ICS Protocols Anomaly Behavior Analysis: Modbus

Part 1

Modbus

Original Implementation

- Modbus was introduced in 1979 by the company “Modicon”
- Master-slave protocol
  - A slave is typically a PLC
  - Masters are programming panels or host computers
- Only masters can initiate a message
- Messages issued by the master are called *Query*
- Modbus has no multicast capability

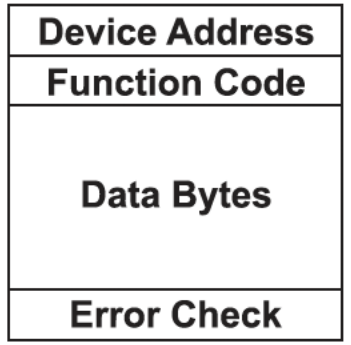

Figure 1. Simplified Modbus message format.

Modbus Serial

- **ASCII**
  - Human readable
  - Timing between characters is not an issue

Start of Frame	Device Address	Function Code	Data	LRC Check	End of Frame
1 character (:)	2 characters	2 characters	n characters	2 characters	2 characters (CRLF)

Figure 3. ASCII framing of a Modbus message.

- **RTU** (Remote Terminal Unit)
  - Not human readable
  - Messages are compact and more efficient to send

Start of Frame	Device Address	Function Code	Data	CRC Check	End of Frame
4 character times	8 bits	8 bits	n x 8 bits	16 bits	4 character times

Figure 4. RTU framing of a Modbus message.

Register Map

- The register map is a structured set of data points
- Devices use this map to organize and store information that can be read from or written to by other devices on the network

I/O Range	Description
00001 – 10000	Read/Write discrete output or coils
10001 – 20000	Read discrete inputs
30001 – 40000	Read input registers – 16-bit registers such as analog inputs
40001 – 50000	Read/Write holding registers – 16-bit storage or I/O

Figure 5. Typical Modbus Register Map.

Function Codes

- Function codes are a set of commands or requests used by the master to request specific actions or data from a slave

Code	1/16-bit	Description	I/O Range
01	1-bit	Read coils	00001 – 10000
02	1-bit	Read contacts	10001 – 20000
05	1-bit	Write a single coil	00001 – 10000
15	1-bit	Write mulitple coils	00001 – 10000
03	16-bit	Read holding registers	40001 – 50000
04	16-bit	Read input registers	30001 – 40000
06	16-bit	Write single register	40001 – 50000
16	16-bit	Write mulitple registers	40001 – 50000
22	16-bit	Mask write register	40001 – 50000
23	16-bit	Read/write mulitple registers	40001 – 50000
24	16-bit	Read FIFO queue	40001 – 50000

Figure 6. Data access function codes.

Modbus Serial (Updated)

- Operates over a physical connection
- Limited data transfer speeds
- Only supports communication over short distances
- Provides good data integrity with CRC

Layer	ISO/OSI Function	Modbus Function
7	Application	Modbus Application Protocol
3–6	Various	Null
2	Data Link	Modbus Serial Line Protocol
1	Physical	EIA-485, EIA-232C

Table 1. Modbus over Serial Line uses a three-layer model.

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 bytes(s)	2 bytes CRC Low | CRC Hi

Figure 2. RTU framing is more condensed than ASCII framing.

Modbus TCP

- Based on Ethernet
- Client-server -> Master-slave
- Message exchange
  - A **request** is sent by the client to initiate a transaction
  - An **indication** is sent by the server to confirm that a request has been received
  - A **response** is sent by the server to comply with the client's request
  - A **confirmation** is sent by the client to acknowledge receipt of the response

Layer	ISO/OSI Function	Modbus Function
5,6,7	Application	Modbus Application Protocol
4	Transport	Transmission Control Protocol
3	Network	Internet Protocol
2	Data Link	IEEE 802.3
1	Physical	IEEE 802.3

Table 2. Modbus TCP uses a five-layer Internet model.

Modbus Vulnerabilities

- Complexity and real-time constraints
- Lack of confidentiality and integrity
- Authentication deficiencies
- Absence of anti-repudiation and anti-reply techniques
- These vulnerabilities can be exploited with the following key attacks
  - Unauthorized command execution
  - Denial-of-service
  - Man-in-the-middle
  - Reply attacks

Modbus attacks

- Attacks on Modbus systems and networks can be categorized into three main groups:
  - Attacks exploiting the Modbus protocol specifications.
  - Attacks targeting vendor implementations of the Modbus protocols.
  - Attacks focused on the support infrastructure, including IT, networking, and telecommunications assets.

- Primary targets include the master, field devices, serial communication links, network communication paths, and messages.

Modbus Secure Implementation

- Proposed by *Nai Fovino, Carcano, Masera & Trombetta*
- Aims to fulfill specific security requirements
  - Integrity
  - Authentication
  - Non-repudiation
  - Reply protection

- Implementation of NTP timestamps for evaluation of packets

Modbus Secure Implementation

- Introduction of Modbus secure gateway

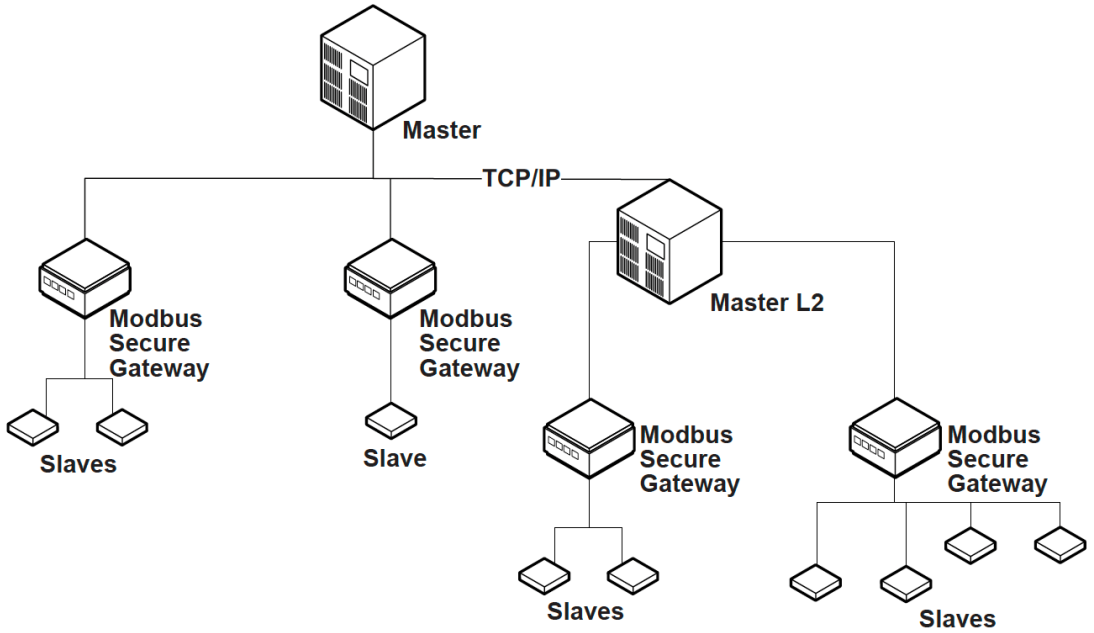

Figure 3. Modbus Secure Gateway.

Modbus Secure Implementation

- When the Modbus Secure Gateway receives a packet on the process network interface:
  - It accepts only authenticated Secure Modbus TCP traffic from allowed masters
  - It extracts the Modbus packet from the Secure Modbus packet
  - It forwards the packet to the appropriate slave using the related point-to-point (serial or TCP) link

Modbus Secure Implementation

- When the Modbus Secure Gateway receives a packet on one of the point-to-point links connected with a slave:
  - It creates a Secure Modbus packet containing the received original Modbus packet
  - It signs the packet digest with the private key associated with the slave
  - It forwards the new packet to the appropriate master through its process network interface

Modbus Secure Implementation

- Steps involved in sending and verifying a Secure Modbus request message:
  - The master creates a valid Modbus request with a timestamp and the serial slave address
  - The master computes the digest of the Modbus request, encrypts the digest with its private key, and sends the request along with the encrypted digest to a slave or to the Modbus Secure Gateway
  - The slave or the Modbus Secure Gateway verifies that the Modbus request is genuine using the master's public key
- After verifying that the request is genuine, the Modbus Secure Gateway reads the unit identifier in the MBAP header and sends the Modbus request to the addressed slave

Modbus Secure Implementation

- Secure Modbus module
  - Modbus stream builder
  - RSA encryption/decryption unit
  - SHA2 validator
  - Modbus ADU builder/reader
  - Timestamp analyzer

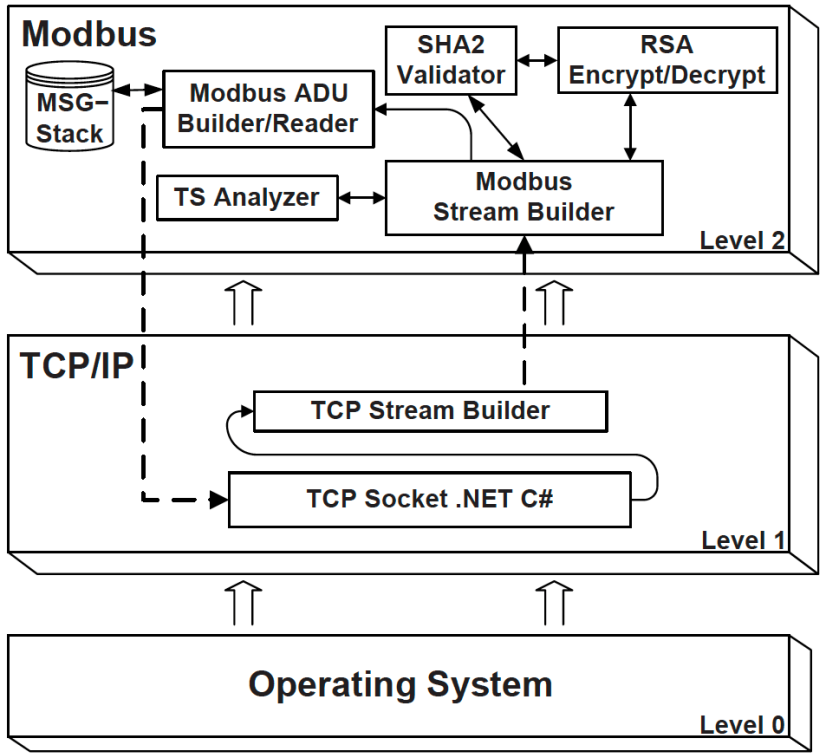

Figure 4. Secure Modbus module.

References

- Thomas, George. "Introduction to the modbus protocol." The Extension 9.4 (2008): 1-4.
- Fovino, I. N., Carcano, A., Masera, M., & Trombetta, A. (2009). Design and implementation of a secure modbus protocol. In Critical Infrastructure Protection III: Third Annual IFIP WG 11.10 International Conference on Critical Infrastructure Protection, Hanover, New Hampshire, USA, March 23-25, 2009, Revised Selected Papers 3 (pp. 83-96). Springer Berlin Heidelberg.
- Huitsing, P., Chandia, R., Papa, M., & Shenoi, S. (2008). Attack taxonomies for the Modbus protocols. International Journal of Critical Infrastructure Protection, 1, 37-44.