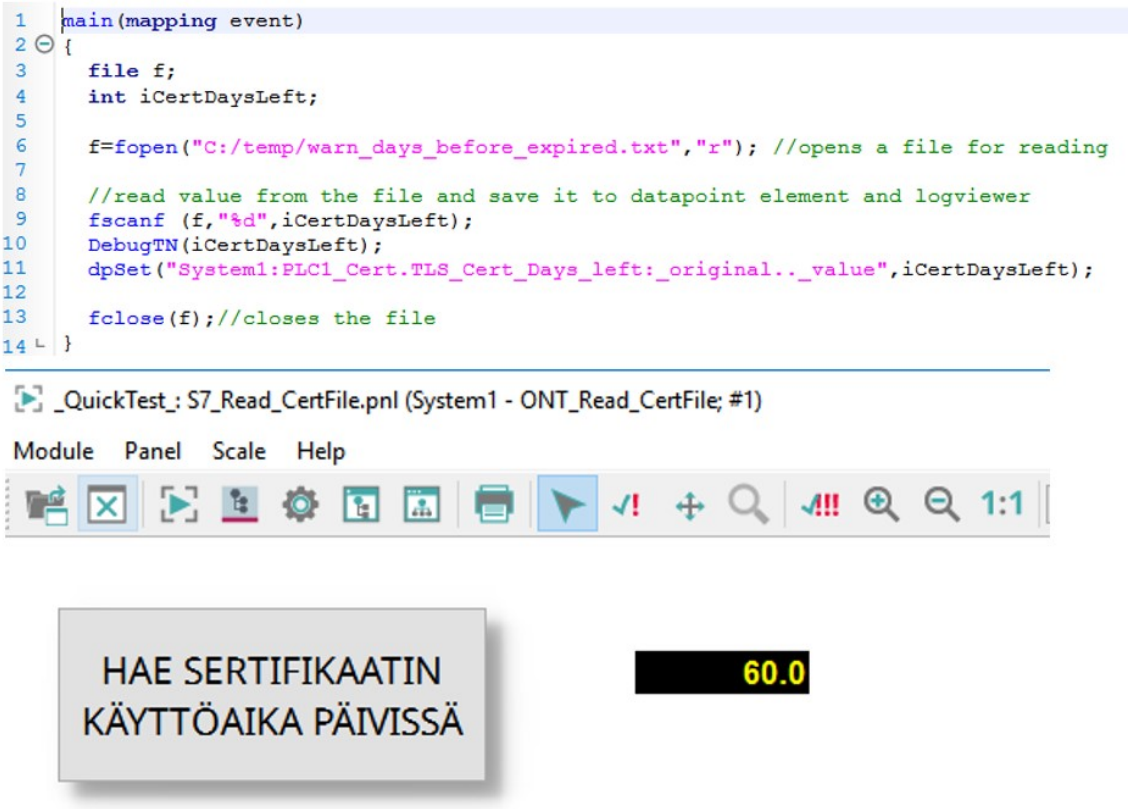
Kuva 21. VBS implementaatio.

13.3 WinCC OA

WinCC OA:ssa projektissa tulee luoda DPT-luokka ja sille DPe-instanssi sisältäen nimen ja string-tyyppisen elementin (Kuva 22) käyttäen Para-editoria. Seuraavaksi CTRL:n (Control Script) avulla voidaan esimerkiksi button-objektin click -tapahtussa lukea datatiedoston sisältö (Kuva 23).



Kuva 22. Para -editori.



Kuva 23. WinCC OA implementaatio.

14 Muita mahdollisia keinoja selvittää S7-1500:n kommunikaatiosertifikaatin elinaika

Ensisijainen vaihtoehto tyypillisesti on tehdä tarkistus offline-tilassa avaamalla TIA Portal -ympäristössä S7-1500- tai S7-1200-sovellusprojekti ja käsin manuaalisesti katsoa PG/HMI-kommunikaatiosertifikaatin tiedot. Kuitenkin dynaaminen tapa usein olisi suotavaa ja ohessa pohdiskelun ja tutkimustyön pohjalta muutamia tapoja sen selvittämiseksi ilman TIA Portal sovellusprojektia.

14.1 Windows CertStore

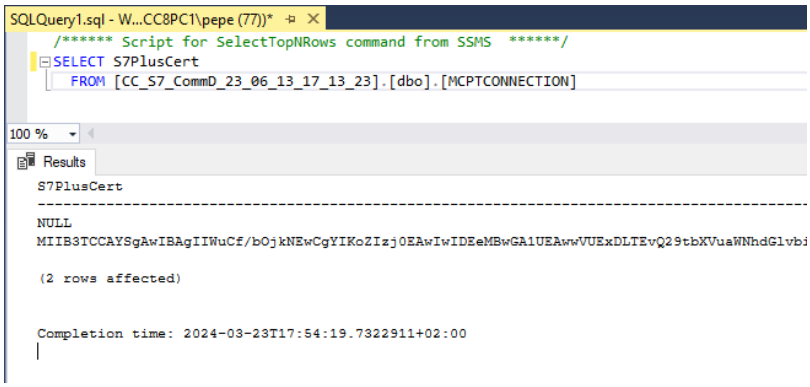
Sertifikaattien hallinnassa ja niiden voimassaoloaikojen seurannassa Windows-ympäristössä on olennaista ymmärtää, miten CertStore toimii ja miten tiedostopohjaisia sertifikaatteja käsitellään. Windowsin CertStore tarjoaa keskitetyn hallinnan järjestelmän sertifikaateille, jota voidaan tutkia Microsoft Management Console (MMC) avulla käyttämällä Sertifikaatit-snap-in -toimintoa. Tämä mahdollistaa sertifikaattien yksityiskohtaisten tietojen, kuten voimassaoloaikojen ja myöntäjätietojen, tarkastelun (Microsoft MMC. 2024).

Komentorivityökalut, kuten ”certutil”, ovat myös tärkeitä sertifikaattien hallinnassa, sillä niiden avulla voidaan nopeasti tarkistaa sertifikaattien voimassaoloajat ja muut kriittiset tiedot suoraan CertStoresta. Tiedostopohjaisten sertifikaattien, esimerkiksi .pfx tai .cer tiedostomuodoissa, osalta voimassaoloajan selvittäminen edellyttää tiedoston metatiedon tutkimista, josta voidaan päätellä sertifikaatin myöntäjä, voimassaoloaika ja muut keskeiset tiedot.

Hallintaan kuuluu myös automatisointityökalujen ja skriptien hyödyntäminen, jotka auttavat ylläpitämään sertifikaattien luotettavuutta ja turvallisuutta, erityisesti suurissa ympäristöissä, missä sertifikaatteja on paljon. Tämä lähestymistapa on oleellinen ylläpidettäessä tietoturvan korkeaa tasoa.

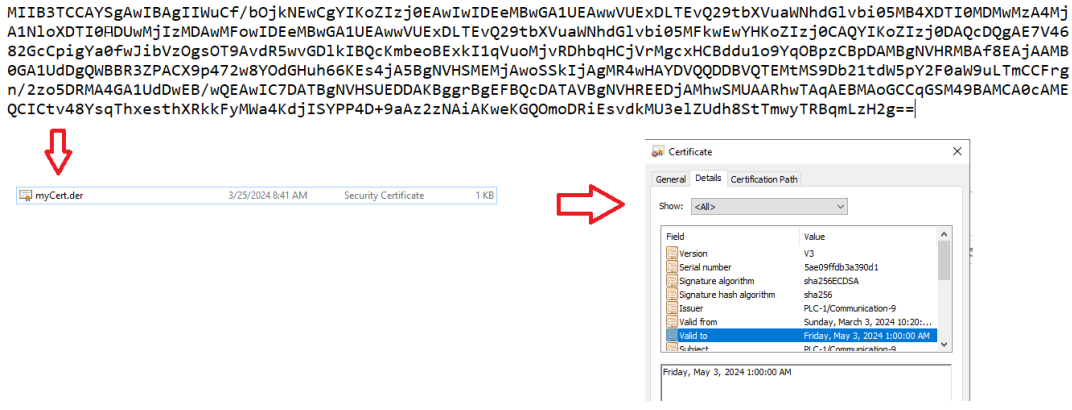
14.2 WinCC 7/8 ja SCADA Export

Oletetaan että käytössä on WinCC 7.5/8.0. Tällöin ”SCADA Export” -työkalua (Siemens. 2024b) on pakko käyttää, jotta TLS-yhteys voi toimia. ”SCADA Export”-ajon jälkeen S7Plus sertifikaatin tiedot löytyvät WinCC:n käyttämän Microsoft SQL Serverin taulusta MCPVCONNECTIONLIST (Kuva 24).



Kuva 24. SQL-haku WinCC:n tietokannasta s7plus -ajurin osalta. Hakutulos palauttaa PG/HMI-kommunikaatiosertifikaatin tiedot.

Jos kyseinen kentän sisältämän infodatan tallentaa tiedostoon ja nimeää sen ”.der” -päätteellä niin lopulta sertifikaatin validiteetin voi tarkistaa ”certutil” -työkalun avulla (Kuva 25).



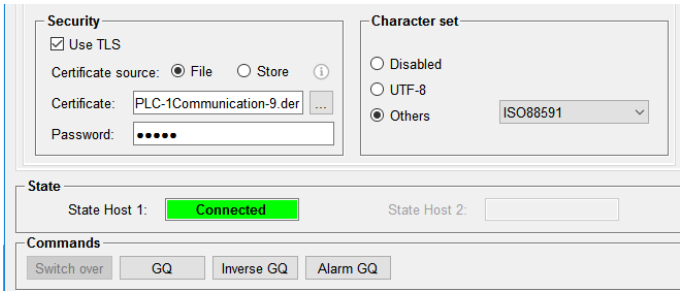
Kuva 25. Tallennus .der -tiedostoon.

Jos edelliset vaiheet ja toimenpiteet haluaisi automatisoida niin WinCC -sovellukseen olisi mahdollista lisätä vbscripti, joka lukisi projektin SQL

ADO/ODBC:n avulla tietokannasta s7pluscert -kentän, tallentaisi sen sisältämän datan tekstitiedostoon .der -päätteellä ja lopulta suorittaisi shell -komennon certutil -käskylle, joka puolestaan purkaisi halutut aikakentät sertifikaatista ja tallentaisi nämä tiedostoon tai lopulta takaisin WinCC:n hmi tag -tietokantamuuttujaan.

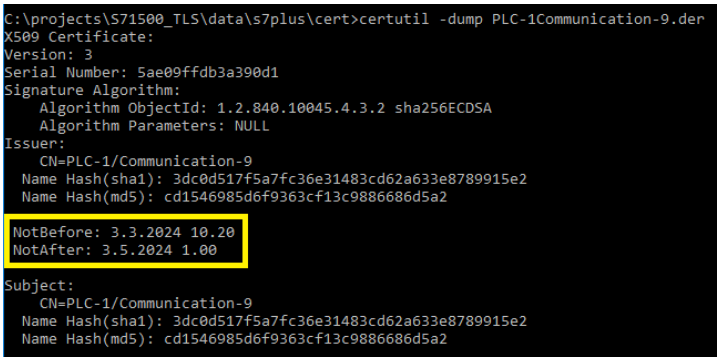
14.3 WinCC OA ja Certutil / OpenSSL

Jos tutkitaan WinCC OA -valvomoa niin tällöin s7plus -sertifikaatin on oltava jo siirrettynä TIA Portal -ohjelmistosta WinCC OA palvelimelle, jotta TLS-pohjainen s7plus -ajurin kommunikaatio voisi yleensäkin toimia. Sertifikaattitiedosto tulee olla tallennettu DER-formaatissa (Kuva 26).

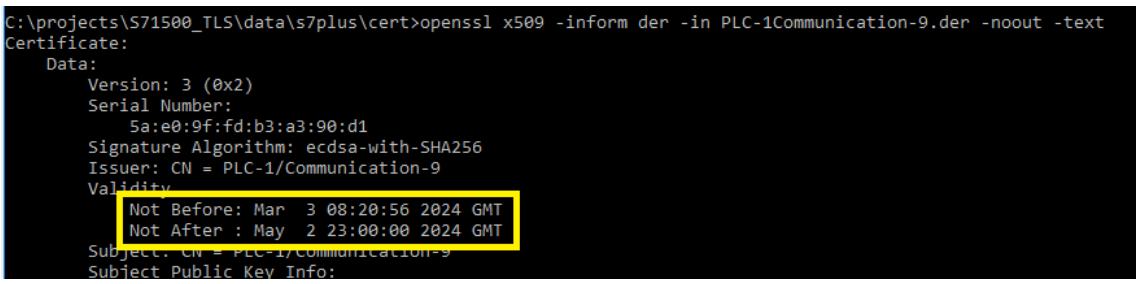


Kuva 26. Näkymä WinCC OA -projektin konfiguraatioruutu, jossa TLS yhteys on luotuna S7-1500 prosessiasemaan.

Myös tällöin ”certutil” -komennolla (Kuva 26) saa selville sertifikaatin validiteettiajan ja tämän informaation voisi putkittaa ja tallentaa ascii -tiedostoon WinCC OA valvomon luettavaksi. Soveltuva komento olisi tässä yhteydessä ”certutil -dump PLC-1Communication-9.der > myCertExport.txt”. Jos ”certutil” -komento ei ole käytettävissä niin vaihtoehtoisesti ”openssl” -komennolla voi myös purkaa sertifikaatin elementit (Kuva 27).



Kuva 27. Certutil -käyttö datan parseroinnissa.



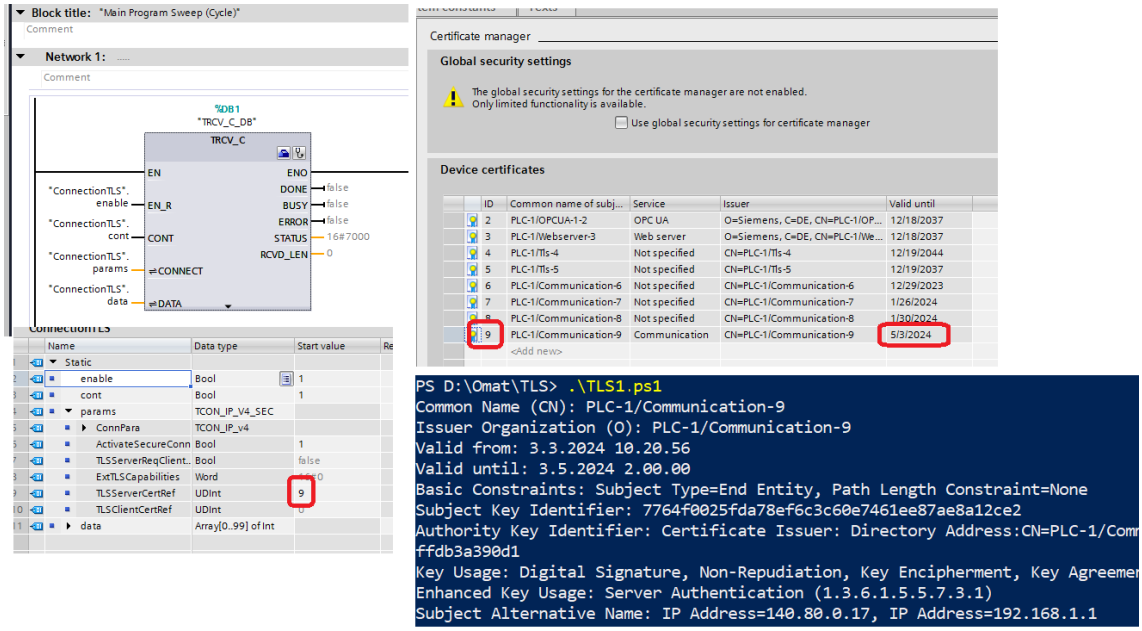
Kuva 28. Openssl -komento datan parseroinnissa.

14.4 WinCC Unified RDF-tiedosto

Jos puolestaan on käytössä WinCC Unified PC Runtime niin PG/HMI-sertifikaatin validiteettiajan tiedot ovat ajonaikaisen projektin kansiossa C:\ProgramData\SCADAProjects\servernimi\currentConfiguration\general ja tiedostossa config_SecureConnectionAttributes.rdf. Tässä opinnäytetyössä rajasin teknisen toteutuksen pois, kuinka tämän informaation aukiparserointi tulisi tehdä.

14.5 S7-1500:n ohjelmointi OUC-käskyjen avulla

Yksi lähestymistapa, joka testattiin, oli S7-1500 CPU:n konfigurointi niin, että se odottaa turvallista yhteyttä muodostettavan jonkun sen portin kautta, esimerkiksi käyttämällä TRCV_C-lohkoa (Kuva 28). Tämän seurauksena S7-CPU myös reagoi yhteydenmuodostuspyyntöihin lähettämällä asiakasohjelmalle sen omat kommunikaatiosertifikaatit. Valittava kommunikaatiosertifikaatti määritetään viittaamalla kyseisen lohkon parametreihin. Tässä yhteydessä on järkevää valita PG/HMI-kommunikaatiosertifikaatti, jonka tiedot olisivat asiakasohjelman kannalta relevantteja. Asiakasohjelma, saatuaan sertifikaatin, voi sen jälkeen hyödyntää sen tietoja esimerkiksi sen validiteetin varmentamiseen. Tämän menetelmän huonona puolena on se, että se vaatisi sovelluskoodin kirjoittamista ohjausjärjestelmän puolelle, mikä tyyppisesti on sellaista, jota halutaan välttää.



Kuva 29. Näkymä TIA Portal -projektista sekä tuloste powershell-skriptistä komentoriviltä.

15 Pohdintaa ja johtopäätökset

Ohjelmiston kehittäminen TLS-sertifikaattien automaattiseen valvontaan S7-1500 -automaatiolaitteista tuo merkittäviä etuja teollisuusautomaation tietoturvan ja hallinnan kannalta. Tällainen ohjelmisto ei ainoastaan vähennä manuaalisen seurannan tarvetta vaan myös tehostaa tietoturvan ylläpitoa kriittisissä infrastruktuureissa. Kuitenkin, kuten kaikessa teknologiassa, ohjelmistolla on sekä vahvuuksia että haasteita, jotka vaikuttavat sen käyttöikään ja tehokkuuteen.

Opinnäytetyön lopputuloksena syntyneen ohjelmiston hyviä puolia ovat:

Automatisoitu tietoturvan parannus

Automatisoimalla TLS-sertifikaattien valvonta, ohjelmisto vähentää inhimillisen virheen riskiä, joka liittyy sertifikaattien vanhentumisen ja konfiguraatiovirheiden manuaaliseen tarkistukseen. Tämä parantaa merkittävästi järjestelmän tietoturvaa ja käytettävyyttä.

Reaaliaikainen seuranta ja hälytykset

Ohjelmiston kyky valvoa sertifikaatteja reaaliajassa ja lähettää WinCC:n kautta hälytyksiä poikkeamista tai vanhentumisista varmistaa, että tietoturvapäälliköt ja IT-henkilöstö voivat reagoida välittömästi mahdollisiin ongelmiin.

Resurssien säästö

Automatisoimalla rutiininomaiset valvontatehtävät, yritykset voivat vapauttaa IT sekä kunnossapidon henkilöstön resursseja muita kriittisiä tehtäviä varten, parantaen näin työn tehokkuutta ja vähentäen kustannuksia.

Mahdollisia haasteita saattaisi olla ohjelmiston käyttöönotossa. Sen alkuvaiheen konfigurointi edellyttää teknistä tietämystä, mikä voi muodostua esteeksi erityisesti pienemmissä organisaatioissa, joissa ei ole suoraa pääsyä IT-asiantuntijaresursseihin. Lisäksi TLS-protokollien ja sertifikaattistandardien

jatkuvan kehityksen vuoksi ohjelmisto vaatii säännöllisiä päivityksiä, jotta se pysyy ajantasaisena uusimpien turvallisuusvaatimusten kanssa. Tämän jatkuva päivitystarve voi haastaa ohjelmiston pitkäikäisyyden ja edellyttää aktiivista ylläpitoa.

Opinnäytetyön aikana käytetty SAT SDK -paketti sisälsi tuen S7-1500 CPU fimware 3.0:lle. Jatkossa Siemens AG toki julkaisee tuen sille sisältäen S7-1500 CPU firmware 3.1 ja uudemmat mallit, jolloin mahdollisesti tarvitaan myös uudempi päivitysversio PKIMonitorista -sovelluksesta.

PKIMonitor on ilmainen sovellus ja tarvittaessa sen saa pyytämällä allekirjoittaneelta ottamalla yhteyttä Suomen Siemens Osakeyhtiön teollisuuden tekniseen tukeen.

16 Yhteenveto

Opinnäytetyössä analysoitiin S7-1500 -ohjausjärjestelmän ja WinCC -valvomon välisen TLS-suojatun yhteyden sertifikaatin voimassaoloa.

Jatkuvasti tiukentuvat EU direktiivit (NIS2 ja CRA) johtavat tosiasiaan, että IT/OT-laitteista ja verkoista tulee aktivoida tarpeelliset tietoturvaelementit näiltä osin. Automaation operatiivisissa toiminnoissa oleellista on varmistua, että vanhentunut sertifikaatti ei johda yllätyksenä tapahtuvaan kommunikoinnin loppumiseen automaatiolaitteiden välillä, mikä puolestaan voi aiheuttaa tuotantolaitoksessa odottamattomia seisokkeja.

Opinnäytetyön aikana luontiin ohjelmisto, jonka avulla PG/HMI-kommunikaatioon liittyvä sertifikaatti voidaan automaattisesti ladata Ethernet/Profinet -verkon ylitse S71500 -ohjausjärjestelmästä tietokoneelle. Ohjelmisto osaa myös purkaa tarvittavat elementit sertifikaatista ja luoda paikallisen datatiedoston sisältäen informaation, jossa on ilmoitettuna, kuinka monta päivää on jäljellä ennen varmenteen vanhentumista. Ohjelmiston tekoon käytettiin Visual Studio 2022 -kehitintä sekä Simatic Automation Tool SDK C# API-pakettia.

Ohjelmiston jatkokehitystä ajatellen seuraavat askeleet olisivat implementoida profiilit uudemmille S7-1500/1200/ET200SP -malleille, Linux-käyttöjärjestelmän huomioonotto sekä tuki S7-1500:n JSON-RPC Web API-protokollalle (Siemens. 2023g. 153), jonka avulla voidaan analysoida ohjausjärjestelmän muita diagnostiikkatietoja.

Lopuksi voidaan todeta, että vaikka ohjelmisto tarjoaa etuja TLS-sertifikaattien hallinnassa, sen tehokas ja turvallinen käyttö edellyttää jatkuvaa sitoutumista ylläpitoon, päivityksiin ja teknologian seurantaan. Tällaisen ohjelmiston kehittäminen ja ylläpito vaatii jossain määrin resursseja, mutta sen tarjoamat hyödyt painavat vaakakupissa enemmän kuin käyttöönoton alkuvaiheen mahdolliset haasteet ja jatkuva ylläpidon tarve.

Lähteet

Arfan Sharif. 2023. Syslog Logging Guide: The Basics. Viitattu 15.3.2024. https://www.crowdstrike.com/guides/syslog-logging

Baivab Kumar Jena. Digital Signature Algorithm (DSA) in Cryptography: How It Works & More. Viitattu 15.3.2024. https://www.simplilearn.com/tutorials/cryptography-tutorial/digital-signature-algorithm

Carlisle Adams, Steve Lloyd. 2023. Understanding PKI: Concepts, Standards, and Deployment Considerations. ISBN 0-672-32391-5

Dionisie Gitlan. 2024a. Symmetric vs Asymmetric Encryption: The Ultimate Comparative Guide. Viitattu 10.3.2024. https://www.ssldragon.com/blog/symmetric-asymmetric-encryption/

Dionisie Gitlan. 2024b. TLS Handshake Explained: Protocols, Processes, and Encryption Standards. Viitattu 12.3.2024. https://www.ssldragon.com/blog/tls-handshake/

Eric Rescorla. 2018. “The Transport Layer Security (TLS) Protocol Version 1.3”. Viitattu 6.3.2024. https://datatracker.ietf.org/doc/html/rfc8446

Hans Berger. 2014. Automation with Simatic S7-1500, ePDF ISBN 978-3-89578-919-9

Hazem Sulaiman. Siemens. 2022. Certificate Handling in FA Environment. Viitattu 3.1.2024. Vaatii kirjautumisen. https://see-siemens.highspot.com/items/6565cc017bc76a5e06ca5948?lfrm=srp.0#43

Heikki Kallankari. 2020. TLS ja palvelimien turvallisuuden arviointi. Viitattu 3.3.2024. https://urn.fi/URN:NBN:fi:amk-2020122930037

IETF RFC 5280. 2008. "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile". Viitattu 25.3.2024. https://www.rfc-editor.org/info/rfc5280

JetBrains. 2024. Viitattu 22.2.2024. https://www.jetbrains.com

Kittler & Sulaiman. 2023. Impact of Cyberresiliance Act. Viitattu 3.1.2024. Vaatii kirjautumisen. https://see-siemens.highspot.com/items/65607b608705469d28903605?lfrm=srp.0

Mordor Intelligence. PLC Market Size & Share Analysis - Growth Trends & Forecasts (2024 - 2029). Viitattu 3.3.2024. https://www.mordorintelligence.com/industry-reports/programmable-logic-controller-plc-market

Microsoft. .NET Support Policy. Viitattu 13.3.2024. https://dotnet.microsoft.com/en-us/platform/support/policy

Microsoft. How to: View certificates with the MMC snap-in. Viitattu 29.3.2024. https://learn.microsoft.com/en-us/dotnet/framework/wcf/feature-details/how-to-view-certificates-with-the-mmc-snap-in

Micosoft Teams. 2024. Viitattu 1.4.2024. https://www.microsoft.com/fi-fi/microsoft-teams/group-chat-software

NIST SP 800-57. 2020a. Recommendation for Key Management. Viitattu 25.3.2024. https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final

NIST SP 800-57, 2020b. Recommendation for Key Management, Part 1 Revision 5 – General. Viitattu 25.3.2024. https://csrc.nist.gov/Projects/Key-Management/Key-Management-Guidelines

Olivia Joki-Hollanti. 2021. WinCC Unifiedin ja WinCC Professionalin tuoteominaisuuksien vertailu. Viitattu 3.3.2024. https://urn.fi/URN:NBN:fi:amk-202105189228

Sectigo. 2023. Viitattu 2.3.2024. https://securityboulevard.com/2023/07/an-overview-of-enterprise-certificate-life-cycle-management/

Siemens. SIMATIC WinCC Unified System. Viitattu 4.3.2024. https://new.siemens.com/global/en/products/automation/simatic-hmi/wincc-unified.html

Siemens. 2020a. Security guide for SIMATIC WinCC Unified and SIMATIC HMI Unified operator devices. Viitattu 10.1.2024. https://cache.industry.siemens.com/dl/files/300/109481300/att_1039769/v2/109481300_SecurityGuidelineUnified_V10_en.pdf

Siemens. 2020b. System overview - SCADA System SIMATIC WinCC Professional. Viitattu 1.3.2024. https://assets.new.siemens.com/siemens/assets/api/uuid:1f589ab9-4467-405d-b667-89a66a35e17f/df-fa-i10077-00-7600-ipdf-wincc-systemoverview-en.pdf

Siemens. 2021. Simatic Transfer Tool. Viitattu 29.3.2024 https://support.industry.siemens.com/cs/ww/en/view/109782475

Siemens. 2022a. Security with SIMATIC controllers. Viitattu 2.4.2024. https://cache.industry.siemens.com/dl/files/010/90885010/att_1121268/v1/90885010_Security_SIMATIC_Controller_V30_en.pdf

Siemens. 2022b. SIMATIC STEP 7 and WinCC Engineering V18. A5E51856270-AA. Viitattu 2.1.2024. https://cache.industry.siemens.com/dl/files/056/109815056/att_1121875/v5/STEP_7_WinCC_V18_enUS_en-US.pdf

Siemens. 2022c. WinCC Open Architecture Your SCADA in few words and facts. Viitattu 22.2.2024. Vaatii kirjautumisen. https://see-siemens.highspot.com/items/62a368bba6e78907050ecd26?lfrm=srp.3#1

Siemens. 2023a. Configuration of TLS-based PG/HMI Communication And The Protection of Confidential PLC Configuration Data. 2/2023. Viitattu 25.3.2024. https://cache.industry.siemens.com/dl/files/583/109798583/att_1130822/v2/109798583_SecurityConcept_TIA_V_17_V11_en.pdf

Siemens. 2023b. Libraries for Communication for SIMATIC Controllers. Viitattu 29.3.2024. https://cache.industry.siemens.com/dl/files/503/109780503/att_1155732/v2/109780503_Libraries_Comm_Controller_DOC_V2_0_1_en.pdf

Siemens. 2023c. SIMATIC Automation Tool SDK Linux user guide. Viitattu 29.3.2024. https://cache.industry.siemens.com/dl/files/374/109817374/att_1131740/v1/SAT_SDK_UserGuide_V5_0_Linux_en-US_en-US.pdf

Siemens. 2023d. SIMATIC Automation Tool SDK Linux installation notes. Viitattu 2.4.2024. https://cache.industry.siemens.com/dl/files/368/109817368/att_1131144/v1/SAT_SDK_InstallationNotes_V5_0_Linux_en-US_en-US.pdf

Siemens. 2023e. SIMATIC Automation Tool SDK Windows user guide - Programming Manual. A5E45253030-AI. https://cache.industry.siemens.com/dl/files/364/109817364/att_1131129/v1/SAT_SDK_UserGuide_V5_0_Windows_en-US_en-US.pdf

Siemens. 2023f. Simatic Communication Function Manual A5E03735815-AL. Viitattu 2.2.2024. https://cache.industry.siemens.com/dl/files/925/59192925/att_901175/v2/s71500_communication_function_manual_en-US_en-US.pdf

Siemens. 2023g. S7-1500 Web server Function Manual. Viitattu 4.4.2024. https://cache.industry.siemens.com/dl/files/560/59193560/att_898124/v2/s71500_webserver_function_manual_en-US_en-US.pdf

Siemens. 2024a. SIMATIC Automation Tool user guide Application Manual. A5E45044277-AK. Viitattu 4.4.2024. https://cache.industry.siemens.com/dl/files/741/109827741/att_1167048/v1/SAT_UserGuide_en-US.pdf

Siemens. 2024b. SIMATIC SCADA Export for TIA Portal. Viitattu 29.3.2024 https://support.industry.siemens.com/cs/us/en/view/109748955

Stephen A. Thomas. 2000. SSL & TLS Essentials Securing the Web. ISBN-13 978-0471383543

William Stallings. 2017. Cryptography and Network Security: Principles and Practice. ISBN 13: 978-1-292-15858-7

Yu Jordan Zhao, 2023. Popular server-based SCADA platforms in the world. Viitattu 29.3.2024. https://www.acectrl.com/a-list-of-the-most-popular-server-based-scada-platforms-in-the-world/.

Kappaleen 14.4. Powershell-osuus

```powershell
$IP = '140.80.0.17'
$PORT = 2000
# Create a callback to bypass certificate validation (use cautiously)
$ServerCertificateValidationCallback = {
    param(
        $sender,
        $certificate,
        $chain,
        $sslPolicyErrors
    )
    return $true
}
# Establish the TCP connection
$tcpClient = New-Object Net.Sockets.TcpClient
$tcpClient.Connect($IP, $PORT)
# Create the SSL stream with the certificate validation callback to bypass certificate errors
$sslStream = New-Object Net.Security.SslStream($tcpClient.GetStream(), $false, $ServerCertificateValidationCallback)
# Attempt to authenticate the SSL connection
try {
    $sslStream.AuthenticateAsClient($IP)
} catch {
    Write-Host "Failed to authenticate: $_"
    $tcpClient.Close()
    return
}
# Retrieve the certificate from the SSL stream
$cert = [System.Security.Cryptography.X509Certificates.X509Certificate2]$sslStream.RemoteCertificate
# Get the certificate's subject common name (CN)
$cn = $cert.GetNameInfo([System.Security.Cryptography.X509Certificates.X509NameType]::SimpleName, $false)
Write-Host "Common Name (CN): $cn"
# Get the certificate's issuer organization (O)
$issuer = $cert.GetNameInfo([System.Security.Cryptography.X509Certificates.X509NameType]::SimpleName, $true)
Write-Host "Issuer Organization (O): $issuer"
# Get the certificate's validity period
$not_before = $cert.GetEffectiveDateString()
$not_after = $cert.GetExpirationDateString()
Write-Host "Valid from: $not_before"
Write-Host "Valid until: $not_after"
# Get extensions
foreach ($ext in $cert.Extensions) {
    $ext_name = $ext.Oid.FriendlyName
    $ext_value = $ext.Format($false)
    Write-Host "${ext_name}: $ext_value"
}
# Cleanup
$sslStream.Close()
$tcpClient.Close()
```

Otos valvontaohjelmistossa käytetystä funktiosta, jossa on kuvattu vaiheittain tapahtumat PG/HMI-kommunikaatiossa käytetyn sertifikaatin analysointiin

```
// PKI Info
Console.WriteLine(resourceString.GetString("Certificates", language));

// Do we have any certificates?
if (CurrentCPU.CertificateStore.Count > 0)
{
   Console.WriteLine("number of certs: " + CurrentCPU.CertificateStore.Count);

   foreach (ICertificate mycert in CurrentCPU.CertificateStore)
   {
       Byte[] CertDataInfo;
       CertDataInfo = mycert.Data;
       Console.WriteLine("Certificate information in raw format: " + Encoding.Default.GetString(CertDataInfo));

       try
       {
           // Create an X509Certificate2 object from the binary blob.
           X509Certificate2 certificate = new X509Certificate2(CertDataInfo);

           // Extract the subject's name from the certificate.
           string subjectName = certificate.SubjectName.Name;

           // Extract the certificate's start time (NotBefore) and end time (NotAfter).
           DateTime startTime = certificate.NotBefore;
           DateTime endTime = certificate.NotAfter;

           string thumbPrint = certificate.Thumbprint;


           Console.WriteLine("Certificate Subject Name: " + subjectName);
           Console.WriteLine("Certificate Thumbprint: " + thumbPrint);

           //save data to txt-file.
           File.WriteAllText(AppDomain.CurrentDomain.BaseDirectory + @"\options\\thumbprint.txt", thumbPrint);

           Console.WriteLine("Start Time: " + startTime);
           Console.WriteLine("End Time: " + endTime);


           //calc time difference
           var today = DateTime.Now;

           //get difference of two dates
           var diffOfDates = endTime -today;
           Console.WriteLine("Days left before the certificate expires: " + diffOfDates.Days);

           //save data to txt-file.
           File.WriteAllText(AppDomain.CurrentDomain.BaseDirectory + @"\options\\warn_days_before_expired.txt", diffOfDates.Days.ToString());

       }
       catch (Exception ex)
       {
           Console.WriteLine("Error: " + ex.Message);
       }
   }
}
```