Wheels on the Modbus - Attacking ModbusTCP Communications

Abubakar Sadiq Mohammed
mohammedas@cardiff.ac.uk
Cardiff University
Cardiff, United Kingdom

Neetesh Saxena
Cardiff University
Cardiff, United Kingdom
saxenan4@cardiff.ac.uk

Omer Rana
Cardiff University
Cardiff, United Kingdom
ranaof@cardiff.ac.uk

ABSTRACT

Industrial Cyber-Physical Systems (ICPS) make significant use of Supervisory Control and Data Acquisition (SCADA) for control. Such SCADA systems are known to utilise insecure communication protocols such as Modbus, DNP3 and OPC DA. This leads to increased cyber risks faced in critical infrastructures, as these protocols allow threat actors to mount attacks like Denial of Service (DoS). We present a novel field flooding attack, compromising the structure of the ModbusTCP packet and disrupting a controller's interpretation of the commands sent to it. This can disrupt the ability of an operator to control hazardous operations leading to potentially unsafe scenarios.

CCS CONCEPTS

• **Security and privacy → Denial-of-service attacks**.

KEYWORDS

Modbus, cybersecurity, Industrial Cyber-Physical Systems

ACM Reference Format:
Abubakar Sadiq Mohammed, Neetesh Saxena, and Omer Rana. 2022. Wheels on the Modbus - Attacking ModbusTCP Communications. In *WiSec '22: 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks, May 16–May 19, 2022, San Antonio, Texas, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/xxxxxxx.xxxxxxx

1 INTRODUCTION

The Modbus protocol and its variants are widely used communications protocols for Supervisory Control and Data Acquisition (SCADA) in the oil and gas industry, especially for pipeline and monitoring remote offshore operations [1]. The ModbusTCP variant is commonly used in Industrial Cyber-Physical systems (ICPS) communications utilised in controlling hazardous operations. This has introduced cybersecurity risks in critical infrastructure communications mainly due insecurities of the Modbus protocol [4].

We focus on the ModbusTCP protocol and its vulnerabilities, which have the potential to cross over from a cyber attack incident to real life-threatening physical impact. ModbusTCP does not utilise any authentication or access control mechanisms, which allows attackers to perform various attacks such as Denial of Service (DoS), Man-in-the-Middle (MitM) and unauthorised access [4].

Numerous studies on ModbusTCP security vulnerabilities have been carried out, because of its widespread adoption in SCADA communications. Parian et al. [3] exploited the ModbusTCP protocol using a MitM attack in a virtualised experimental setup. In particular, they demonstrated how a legitimate Modbus request could be manipulated using the Modbus server simulation tool ModbusPal. Similarly, Satyanarayana et al.[5] examined the vulnerability of ModbusTCP to false command injection, false access injection and replay attacks. Their proposed detection technique involved incorporating time stamps and sequence numbers in Modbus communication. The attack described in this paper demonstrates the ability to bypass these detection techniques. Morris et al. [2] describe rules that could be combined with popular signature-based Intrusion Detection Systems (IDS) (e.g. Snort) to prevent exploitation of the Modbus protocol.

Contributions: We introduce a novel attack on the ModbusTCP protocol, taking advantage of its packet memory structure to perform a field flooding attack capable of overflowing the memory bank allocated in the Programmable Logic Controller (PLC) for Modbus operations. We also demonstrate the results of this attack on a real industrial testbed mapping the impact on the Mitre ATT&CK framework.

2 ATTACKING THE MODBUSTCP PROTOCOL

This section presents the basic structure of a ModbusTCP packet, experimental setup and tools used in the attack development process. We also describe attacks carried out on the testbed to disrupt the communication between the Human Machine Interface (HMI) and the PLC.

The ModbusTCP protocol operates in a client-server model [3], with requests made using Function Codes (FC). Figure 1 shows the basic structure and size allocated to each header. The Modbus Application Data Unit (ADU) has a total size of 260bytes. This is shared by the Modbus Application (MBAP) header and the Protocol Data Unit (PDU) in the order of 7bytes and 253bytes respectively.

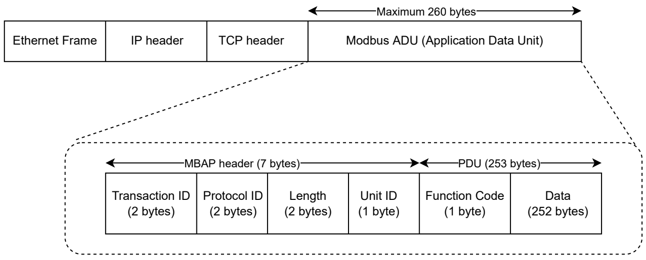

Figure 1: ModbusTCP packet structure

WiSec '22, May 16–May 19, 2022, San Antonio, Texas, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/21/06…$15.00
https://doi.org/10.1145/xxxxxxx.xxxxxxx

2.1 Experimental Setup and Tools Used

The experiment testbed comprised a Siemens Logo PLC, temperature and humidity sensors. The temperature and humidity values are constantly read and displayed in real-time on the HMI. The sensors are both hard wired to the Siemens Logo PLC, while the PLC communicates the values in real time to the HMI using ModbusTCP. The temperature value is stored in a holding register while the humidity value is stored in an input register. The HMI periodically polls the Siemens Logo PLC for the temperature and humidity values using the Modbus function code 0x03 (read holding registers) and 0x04 (read input registers) respectively. The tools used in these attacks were `scapy` and `wireshark`.

2.2 Description of Attacks

We used `scapy` to sniff network traffic between the HMI and PLC that was analysed with `wireshark`. ModbusTCP communication between HMI and PLC is usually in a continuous loop. In our case, this was a loop comprising of two queries (HMI polling PLC for data/status) and two responses (PLC sending requested data/ status to HMI). Each query (from HMI) is followed by a corresponding response (from PLC) and an acknowledgement of receipt of data by the PLC. The key metric is the communication time, approximately 7ms between a query-response-ack loop and 100ms between loops – 100ms is the time period utilised to craft and inject our malicious packets into the network. To craft a packet that will be accepted by the PLC, it needs to:

- conform with the ModbusTCP standard format (contain function code, transaction and protocol identifiers, unit ID, length and register starting address); and
- utilise sequence (SEQ) and acknowledgement (ACK) numbers in previous packet to use as its own SEQ and ACK numbers.

We also altered the number of fields in the PDU header to craft malicious packets. Recall that the maximum memory allocated for ModbusTCP ADU header is 260bytes. By altering the length field in the MBAP header and increasing the number fields in the PDU layer, this limit is exceeded which can potentially disrupt the communication between the HMI and PLC. The following experiments were carried out with varying parameters:

- Modify ModbusTCP write packet (FC 06/16) with increased length field in MBAP header and inject;
- modify ModbusTCP write packet (FC 06/16) with 2 additional fields (4 bytes) in PDU layer and inject.

3 RESULTS

Four malicious packets were successfully injected with each having two additional fields "Word Count" and "Register Value", within the PDU. This corrupted the TCP session, with the PLC responding to Modbus queries from HMI with RST ACK packets in an attempt to reset the TCP session. The field flooding attack effectively forced the PLC into listen-only mode for approximately 7minutes, culminating in a DoS scenario as shown in Figure 2. In oil and gas production platforms, where SCADA is used to control heating and separation of volatile hydrocarbons, operators monitor and ensure safe operations via HMI equipped with override functions for emergency shutdowns. This attack has the potential to impair process control leading to pipeline explosions, loss of lives and damage to the environment.

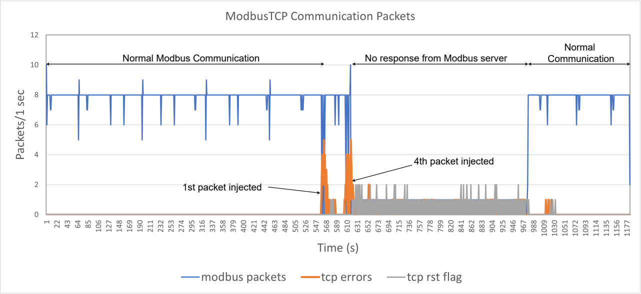

Figure 2: Disruption of Communication loop as result of Modbus field flooding attack - a DoS scenario

Mapping the field flooding attack on the Mitre ATT&CK for ICS framework, the following tactics, techniques and impact were identified: **ATT&CK Tactics/Techniques (Code):** 1. Execution/Command-Line Interface (T0807), Scripting (T0853); 2. Discovery/Network Sniffing (T0842); 3. Inhibit Response Function/Block Reporting Message (T0804), Denial of Service (T0814); 4. Impair Process Control/Modify Parameter (T0836), Unauthorized Command Message (T0855); 5. Impact/Denial of View (T0815).

4 CONCLUSION

Previous work has focused on protecting the ModbusTCP message by ensuring the size allocated to a particular field in the MBAP and PDU headers are within set limits. In this study, we were able to demonstrate a novel field flooding attack where the consideration was keeping the fields within their bytes limit, but increasing the number fields by 2, resulting in an additional 4 bytes of fields to the PDU header. This caused a DoS for a significant period that has the potential, in oil and gas operations, to lead to unsafe conditions like pipeline explosions and damage to the environment due to the hazardous nature of hydrocarbons. In future work, we will carry out sensitivity analysis on variants of this attack to gain better insight on the PDU field flooding memory violation, then design an Intrusion Prevention System (IPS) to better protect the protocol from cyberattacks.

REFERENCES

[1] Peter Huitsing, Rodrigo Chandia, Mauricio Papa, and Sujeet Shenoi. 2008. Attack taxonomies for the Modbus protocols. *International Journal of Critical Infrastructure Protection* 1 (2008), 37–44.

[2] Thomas H Morris, Bryan A Jones, Rayford B Vaughn, and Yoginder S Dandass. 2013. Deterministic intrusion detection rules for MODBUS protocols. In *2013 46th Hawaii International Conference on System Sciences*. IEEE, 1773–1781.

[3] Christopher Parian, Terry Guldimann, and Sajal Bhatia. 2020. Fooling the master: Exploiting weaknesses in the Modbus protocol. *Procedia Computer Science* 171 (2020), 2453–2458.

[4] Panagiotis Radoglou-Grammatikis, Ilias Siniosoglou, Thanasis Liatifis, Anastasios Kourouniadis, Konstantinos Rompolos, and Panagiotis Sarigiannidis. 2020. Implementation and detection of modbus cyberattacks. In *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, 1–4.

[5] Penke Satyanarayana et al. 2021. Detection and Blocking of Replay, False Command, and False Access Injection Commands in SCADA Systems with Modbus Protocol. *Security and Communication Networks* 2021 (2021).