

Spółeczeństwo informacyjne
Projekt
„Protokoły komunikacyjne w przemyśle”
Bazan Michał – 163881

Spis treści

1. Wstęp.....	3
1.1. Wprowadzenie do tematu komunikacji w systemach przemysłowych.....	3
1.2. Znaczenie niezawodnej i bezpiecznej komunikacji w przemyśle.....	3
2. Przegląd teoretyczny protokołów komunikacyjnych.....	4
2.1. Modbus TCP/IP.....	4
2.2. Ethernet/IP.....	7
2.3. Profinet.....	8
2.4. CANopen.....	8
2.5. Profibus.....	9
2.6. DeviceNet.....	9
3. Praktyczne zastosowanie protokołów.....	10
3.1. Konfiguracja komunikacji Modbus TCP/IP pomiędzy dwoma komputerami.....	10
3.2. Konfiguracja komunikacji Ethernet/IP między dwoma komputerami.....	12
4. Bezpieczeństwo komunikacji.....	14
4.1. Przegląd zagrożeń w komunikacji przemysłowej.....	14
4.1.1. Modbus TCP/IP.....	14
4.1.2. Ethernet/IP.....	16
5. Podsumowanie.....	18
Modbus TCP/IP:.....	18
Ethernet/IP:.....	18
Źródła.....	20

1. Wstęp

Projekt ma na celu przedstawienie i zrozumienie różnych protokołów komunikacyjnych stosowanych w systemach przemysłowych oraz przeprowadzenie testów komunikacyjnych przy użyciu dwóch komputerów. Projekt będzie obejmować aspekty teoretyczne i praktyczne, a także zagadnienia związane z bezpieczeństwem komunikacji.

1.1. Wprowadzenie do tematu komunikacji w systemach przemysłowych

W dzisiejszych czasach systemy przemysłowe są coraz bardziej złożone i wymagają niezawodnej komunikacji między różnymi urządzeniami i systemami. Komunikacja w systemach przemysłowych odnosi się do wymiany informacji między maszynami, czujnikami, sterownikami i innymi komponentami automatyki. Dzięki skutecznej komunikacji możliwe jest monitorowanie, kontrolowanie i optymalizowanie procesów produkcyjnych oraz zarządzanie operacjami w czasie rzeczywistym.

Protokoły komunikacyjne odgrywają kluczową rolę w zapewnieniu skutecznej wymiany danych. Protokoły te definiują zasady, zgodnie z którymi urządzenia mogą komunikować się ze sobą, określając format danych, metody synchronizacji oraz mechanizmy transmisji. W systemach przemysłowych stosowane są różne protokoły, w tym Modbus, Profibus, CAN, Ethernet/IP oraz OPC UA, które różnią się pod względem zastosowania, topologii sieci oraz poziomu złożoności.

1.2. Znaczenie niezawodnej i bezpiecznej komunikacji w przemyśle

Niezawodność komunikacji w systemach przemysłowych jest kluczowa dla zapewnienia ciągłości procesów produkcyjnych i minimalizacji ryzyka awarii. W przemyśle, gdzie każda minuta przestoju może wiązać się z dużymi stratami finansowymi, niezawodna wymiana informacji między urządzeniami pozwala na szybkie reagowanie na zmiany warunków produkcji oraz natychmiastowe podejmowanie działań naprawczych w przypadku wykrycia problemów.

Bezpieczna komunikacja jest równie ważna, zwłaszcza w kontekście rosnącej liczby cyber zagrożeń. Systemy przemysłowe są coraz częściej celem ataków hakerskich, które mogą prowadzić do kradzieży danych, uszkodzenia sprzętu, a nawet do zakłócenia całego procesu produkcyjnego. Dlatego też wprowadzenie odpowiednich środków bezpieczeństwa, takich jak szyfrowanie danych, uwierzytelnianie użytkowników oraz monitorowanie sieci, jest niezbędne dla ochrony systemów przemysłowych przed nieautoryzowanym dostępem i innymi zagrożeniami.

Podsumowując, niezawodna i bezpieczna komunikacja w systemach przemysłowych jest fundamentem efektywnego i bezpiecznego zarządzania procesami produkcyjnymi. W dalszych rozdziałach tego projektu zostaną szczegółowo omówione wybrane protokoły komunikacyjne, ich praktyczne zastosowania oraz metody zapewnienia bezpieczeństwa komunikacji w środowisku przemysłowym.

2. Przegląd teoretyczny protokołów komunikacyjnych

2.1. Modbus TCP/IP

Modbus [1] jest jednym z najstarszych i najbardziej powszechnie stosowanych protokołów komunikacyjnych w automatyce przemysłowej. Został opracowany w 1979 roku przez firmę Modicon (obecnie część Schneider Electric) dla sterowników PLC. Oryginalnie Modbus wykorzystywał transmisję szeregową (RS-232/RS-485), jednak wraz z rozwojem technologii sieciowych, został przystosowany do pracy w sieciach Ethernet, dając początek Modbus TCP/IP.

Modbus TCP/IP jest wersją Modbus opartą na protokole TCP/IP, co pozwala na wykorzystanie standardowej infrastruktury sieciowej Ethernet do komunikacji między urządzeniami. Główne elementy architektury Modbus TCP/IP to:

- **Master/Slave:** Modbus TCP/IP działa w modelu klient-serwer, gdzie jeden z urządzeń pełni rolę klienta (master), wysyłającego zapytania, a inne urządzenia są serwerami (slave), które odpowiadają na te zapytania.
- **Adresacja:** Każde urządzenie w sieci Modbus TCP/IP ma unikalny adres IP, a każdy serwer może mieć do 247 urządzeń podrzędnych (adresowanych od 1 do 247).
- **Ramki danych:** Dane są przesyłane w formie ramek, które składają się z adresu urządzenia, kodu funkcji, danych oraz sumy kontrolnej.

Modbus TCP/IP jest szeroko stosowany w różnych sektorach przemysłu, w tym w automatyce budynkowej, systemach HVAC, systemach monitorowania i kontroli procesów przemysłowych, oraz w energetyce. Jego prostota i niezawodność sprawiają, że jest preferowany w wielu aplikacjach, gdzie wymagana jest stabilna i łatwa do implementacji komunikacja.

Modbus TCP/IP obsługuje różne funkcje, w tym:

- **Czytanie i zapisywanie rejestrów:** Umożliwia odczyt i zapis danych w rejestrach urządzeń.
- **Diagnostyka:** Umożliwia wykonywanie operacji diagnostycznych na urządzeniach.
- **Zarządzanie urządzeniami:** Umożliwia zarządzanie i konfigurację urządzeń w sieci.

Zalety:

- Prosty i łatwy do implementacji.
- Wysoka kompatybilność z różnymi urządzeniami.
- Wykorzystuje istniejącą infrastrukturę sieciową Ethernet.

Wady:

- Ograniczone funkcje w porównaniu z bardziej zaawansowanymi protokołami.
- Brak wbudowanych mechanizmów bezpieczeństwa.

Ramka modbus w trybie ASCII

:	Adres		Kod funkcji		Dane			Suma kontrolna		CR	LF
						...					

Bajty wysyłane są szesnastkowo (po dwa znaki ASCII) a odstępy pomiędzy kolejnymi znakami ramki są mniejsze niż 1 sekunda.

Ramka komunikacyjna w trybie RTU

Adres		Kod funkcji		Dane			Suma kontrolna	
					...			

Bajty wysyłane są jako znaki ośmiobitowe a każda ramka jest poprzedzona odstępem większym niż 3.5T (gdzie T oznacza czas transmisji jednego znaku).

Znaczenie bajtów

- adres:

0 – adres rozgłoszeniowy

1 – 247 – adres jednostki server

- kod funkcji:

1 \$01 odczyt wyjść bitowych

2 \$02 odczyt wejść bitowych

3 \$03 odczyt n rejestrów

4 \$04 odczyt n rejestrów wejściowych

5 \$05 zapis 1 bitu

6 \$06 zapis 1 rejestru

7 \$07 odczyt statusu

8 \$08 test diagnostyczny

15 \$0F zapis n bitów

16 \$10 zapis n rejestrów

17 \$11 identyfikacja urządzenia server

128 – 255 \$80–\$FF zarezerwowane na odpowiedzi błędne

Obsługa błędów komunikacji:

- odesłanie przez server ramki z kodem błędu:

01 – niedozwolona funkcja

02 – niedozwolony numer rejestru

03 – niedozwolona wartość danej

04 – uszkodzenie w przyłączonym urządzeniu

05 – potwierdzenie pozytywne

06 – brak gotowości, komunikat usunięty

07 – potwierdzenie negatywne

08 – błąd parzystości pamięci

- przy przekroczeniu czasu oczekiwania na odpowiedź serwer nie odsyła odpowiedzi.

2.2. Ethernet/IP

Ethernet/IP [2] (Ethernet Industrial Protocol) został opracowany przez Rockwell Automation i jest zarządzany przez organizację ODVA (Open DeviceNet Vendors Association). Ethernet/IP wykorzystuje standardowy protokół Ethernet i warstwę transportową TCP/IP, integrując je z przemysłowym protokołem aplikacyjnym CIP (Common Industrial Protocol), co pozwala na jego szerokie zastosowanie w przemysłowych systemach automatyki.

Ethernet/IP działa w modelu producent-konsument, co oznacza, że dane mogą być przesyłane od jednego nadawcy (producenta) do wielu odbiorców (konsumentów). Kluczowe elementy architektury Ethernet/IP to:

- **CIP (Common Industrial Protocol):** CIP zapewnia jednolity model danych i mechanizmy komunikacji, które mogą być używane w różnych sieciach przemysłowych.
- **TCP/IP i UDP:** Ethernet/IP wykorzystuje zarówno TCP/IP do niezawodnych transmisji oraz UDP do szybkich i mniej wymagających czasowo transmisji danych.
- **Enkapsulacja:** Dane CIP są enkapsulowane w standardowe ramki Ethernet [3], co pozwala na wykorzystanie standardowych urządzeń sieciowych, takich jak switchy i routery.

Ethernet/IP jest szeroko stosowany w automatyce przemysłowej, w tym w systemach sterowania procesami, systemach monitorowania, systemach transportowych, a także w robotyce i automatyzacji produkcji. Jego zdolność do obsługi dużych ilości danych w czasie rzeczywistym sprawia, że jest idealnym rozwiązaniem dla aplikacji wymagających wysokiej wydajności.

Ethernet/IP oferuje szereg funkcji, które obejmują:

- **Kontrola cykliczna (I/O):** Umożliwia szybką i niezawodną wymianę danych wejścia/wyjścia między urządzeniami.
- **Messaging (Explicit Messaging):** Umożliwia wysyłanie specjalnych komunikatów konfiguracyjnych, diagnostycznych i sterujących.
- **Bezpieczeństwo:** Ethernet/IP obsługuje funkcje bezpieczeństwa, takie jak szyfrowanie danych i uwierzytelnianie, co jest istotne w nowoczesnych systemach przemysłowych.

Zalety:

- Wysoka przepustowość i niskie opóźnienia.
- Możliwość obsługi dużej liczby urządzeń w sieci.
- Zintegrowane funkcje bezpieczeństwa.

Wady:

- Złożoność implementacji w porównaniu z prostszymi protokołami.
- Wyższe wymagania sprzętowe i kosztowe.

2.3. Profinet

PROFINET [6] to protokół oparty na standardzie Ethernet, zapewniający szybką transmisję danych w czasie rzeczywistym. Działa w modelu klient-serwer, podobnie jak Modbus TCP/IP.

Struktura i działanie: Każde urządzenie w sieci PROFINET ma unikalny adres IP, a komunikacja odbywa się za pomocą ramek danych Ethernet z dodatkowymi nagłówkami PROFINET. Protokół ten znajduje zastosowanie w różnych obszarach przemysłu, takich jak automatyka fabryczna, robotyka czy systemy wizualizacji.

Zastosowania: PROFINET jest używany w różnych sektorach przemysłu, w tym w automatyce fabrycznej, robotyce, systemach wizualizacji i sterowaniu produkcją.

Zalety:

- Szybka transmisja danych w czasie rzeczywistym,
- Wsparcie dla różnych topologii sieciowych,
- Łatwa integracja z istniejącymi sieciami Ethernet.

Wady:

- Ryzyko zakłóceń w środowisku przemysłowym.

2.4. CANopen

CANopen [7] to protokół oparty na standardzie Controller Area Network (CAN), oferujący szybką komunikację w czasie rzeczywistym.

Struktura i działanie: Komunikacja w CANopen odbywa się za pomocą ramek danych CAN z dodatkowymi nagłówkami zawierającymi informacje o identyfikatorze urządzenia.

Zastosowania: CANopen znajduje zastosowanie w różnych obszarach przemysłu, takich jak silniki, roboty przemysłowe czy urządzenia medyczne.

Zalety:

- Szybka komunikacja w czasie rzeczywistym,
- Łatwa konfiguracja i integracja z różnymi urządzeniami.

Wady:

- Ograniczona przepustowość sieci CAN.

2.5. Profibus

Profibus [8] to starszy protokół przemysłowy oferujący szybką transmisję danych i obsługę dużych sieci przemysłowych.

Struktura i działanie: Protokół ten wykorzystuje ramki danych Profibus z dodatkowymi nagłówkami zawierającymi informacje o typie wiadomości i adresie urządzenia.

Zastosowania: Profibus jest stosowany w różnych aplikacjach przemysłowych, takich jak kontrolery PLC, przemienniki częstotliwości czy sensory.

Zalety:

- Szybka transmisja danych,
- Obsługa dużych sieci przemysłowych.

Wady:

- Możliwość interferencji elektromagnetycznych.

2.6. DeviceNet

DeviceNet [9] to protokół oparty na standardzie CAN, umożliwiający łatwą integrację różnych urządzeń w sieć.

Struktura i działanie: Komunikacja w DeviceNet odbywa się za pomocą ramek danych CAN z dodatkowymi nagłówkami zawierającymi informacje o identyfikatorze urządzenia.

Zastosowania: DeviceNet znajduje zastosowanie w różnych aplikacjach przemysłowych, takich jak systemy monitorowania i sterowania, systemy diagnostyki czy systemy kontroli ruchu.

Zalety:

- Łatwa integracja różnych urządzeń w sieć,
- Prosta konfiguracja.

Wady:

- Ograniczona przepustowość sieci CAN,
- Możliwość interferencji elektromagnetycznych.

3. Praktyczne zastosowanie protokołów

W tej części projektu zostaną omówione kroki konieczne do skonfigurowania komunikacji Modbus TCP/IP oraz Ethernet/IP pomiędzy dwoma komputerami: komputerem fizycznym z systemem Linux oraz maszyną wirtualną również z systemem Linux.

3.1. Konfiguracja komunikacji Modbus TCP/IP pomiędzy dwoma komputerami

```
sudo apt update -y
sudo apt install python3 python3-pip -y
pip3 install pymodbus
```

Listing 3.1.1 Instalacja python3 i biblioteki pymodbus

```
import asyncio
import logging
import sys
import pymodbus.client as modbus_async_client
from pymodbus.exceptions import ModbusException

async def run_modbus_client():
    client = modbus_async_client.ModbusTcpClient('127.0.0.1', port=5020)

    try:
        client.connect()

        rr_coils = client.read_coils(0, 10)
        rr_registers = client.read_holding_registers(0, 10)

        print("Odczytane I/O: ", rr_coils.bits)
        print("Odczytane rejestry przed ustawieniem: ", rr_registers.registers)

        write_registers = [i for i in range(10)]
        client.write_registers(0, write_registers)
        rr_registers_after = client.read_holding_registers(0, 10)
        print("Odczytane rejestry po ustawieniu: ", rr_registers_after.registers)

    except ModbusException as e:
        print("Błąd Modbus:", e)

    finally:
        client.close()

async def main():
    await run_modbus_client()

if __name__ == "__main__":
    asyncio.run(main())
```

Listing 3.1.2. Klient modbus

Klient Modbus został zaprojektowany do komunikacji z serwerem Modbus za pomocą protokołu TCP/IP. Skrypt klienta napisany jest w języku Python i korzysta z biblioteki `pymodbus`. Po nawiązaniu połączenia z serwerem, klient odczytuje dane wejścia/wyjścia (I/O) oraz rejestry, a

następnie wykonuje operację zapisu na wybranych rejestrach. Każda operacja jest obsługiwana asynchronicznie, co pozwala na efektywne zarządzanie komunikacją Modbus. Po zakończeniu operacji, klient zamyka połączenie, co zapewnia prawidłowe zakończenie sesji komunikacyjnej.

```
import asyncio
import logging
import sys

from pymodbus import __version__ as pymodbus_version
from pymodbus.datastore import (
    ModbusSequentialDataBlock,
    ModbusServerContext,
    ModbusSlaveContext,
    ModbusSparseDataBlock,
)
from pymodbus.device import ModbusDeviceIdentification
from pymodbus.server import (
    StartAsyncSerialServer,
    StartAsyncTcpServer,
    StartAsyncTlsServer,
    StartAsyncUdpServer,
)

block = ModbusSequentialDataBlock(0x00, [0] * 100)
store = ModbusSlaveContext(di=block, co=block, hr=block, ir=block)
context = ModbusServerContext(slaves=store, single=True)

async def run_server():
    server = await StartAsyncTcpServer(context, address=("0.0.0.0", 5020))
    await server.serve_forever()

asyncio.run(run_server())
```

Listing 3.1.3. Serwer modbus

Serwer Modbus został stworzony w celu obsługi żądań klientów Modbus TCP/IP. Skrypt serwera został również napisany w języku Python, wykorzystując bibliotekę `pymodbus`. Serwer został skonfigurowany do obsługi komunikacji na porcie 5020. Po uruchomieniu, serwer nasłuchuje na wszystkich interfejsach sieciowych (0.0.0.0), co umożliwia klientom nawiązanie połączenia z dowolnego adresu IP w sieci lokalnej. Serwer Modbus obsługuje komunikację asynchroniczną, co pozwala na efektywne przetwarzanie wielu żądań klientów jednocześnie.

```
michael@debian:~/spoleczenstwo_informacyjne_projekt/scripts/modbus$ python3
modbus_client.py

Odczytane I/O: [False, False, False, False, False, False, False, False, False, False,
False, False, False, False, False]
Odczytane rejestry przed ustawieniem: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Odczytane rejestry po ustawieniu: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Listing 3.1.4. Logi klienta

Porównanie danych przed i po operacji zapisu pozwala na weryfikację poprawności działania klienta oraz potwierdzenie wykonania operacji zapisu na serwerze Modbus.

3.2. Konfiguracja komunikacji Ethernet/IP między dwoma komputerami

Aby pokazać przykład aplikacji protokołu Ethernet/IP na dwóch komputerach, można opisać proces konfiguracji prostego połączenia klient-serwer przy użyciu języka Python i biblioteki socket. Ethernet/IP jest standardem komunikacji przemysłowej bazującym na protokole TCP/IP, więc wykorzystano podstawowe programowanie socketów do zademonstrowania komunikacji między dwoma komputerami.

```
import socket

HOST = '192.168.50.188'
PORT = 65432

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((HOST, PORT))

message = "Witaj, serwerze!"
client_socket.sendall(message.encode('utf-8'))

data = client_socket.recv(1024)
print(f"Otrzymano od serwera: {data.decode('utf-8')}")

client_socket.close()
```

Listing 3.2.1. Kod klienta

Ten kod tworzy klienta, który łączy się z serwerem i wysyła wiadomość. Następnie klient oczekuje na odpowiedź od serwera i wyświetla ją. Po zakończeniu komunikacji z serwerem gniazdo klienta jest zamykane.

Połączenie między serwerem a klientem jest zrealizowane poprzez użycie gniazd TCP/IP. Serwer nasłuchuje na określonym porcie, a klient łączy się z serwerem, wysyła dane i oczekuje na odpowiedź.

```
import socket

HOST = '0.0.0.0'
PORT = 65432

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((HOST, PORT))
server_socket.listen()

print(f"Serwer nasłuchuje na {HOST}:{PORT}")

while True:
    conn, addr = server_socket.accept()
    print(f"Połączono z {addr}")
    data = conn.recv(1024)
    if not data:
        break
    print(f"Otrzymano: {data.decode('utf-8')}")
    conn.sendall(data) # echo received data back to the client

conn.close()
```

Listing 3.1.2. Kod serwera

Ten kod serwera implementuje prosty serwer, który nasłuchuje na określonym porcie i obsługuje przychodzące połączenia od klientów. Po nawiązaniu połączenia serwer odbiera dane od klienta, wyświetla je, a następnie odsyła te same dane z powrotem do klienta.

Wykorzystuje on gniazda TCP/IP do komunikacji między klientem a serwerem. Gniazda TCP (Transmission Control Protocol) zapewnia niezawodne, połączeniowe przesyłanie danych między komputerami w sieci. Protokół TCP dba o to, aby dane zostały dostarczone do celu w odpowiedniej kolejności i bez utraty, co sprawia, że jest idealny do zastosowań, gdzie ważna jest integralność danych, takich jak komunikacja klient-serwer.

4. Bezpieczeństwo komunikacji

4.1. Przegląd zagrożeń w komunikacji przemysłowej

4.1.1. Modbus TCP/IP

Podczas analizy pakietów Modbus TCP/IP, zwróć uwagę na następujące potencjalne problemy z bezpieczeństwem:

1. Brak szyfrowania:

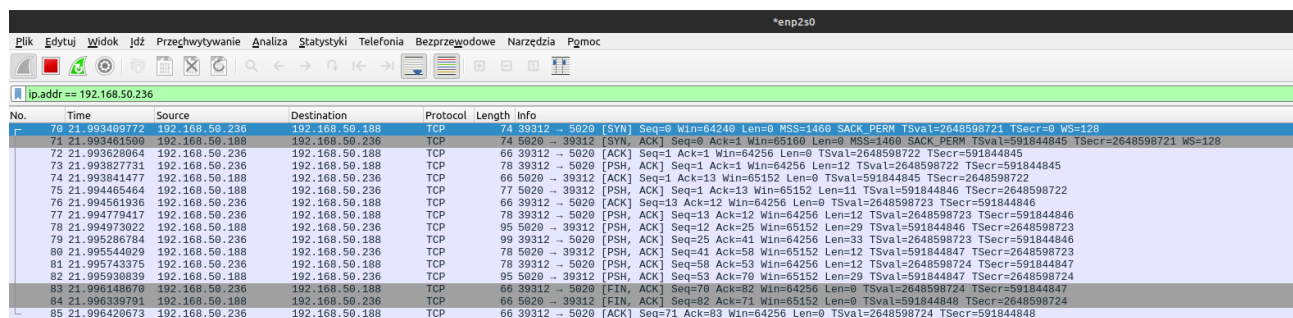
- Wszystkie dane przesyłane w protokole Modbus TCP/IP są przesyłane bez żadnego szyfrowania. Oznacza to, że każdy, kto przechwyci pakiety, może odczytać dane.

2. Brak uwierzytelniania:

- Modbus TCP/IP nie posiada wbudowanego mechanizmu uwierzytelniania, co oznacza, że każdy klient, który ma dostęp do sieci, może komunikować się z serwerem Modbus.

3. Niezabezpieczone komendy kontrolne:

- Możliwość przechwycenia i modyfikacji pakietów kontrolnych (np. polecenia do zmiany stanu urządzeń lub rejestrów), co może prowadzić do nieautoryzowanych zmian w systemie przemysłowym.



No.	Time	Source	Destination	Protocol	Length	Info
70	21.993499772	192.168.50.236	192.168.50.188	TCP	74	5029 → 5020 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2648598721 TSecr=0 WS=128
71	21.993461509	192.168.50.188	192.168.50.236	TCP	74	5020 → 5029 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=591844845 TSecr=2648598721 WS=128
72	21.993629064	192.168.50.236	192.168.50.188	TCP	66	5029 → 5020 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2648598722 TSecr=591844845
73	21.993827731	192.168.50.236	192.168.50.188	TCP	78	5029 → 5020 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=12 TSval=2648598722 TSecr=591844845
74	21.993841477	192.168.50.188	192.168.50.236	TCP	66	5020 → 5029 [ACK] Seq=1 Ack=13 Win=65152 Len=0 TSval=591844845 TSecr=2648598722
75	21.994465464	192.168.50.188	192.168.50.236	TCP	77	5020 → 5029 [PSH, ACK] Seq=1 Ack=13 Win=65152 Len=11 TSval=591844846 TSecr=2648598722
76	21.994561936	192.168.50.236	192.168.50.188	TCP	66	5029 → 5020 [ACK] Seq=13 Ack=12 Win=64256 Len=0 TSval=2648598723 TSecr=591844846
77	21.994779417	192.168.50.236	192.168.50.188	TCP	78	5029 → 5020 [PSH, ACK] Seq=13 Ack=12 Win=64256 Len=12 TSval=2648598723 TSecr=591844846
78	21.994973022	192.168.50.188	192.168.50.236	TCP	95	5020 → 5029 [PSH, ACK] Seq=12 Ack=25 Win=65152 Len=20 TSval=591844846 TSecr=2648598723
79	21.995286784	192.168.50.236	192.168.50.188	TCP	99	5029 → 5020 [PSH, ACK] Seq=25 Ack=41 Win=64256 Len=33 TSval=2648598723 TSecr=591844846
80	21.995544029	192.168.50.188	192.168.50.236	TCP	78	5020 → 5029 [PSH, ACK] Seq=41 Ack=58 Win=65152 Len=12 TSval=591844847 TSecr=2648598723
81	21.995743375	192.168.50.236	192.168.50.188	TCP	78	5029 → 5020 [PSH, ACK] Seq=58 Ack=53 Win=64256 Len=12 TSval=2648598724 TSecr=591844847
82	21.995938039	192.168.50.188	192.168.50.236	TCP	95	5020 → 5029 [PSH, ACK] Seq=53 Ack=70 Win=65152 Len=20 TSval=591844847 TSecr=2648598724
83	21.996148670	192.168.50.236	192.168.50.188	TCP	66	5029 → 5020 [FIN, ACK] Seq=70 Ack=82 Win=64256 Len=0 TSval=2648598724 TSecr=591844847
84	21.996339791	192.168.50.188	192.168.50.236	TCP	66	5020 → 5029 [FIN, ACK] Seq=82 Ack=71 Win=65152 Len=0 TSval=591844848 TSecr=2648598724
85	21.996420673	192.168.50.236	192.168.50.188	TCP	66	5029 → 5020 [ACK] Seq=71 Ack=83 Win=64256 Len=0 TSval=2648598724 TSecr=591844848

Figura 1: Przechwycona komunikacja pomiędzy klientem a serwerem

Na wyżej ukazanej ilustracji przedstawiony został zrzut ekranu z oprogramowania wireshark [4]. Wskazuje on na prostotę, z jaką można przechwycić nieszyfrowane pakiety przesyłane pomiędzy urządzeniami.

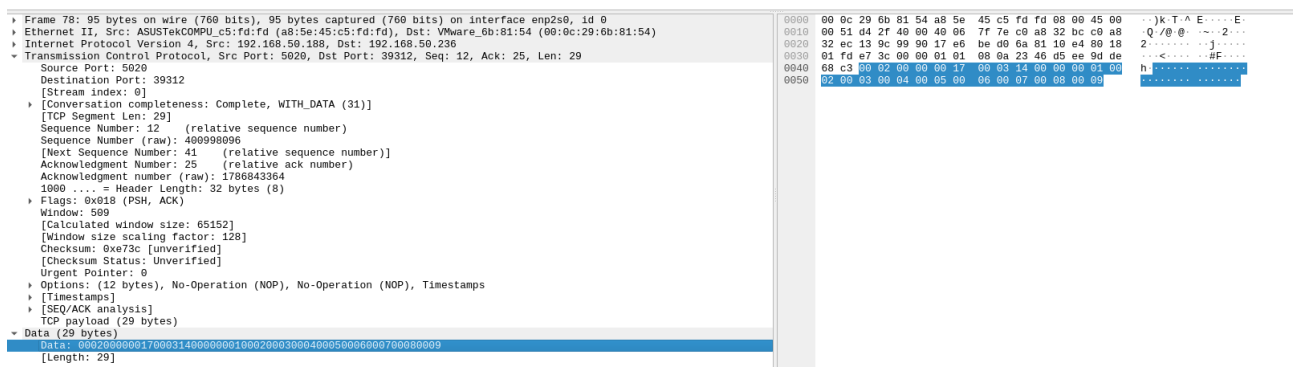
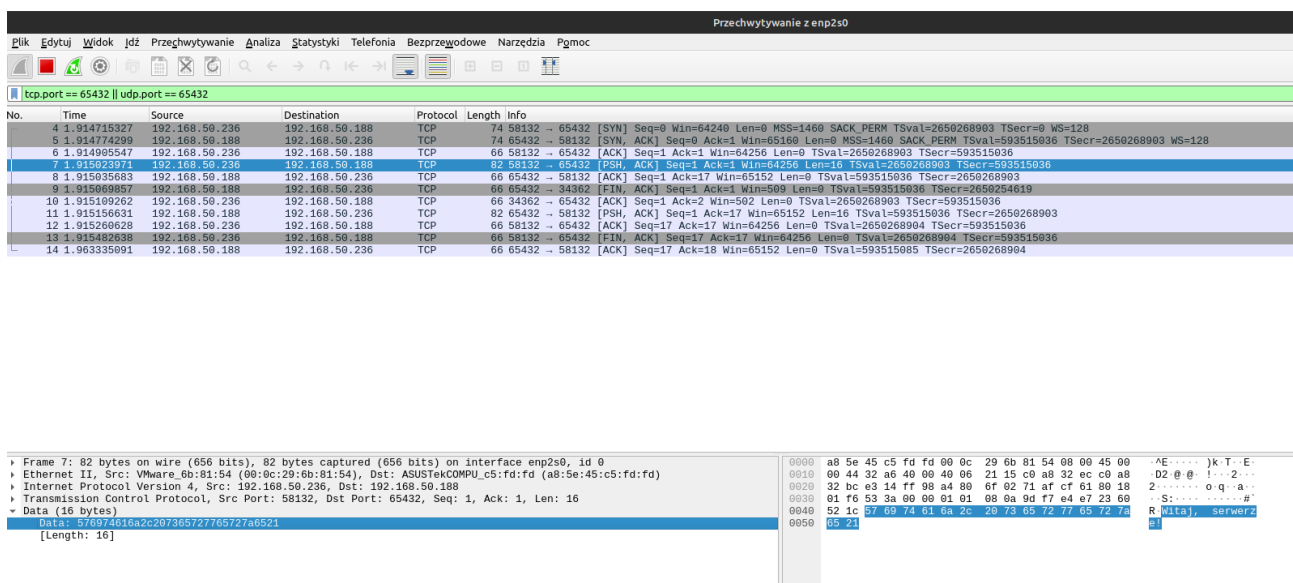


Figura 2: Ustawienie wartości rejestrów

Wyżej ukazana ilustracja ukazuje sekcję danych jednego z przechwyconych pakietów. Ta konkretna ramka ustawia wartości w „rejestrach” serwera na wartości od 0 do 9. Nie występuje tu żadne szyfrowanie, więc dane przesyłane przez sieć mogą zostać przechwycone. Aby zapewnić bezpieczną transmisję bez podatności na modyfikację zawartości rejestrów urządzeń w sieci należałoby zapewnić uwierzytelnianie i szyfrowanie.



4.1.2. Ethernet/IP

W porównaniu do protokołu Modbus, pod względem bezpieczeństwa, Ethernet/IP jest o wiele lepszym wyborem. Zezwala on na większą elastyczność w kontekście zabezpieczeń:

- zastosowanie firewall - stosowanie firewalli na poziomie sieci komputerowej może pomóc w zabezpieczeniu urządzeń Ethernet/IP przed nieautoryzowanym dostępem. Firewallle mogą kontrolować ruch sieciowy na granicy sieci, blokować podejrzane pakiety oraz monitorować i zarządzać połączeniami sieciowymi,
- VPN - wykorzystanie VPN umożliwia bezpieczne połączenie między różnymi lokalizacjami lub urządzeniami poprzez publiczną infrastrukturę sieciową, jaką jest Internet. Poprzez zastosowanie VPN można zapewnić poufność, integralność i autentyczność danych przesyłanych między urządzeniami Ethernet/IP,
- autoryzacja - wdrożenie mechanizmów autoryzacji, takich jak loginy i hasła, na urządzeniach Ethernet/IP może zapobiec nieautoryzowanemu dostępowi do systemów i urządzeń. Jest to podstawowa warstwa zabezpieczeń, która może być stosowana na poziomie aplikacji lub protokołu,
- kontrola dostępu - implementacja kontroli dostępu pozwala na określenie, które urządzenia lub użytkownicy mają prawo do dostępu do określonych zasobów w sieci Ethernet/IP. Jest to szczególnie ważne w przypadku środowisk przemysłowych, gdzie dostęp do krytycznych systemów musi być ściśle kontrolowany,
- szyfrowanie danych - szyfrowanie danych przesyłanych przez sieć Ethernet/IP zapewnia dodatkową warstwę ochrony przed nieautoryzowanym dostępem oraz przechwytywaniem danych. Można stosować różne protokoły szyfrowania, takie jak SSL/TLS, do zabezpieczenia komunikacji między urządzeniami.

Zastosowanie tych zabezpieczeń w sieci Ethernet/IP może pomóc w ochronie urządzeń i danych przemysłowych przed różnego rodzaju zagrożeniami oraz zapewnić stabilność i niezawodność działania systemów przemysłowych.

9	1.915469857	192.168.50.188	192.168.50.236	TCP	66	65432	→	34362	[FIN, ACK]	Seq=1	Ack=1	Win=569	Len=0	TSval=593515036	TSecr=2650268904
10	1.915109262	192.168.50.236	192.168.50.188	TCP	66	34362	→	65432	[ACK]	Seq=1	Ack=2	Win=582	Len=0	TSval=2650268903	TSecr=593515036
11	1.915156631	192.168.50.188	192.168.50.236	TCP	82	65432	→	58132	[PSH, ACK]	Seq=1	Ack=17	Win=65152	Len=10	TSval=593515036	TSecr=2650268903
12	1.915260628	192.168.50.236	192.168.50.188	TCP	66	58132	→	65432	[ACK]	Seq=17	Ack=17	Win=64256	Len=0	TSval=2650268904	TSecr=593515036
13	1.915482636	192.168.50.236	192.168.50.188	TCP	66	58132	→	65432	[FIN, ACK]	Seq=17	Ack=17	Win=64256	Len=0	TSval=2650268904	TSecr=593515036
14	1.963335891	192.168.50.188	192.168.50.236	TCP	66	65432	→	58132	[ACK]	Seq=17	Ack=18	Win=65152	Len=0	TSval=593515085	TSecr=2650268904

<p>Frame 11: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface enp2s0, id 0</p> <p>Ethernet II, Src: ASUSTekCOMPU_c5:fd:fd (a8:5e:45:c5:fd:fd), Dst: VMware_6b:81:54 (08:0c:29:6b:81:54)</p> <p>Internet Protocol Version 4, Src: 192.168.50.188, Dst: 192.168.50.236</p> <p>Transmission Control Protocol, Src Port: 65432, Dst Port: 58132, Seq: 1, Ack: 17, Len: 16</p> <p>Data (16 bytes)</p> <p>Data: 576974616a2c207365727765727a6521</p> <p>[Length: 16]</p>	<pre> 0000 00 0c 29 6b 81 54 a8 5e 45 c5 fd fd 08 00 45 00 ..k.T.A.E....E 0010 00 44 8f 29 40 00 40 06 c4 91 c0 a8 32 bc c0 a8 .D.)@:....2... 0020 32 ec ff 98 e3 14 71 af cf 61 a4 80 6f 12 80 18 2.....q:a.o... 0030 61 fd e7 2f 08 00 01 01 08 6a 23 60 52 1c 9d f7 ..f.....fR... 0040 e4 e7 57 69 74 61 6a 2c 20 73 65 72 77 65 72 7a .Witaj, serwer 0050 65 21 </pre>
--	---

Figura 3: Przechwycone pakiety Ethernet

Na wyżej załączonym obrazku ukazane zostały przechwycone pakiety przesłane po uruchomieniu aplikacji klienta. Dane przesłane pomiędzy klientem a serwerem są łatwe do odczytania, ponieważ komunikacja odbywa się w formie tekstowej, a nie jest zabezpieczona żadnym mechanizmem szyfrowania.

Korzystając z protokołu Ethernet/IP w takim prostym, niezabezpieczonym kształcie, istnieje ryzyko przechwycenia i nieautoryzowanego dostępu do danych przesyłanych między klientem a serwerem. Na przykład, atakujący mógłby przechwycić poufne informacje, takie jak dane produkcyjne, hasła lub inne poufne informacje przesyłane w jawnym tekście.

W związku z tym, przy wyborze protokołu komunikacyjnego dla systemów przemysłowych, szczególnie ważne jest zastosowanie odpowiednich mechanizmów zabezpieczeń, takich jak szyfrowanie. Szyfrowanie danych umożliwia zakodowanie przesyłanych informacji w taki sposób, żeby były one nieczytelne dla osób nieuprawnionych.

Implementacja szyfrowania w komunikacji opartej na protokole Ethernet/IP może zapewnić dodatkową warstwę ochrony danych przesyłanych między klientem a serwerem. Szyfrowanie może być realizowane poprzez protokoły takie jak SSL/TLS, które zapewniają poufność, integralność i autentyczność danych.

Dzięki zastosowaniu mechanizmów szyfrowania, nawet jeśli pakiet zostanie przechwycony przez atakującego, dane w nim zawarte będą nieczytelne, co skutecznie zabezpieczy informacje przed nieautoryzowanym dostępem i uchroni system przed potencjalnymi zagrożeniami.

Wniosek ten podkreśla wagę stosowania odpowiednich zabezpieczeń w komunikacji przemysłowej, zwłaszcza w środowiskach, gdzie bezpieczeństwo danych jest kluczowe dla działania systemu oraz dla ochrony przed potencjalnymi zagrożeniami i atakami.

5. Podsumowanie

Modbus TCP/IP:

1. Zastosowanie:

- Modbus TCP/IP jest popularnym protokołem komunikacyjnym stosowanym w przemyśle do komunikacji między urządzeniami w systemach automatyzacji, takich jak PLC (programowalne kontrolery logiczne), czujniki, regulatory, itp.
- Jest stosowany w różnych aplikacjach, od monitorowania i sterowania procesami przemysłowymi po zarządzanie danymi.

2. Zagrożenia:

- **Brak szyfrowania:** W standardowym Modbus TCP/IP dane przesyłane są w formie tekstu, co oznacza, że są one podatne na przechwycenie i odczytanie przez atakujących. Brak szyfrowania sprawia, że dane mogą być łatwo podsłuchiwane.
- **Brak autoryzacji:** Modbus TCP/IP nie zapewnia wbudowanych mechanizmów autoryzacji, co oznacza, że potencjalnie każde urządzenie w sieci może uzyskać dostęp do danych i komunikować się z innymi urządzeniami w sieci.
- **Brak integralności danych:** Protokół Modbus TCP/IP nie zapewnia mechanizmów zapobiegających zmianom w danych w trakcie transmisji, co oznacza, że dane mogą zostać podmienione lub zmodyfikowane przez atakującego.

Ethernet/IP:

1. Zastosowanie:

- Ethernet/IP jest powszechnie stosowanym protokołem komunikacyjnym w przemyśle, umożliwiającym komunikację między różnymi urządzeniami w sieciach przemysłowych.
- Jest wykorzystywany w szerokim zakresie zastosowań, od sterowania procesami produkcyjnymi po monitorowanie stanu urządzeń.

2. Zagrożenia:

- **Ataki typu DoS/DDoS:** Ethernet/IP, podobnie jak inne protokoły sieciowe, jest podatny na ataki typu DoS (Denial of Service) i DDoS (Distributed Denial of Service), które mogą zakłócić działanie systemów przemysłowych poprzez przepełnienie sieci lub urządzenia nadmierną ilością żądań.
- **Ataki MITM:** Atak Man-in-the-Middle (MITM) może być używany do przechwycenia i modyfikacji danych przesyłanych między urządzeniami w sieci Ethernet/IP.
- **Nieautoryzowany dostęp:** Bez odpowiednich zabezpieczeń, nieautoryzowane urządzenia lub użytkownicy mogą uzyskać dostęp do sieci Ethernet/IP i komunikować się z urządzeniami, co może prowadzić do naruszenia poufności danych lub zmiany stanu urządzeń.

Zarówno Modbus TCP/IP, jak i Ethernet/IP są powszechnie stosowane w przemyśle do komunikacji między urządzeniami, jednakże oba protokoły mogą być podatne na różne rodzaje zagrożeń związanych z bezpieczeństwem sieciowym.

Aby zabezpieczyć systemy przemysłowe oparte na tych protokołach, konieczne jest stosowanie odpowiednich środków bezpieczeństwa, takich jak szyfrowanie danych, autoryzacja, kontrola dostępu oraz monitorowanie i reagowanie na niepokojące zachowania w sieci.

Regularne audyty bezpieczeństwa oraz aktualizacje oprogramowania mogą również pomóc w zapewnieniu ochrony przed zagrożeniami w środowiskach opartych na Modbus TCP/IP i Ethernet/IP.

Źródła

- [1]. <https://pl.wikipedia.org/wiki/Modbus>
- [2]. <https://en.wikipedia.org/wiki/EtherNet/IP>
- [3]. <https://pl.wikipedia.org/wiki/Ethernet>
- [4]. <https://www.wireshark.org/download.html>
- [5]. https://github.com/DevxMike/spoleczenstwo_informacyjne_projekt
- [6]. <https://en.wikipedia.org/wiki/Profinet>
- [7]. <https://en.wikipedia.org/wiki/CANopen>
- [8]. <https://pl.wikipedia.org/wiki/Profibus>
- [9]. <https://en.wikipedia.org/wiki/DeviceNet>