

Author : Muhammad Mardiansyah

Simulasi Man-in-the-Middle (MitM) pada Komunikasi Modbus TCP/IP dengan Wireshark

1. Judul

Analisis Lalu Lintas Modbus TCP/IP Menggunakan Wireshark dalam Simulasi Man-in-the-Middle

2. Latar Belakang

Dalam sistem kontrol industri (Industrial Control Systems/ICS) seperti SCADA, protokol komunikasi seperti Modbus TCP masih sering digunakan. Sayangnya, Modbus TCP tidak memiliki fitur keamanan seperti enkripsi atau autentikasi. Hal ini memungkinkan pihak ketiga untuk dengan mudah memantau atau bahkan memodifikasi lalu lintas data. Dalam proyek ini, dilakukan simulasi Man-in-the-Middle (MitM) secara pasif menggunakan Wireshark untuk membuktikan kerentanan tersebut.

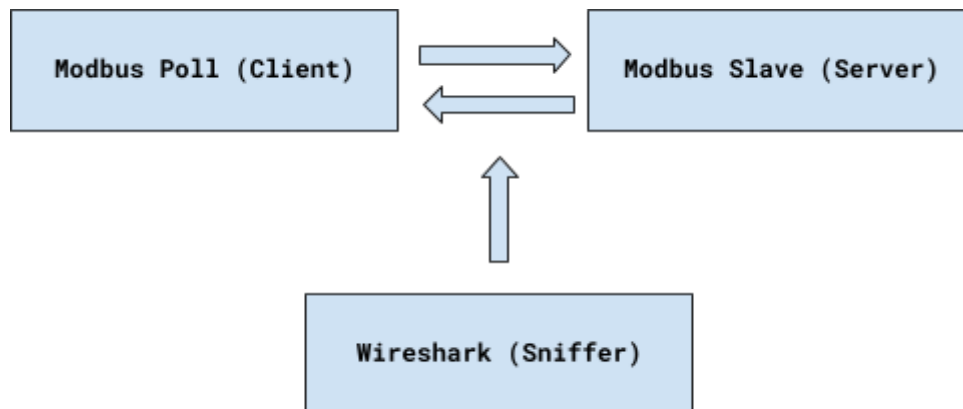
3. Tujuan

- Mensimulasikan komunikasi Modbus TCP antara client dan server.
 - Melakukan sniffing lalu lintas menggunakan Wireshark.
 - Mengidentifikasi isi paket: function code, register address, nilai register.
 - Menganalisis potensi risiko dari protokol yang tidak aman.
-

4. Tools yang Digunakan

Tool	Fungsi
Modbus Slave	Simulator Modbus TCP Server
Modbus Poll	Simulator Modbus TCP Client
Wireshark	Alat sniffing dan analisis protokol

5. Topologi Simulasi



Seluruh simulasi dilakukan di dalam satu komputer (localhost: 127.0.0.1)

6. Langkah-Langkah Simulasi

a. Menjalankan Server (Modbus Slave):

- Buka aplikasi **Modbus Slave**.
- Pilih koneksi: **Modbus TCP/IP Server**.
- Port: **502**.
- Isi name dan address pada register.
- Jalankan dan isi beberapa Holding Register (misalnya alamat 0–9 dengan nilai acak)

b. Menjalankan Client (Modbus Poll):

- Buka aplikasi **Modbus Poll**.
- Koneksi ke 127.0.0.1 : 502, Slave ID: 1
- Lakukan polling:
 - Function: 03 (Read Holding Registers)
 - Start address: 0
 - Quantity: 10

c. Sniffing dengan Wireshark:

- Jalankan Wireshark
- Pilih interface Loopback

Filter:

```
tcp.port == 502
```

7. Hasil Observasi

Contoh paket hasil tangkapan Wireshark:

No	Time	Source	Destination	Protocol	Info
3	0.024652	127.0.0.1	127.0.0.1	Modbus/TCP	Response: Trans: 233; Unit: 1, Func: 3: Read Holding Registers

```
▼ Modbus
  .000 0011 = Function Code: Read Holding Registers (3)
  [Request Frame: 1]
  [Time from request: 0.015526000 seconds]
  Byte Count: 20
  > Register 0 (UINT16): 7
  > Register 1 (UINT16): 8
  > Register 2 (UINT16): 9
  > Register 3 (UINT16): 10
  > Register 4 (UINT16): 11
  > Register 5 (UINT16): 12
  > Register 6 (UINT16): 13
```

Berdasarkan analisis paket:

- **Function Code:** 0x03 → Read Holding Registerr
- **Unit ID:** 1
- **Transaction ID:** 233
- **Data:** nilai register dibalas secara plaintext, misal :
 - Register 0 : 7
 - Register 1 : 8
 - Register 2 : 9
- Komunikasi ini **terbaca sepenuhnya** oleh pihak ketiga tanpa perlu autentikasi

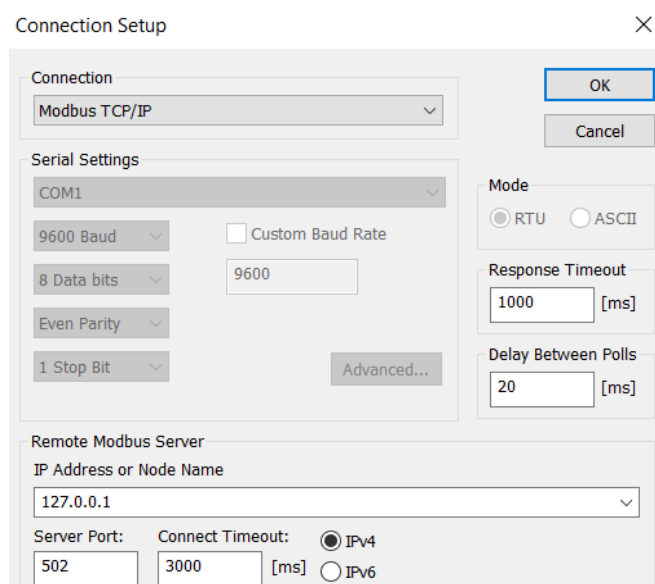
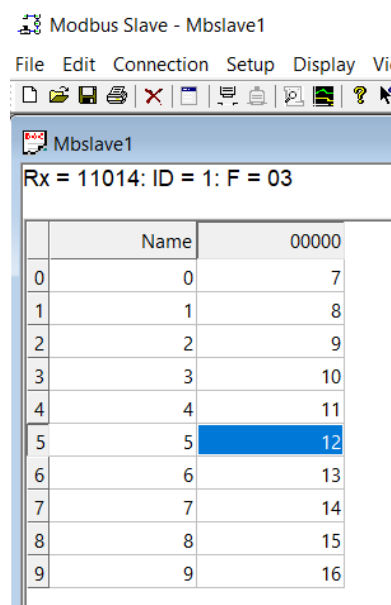
8. Analisis Keamanan

Aspek	Temuan
Enkripsi	✗ Tidak ada, data dalam plaintext
Autentikasi	✗ Tidak diterapkan antara client dan server
Potensi MitM	✓ Sniffing berhasil tanpa hambatan
Risiko di dunia nyata	⚠ Sangat tinggi pada sistem ICS kritikal

9. Kesimpulan

Simulasi ini berhasil menunjukkan bahwa komunikasi Modbus TCP sangat rentan terhadap serangan Man-in-the-Middle. Siapa pun yang dapat mengakses jaringan (bahkan secara lokal) dapat memantau atau menganalisis isi komunikasi tanpa kesulitan. Hal ini membuktikan bahwa implementasi Modbus TCP harus disertai pengamanan tambahan seperti VPN, enkripsi TLS, isolasi jaringan, atau migrasi ke protokol yang lebih aman.

10. Dokumentasi Tambahan



Modbus Poll - Mbpoll2

File Edit Connection Setup Functions Display View

05 06 15 16 17 22 2

Mbpoll2

Tx = 0: Err = 0: ID = 1: F = 03: SR = 1000ms

No connection

	Name	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0

Read/Write Definition



Slave ID:

Function:

Address mode
☒ Dec ☐ Hex

Address: PLC address = 40001

Quantity:

Scan Rate: [ms]

Disable
☐ Read/Write Disabled
☐ Disable on error

View
 Rows
☒ 10 ☐ 20 ☐ 50 ☐ 100 ☐ Fit to Quantity
☐ Hide Name Columns ☐ PLC Addresses (Base 1)
☐ Address in Cell ☐ Enron/Daniel Mode

Request
 RTU

 ASCII

Connection Setup

✕

Connection

Modbus TCP/IP

Serial Settings

COM1

9600 Baud

8 Data bits

Even Parity

1 Stop Bit

Advanced...

☐ Custom Baud Rate

9600

Mode

☒ RTU
 ☐ ASCII

Response Timeout

1000 [ms]

Delay Between Polls

20 [ms]

Remote Modbus Server

IP Address or Node Name

127.0.0.1

Server Port:

502

Connect Timeout:

3000 [ms]










☒ IPv4
 ☐ IPv6

OK

Cancel

Modbus Poll - Mbpoll2

File Edit Connection Setup Functions Display \


05 06 15 16 17 2

Mbpoll2

Tx = 8: Err = 0: ID = 1: F = 03: SR = 1000ms

	Name	00000
0		7
1		8
2		9
3		10
4		11
5		12
6		13
7		14

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	Modbus...	56	Query: Trans: 10622; Unit: 1, Func: 3: Read Holding Registers
2	0.000105	127.0.0.1	127.0.0.1	TCP	44	502 → 49785 [ACK] Seq=1 Ack=13 Win=10233 Len=0
3	0.015526	127.0.0.1	127.0.0.1	Modbus...	73	Response: Trans: 10622; Unit: 1, Func: 3: Read Holding Registers
4	0.015623	127.0.0.1	127.0.0.1	TCP	44	49785 → 502 [ACK] Seq=13 Ack=30 Win=10232 Len=0
5	1.009885	127.0.0.1	127.0.0.1	Modbus...	56	Query: Trans: 10623; Unit: 1, Func: 3: Read Holding Registers
6	1.009972	127.0.0.1	127.0.0.1	TCP	44	502 → 49785 [ACK] Seq=30 Ack=25 Win=10233 Len=0
7	1.029744	127.0.0.1	127.0.0.1	Modbus...	73	Response: Trans: 10623; Unit: 1, Func: 3: Read Holding Registers
8	1.029860	127.0.0.1	127.0.0.1	TCP	44	49785 → 502 [ACK] Seq=25 Ack=59 Win=10232 Len=0
9	2.017295	127.0.0.1	127.0.0.1	Modbus...	56	Query: Trans: 10624; Unit: 1, Func: 3: Read Holding Registers
10	2.017348	127.0.0.1	127.0.0.1	TCP	44	502 → 49785 [ACK] Seq=59 Ack=37 Win=10233 Len=0
11	2.046713	127.0.0.1	127.0.0.1	Modbus...	73	Response: Trans: 10624; Unit: 1, Func: 3: Read Holding Registers
12	2.046781	127.0.0.1	127.0.0.1	TCP	44	49785 → 502 [ACK] Seq=37 Ack=88 Win=10232 Len=0
13	2.114465	192.168.56.1	239.255.255.250	SSDP	133	M-SEARCH * HTTP/1.1
14	2.114592	127.0.0.1	239.255.255.250	SSDP	133	M-SEARCH * HTTP/1.1
15	2.114823	192.168.120.5	239.255.255.250	SSDP	133	M-SEARCH * HTTP/1.1
16	3.029961	127.0.0.1	127.0.0.1	Modbus...	56	Query: Trans: 10625; Unit: 1, Func: 3: Read Holding Registers
17	3.030033	127.0.0.1	127.0.0.1	TCP	44	502 → 49785 [ACK] Seq=88 Ack=49 Win=10232 Len=0
18	3.044872	127.0.0.1	127.0.0.1	Modbus...	73	Response: Trans: 10625; Unit: 1, Func: 3: Read Holding Registers
19	3.044925	127.0.0.1	127.0.0.1	TCP	44	49785 → 502 [ACK] Seq=49 Ack=117 Win=10232 Len=0
20	4.038150	127.0.0.1	127.0.0.1	Modbus...	56	Query: Trans: 10626; Unit: 1, Func: 3: Read Holding Registers
21	4.038219	127.0.0.1	127.0.0.1	TCP	44	502 → 49785 [ACK] Seq=117 Ack=61 Win=10232 Len=0
22	4.053684	127.0.0.1	127.0.0.1	Modbus...	73	Response: Trans: 10626; Unit: 1, Func: 3: Read Holding Registers
23	4.053759	127.0.0.1	127.0.0.1	TCP	44	49785 → 502 [ACK] Seq=61 Ack=146 Win=10231 Len=0
24	5.052003	127.0.0.1	127.0.0.1	Modbus...	56	Query: Trans: 10627; Unit: 1, Func: 3: Read Holding Registers
25	5.052056	127.0.0.1	127.0.0.1	TCP	44	502 → 49785 [ACK] Seq=146 Ack=73 Win=10232 Len=0
26	5.075974	127.0.0.1	127.0.0.1	Modbus...	73	Response: Trans: 10627; Unit: 1, Func: 3: Read Holding Registers
27	5.076060	127.0.0.1	127.0.0.1	TCP	44	49785 → 502 [ACK] Seq=73 Ack=175 Win=10231 Len=0

 tcp.port == 502

Wireshark · Packet 3 · Adapter for loopback traffic capture

```

> Frame 3: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface \Device\NPF_{Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 502, Dst Port: 49785, Seq: 1, Ack: 13, Len: 29
> Modbus/TCP
▼ Modbus
  .000 0011 = Function Code: Read Holding Registers (3)
  [Request Frame: 1]
  [Time from request: 0.015526000 seconds]
  Byte Count: 20
  > Register 0 (UINT16): 7
  > Register 1 (UINT16): 8
  > Register 2 (UINT16): 9
  > Register 3 (UINT16): 10
  > Register 4 (UINT16): 11
  > Register 5 (UINT16): 12
  > Register 6 (UINT16): 13
  > Register 7 (UINT16): 14
  > Register 8 (UINT16): 15
  > Register 9 (UINT16): 16

0000 02 00 00 00 45 00 00 45 0c b2 40 00 80 06 00 00  ....E..E..@....
0010 7f 00 00 01 7f 00 00 01 01 f6 c2 79 2a a1 fa 47  .......y*...G
0020 6b af fc cf 50 18 27 f9 86 43 00 00 29 7e 00 00  k...P-'.-(C...)~
0030 00 17 01 03 14 00 07 00 08 00 09 00 0a 00 0b 00  ....
0040 0c 00 0d 00 0e 00 0f 00 10

```