UID Detection and Enumeration Function on Modbus

Overview

In this report, we will focus on modbus, the most widely used SCADA/ICS protocol. We will use Honeypot as the target.

This Report provides:

- Disclosure of exploit runs on ModBus protocol.
- Simulating an attack scenario against Scada Honeypot (Conpot)
- Modbus attack diagnostic functions in Wireshark
- Writing IDS rules to detect attack

Vulnerability Overview

Modbus protocol defines several function codes for accessing Modbus registers. There are four different data blocks defined by Modbus, and the addresses or register numbers in each of those overlap. Therefore, a complete definition of where to find a piece of data requires both the address (or register number) and function code (or register type). In other words, it consists of the code for the function requested by the Master to send a message to the Slave. In the module we use, it determines which services are running in these function codes. For ID detection, we can detect the device ID by applying a brute force attack.

Explotability This vulnerability can be exploited both remotely and locally.

Existence of Exploit It is an unmodifiable vulnerability in the Modbus protocol.

Difficulty An attacker with a medium skill would be able to exploit this vulnerability.

Affected Produtcs

All devices using the modbus protocol

Impact

Function codes detected by using this module will reveal many things that can be done in the system. The most important of these is that the system stops or breaks down beyond repair. The stoppage of a smart factory leads to huge monetary losses.

Exploitation of Vulnerability

It is a ready-made tool that we can perform operations such as scanning, denial of service and vulnerability scanning for ICS/Scada systems in the Smod tool. In order to exploit the vulnerability we need to download Smod After downloading smod we can run it by typing;

```
cd smod
sudo python smod.py
```

We can see the ModBus Frameworks’ modules by typing;

```
show modules
use modbus/scanner/getfunc
show options
```

We have checked what parameters our exploit needs to run. This exploit needs;

RHOST = remote host ip address UID = Unit id paramateres.

```
set RHOST <ip>
set UID <uid>
```

After giving the parameters which are wanted we are ready to run it.

```
exploit
```

Before we say exploit, we need to detect UID, and we will use the modbus/scanner/uid module in the smod tool.

```
use modbus/scanner/uid
show options
set RHOST <ip>
exploit
```

After the UID detection is performed, we can return to the other operation.

For function code detection, we start the process again with the exploit command of the modbus/scanner/getfunc module and we have determined which record is running in which function code.

Packet Analysing

We perform packet analysis from our attacker machine. Before running your module on our attacker machine, we run our wireshark tool and then run our module. İMAGE-1

We can see how the function code 1 is detected by saying "Modbus.func_code == 1". It is possible to see this number by increasing it in other function codes. İMAGE-2

Package includes "mbtcp.trans_id, mbtcp.prot_id, mbtcp.len, mbtcp.unit_id" values İMAGE-3

MBAP (Modbus Application Protocol Header) Header Section

In Modbus TCP/IP framing, the MBAP header consists of 4 parts. The length of the MBAP header is 7 bytes. The MBAP section is for the communication function. It carries some information so that the master and slave units can communicate with each other.

Transaction Identifier

It is the section that allows two devices to communicate with each other by associating the master and slave points with each other. The process descriptor part

is 2 bytes long.

Protocol Identifier

Designed for multiple systems. It takes the value 0 for Modbus. This area has been reserved for future use. The protocol descriptor part is 2 Bytes long.

Length

The length section specifies the data length in bytes, which includes the unit descriptor section and the Modbus TCP/IP PDU sections. The length part is 2 Bytes long.

Unit Identifier

This field is used for routing for the system. It is used to identify the remote unit that is not on the network. The volume descriptor part is 1 Byte long.

Modbus TCP/IP PDU Section

Modbus TCP/IP PDU section consists of 2 sections, function code and data. In Modbus TCP/IP framing, the Modbus TCP/IP PDU section basically contains the codes for the function.

Function Code

The function code section consists of the code for the requested function when sending a message from the master unit to the slave unit. In response to the master unit from the slave unit, it consists of the function code corresponding to the request. The length of the function code is 1 byte.

Data Section

The data section contains special data for the requested function when sending a message from the master unit to the slave unit. In response to the master unit from the slave unit, it consists of the response data or error codes to be sent against the request. The data section is of variable length.