Roozah Khan

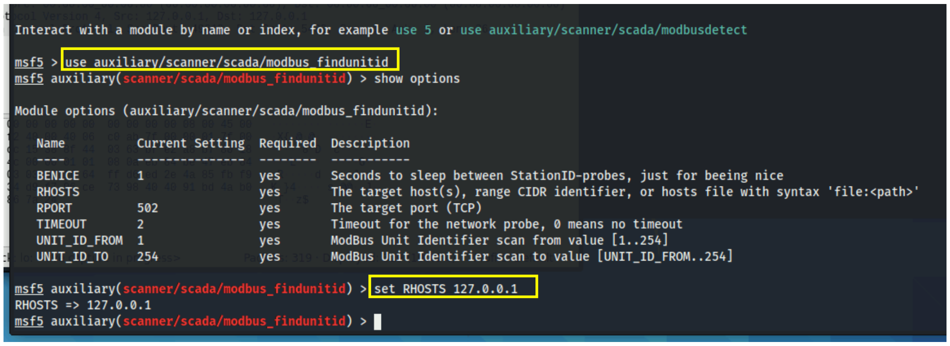
Using Metasploit to exploit Modbus terminals.

1. Watch this video to learn basics of Metasploit: https://www.youtube.com/watch?v=K7y_-JtpZ7I&t=502s
2. Enter the Kali Linux 2020.3 environment
3. Download ModbusPal within Kali. After download, run ModbusPal with `sudo java -jar ModbusPal.jar`.
4. Follow the Help doc of ModbusPal, create one slave. Within the slave, add two holding registers, and two coils. Run ModbusPal after the configuration.
5. Bring up Metasploit, search the keyword "modbus" and find out what scripts are available for modbus. The RHOST should be set as 127.0.0.1, i.e., the same Kali.
6. Bring up Wireshark and monitor on the interface `loopback`.
7. Answer the questions in the following.

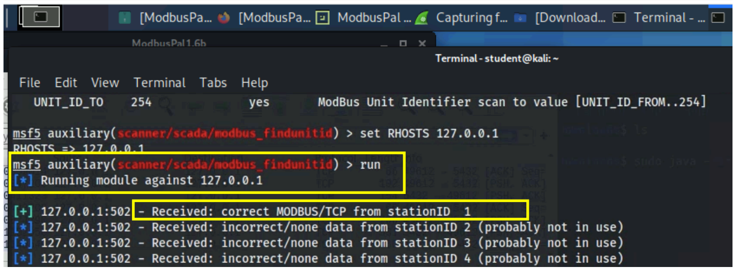
All the following questions need to be answered based on the output from Metasploit, and the trace you have captured in Wireshark. Both verbal explanations and screenshots are needed. 10 points for each sub question.

1. Use the module `modbus_findunitid`, and answer the following questions.
   a. Briefly explain what does this module do, what is the result running this module, and how would you interpret the result.

The yellow squared box in the screenshot are the steps I used to use the modbus_findunitid module and set the RHOST target ID.



```
Interact with a module by name or index, for example use 5 or use auxiliary/scanner/scada/modbusdetect

msf5 > use auxiliary/scanner/scada/modbus_findunitid
msf5 auxiliary(scanner/scada/modbus_findunitid) > show options

Module options (auxiliary/scanner/scada/modbus_findunitid):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   BENICE        1                yes       Seconds to sleep between StationID-probes, just for beeing nice
   RHOSTS                         yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT         502              yes       The target port (TCP)
   TIMEOUT       2                yes       Timeout for the network probe, 0 means no timeout
   UNIT_ID_FROM  1                yes       ModBus Unit Identifier scan from value [1..254]
   UNIT_ID_TO    254              yes       ModBus Unit Identifier scan to value [UNIT_ID_FROM..254]

msf5 auxiliary(scanner/scada/modbus_findunitid) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf5 auxiliary(scanner/scada/modbus_findunitid) > 
```



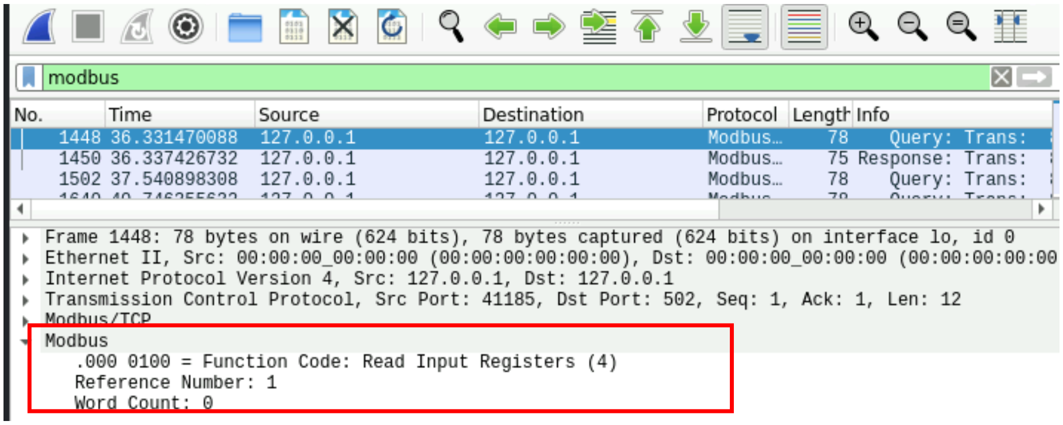
[ModbusPa... [ModbusPa... ModbusPal ... Capturing f... [Download... Terminal - ...
ModbusPal 1.6b
Terminal - student@kali: ~
File Edit View Terminal Tabs Help

```
  UNIT_ID_TO    254              yes       ModBus Unit Identifier scan to value [UNIT_ID_FROM..254]

msf5 auxiliary(scanner/scada/modbus_findunitid) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf5 auxiliary(scanner/scada/modbus_findunitid) > run
[*] Running module against 127.0.0.1

[+] 127.0.0.1:502 - Received: correct MODBUS/TCP from stationID  1
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 2 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 3 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 4 (probably not in use)
```

This modbus module is a modbus unit ID and station ID enumerator. It sends a command to the modbus endpoint and if it is sent to the correct ID it will, it will return with the correct ID. It is used to attack by scanning any modbus slaves.

By running this module, it scans and finds one of the Station ID successfully.

b. What Modbus message/messages are send by this module to ModbusPal? What is the function code of those messages, and what is the intention of this function code?

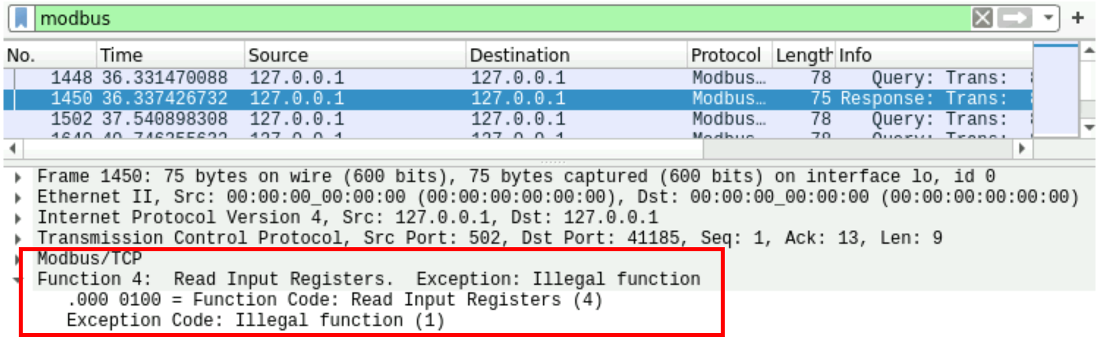


```
No.    Time          Source      Destination  Protocol  Length Info
  1448 36.331470088  127.0.0.1   127.0.0.1    Modbus…       78     Query: Trans:
  1450 36.337426732  127.0.0.1   127.0.0.1    Modbus…       75 Response: Trans:
  1502 37.540898308  127.0.0.1   127.0.0.1    Modbus…       78     Query: Trans:

Frame 1448: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 41185, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
Modbus/TCP
Modbus
    .000 0100 = Function Code: Read Input Registers (4)
    Reference Number: 1
    Word Count: 0
```

The messages that are sent by the module to modbuspal are in the red box in the screenshot above. The function code of the message is “Read Input Registers (4)” which means that it reads the value given to the registers in the slave.

c. What are the responses from ModbusPal, and what those responses indicate?

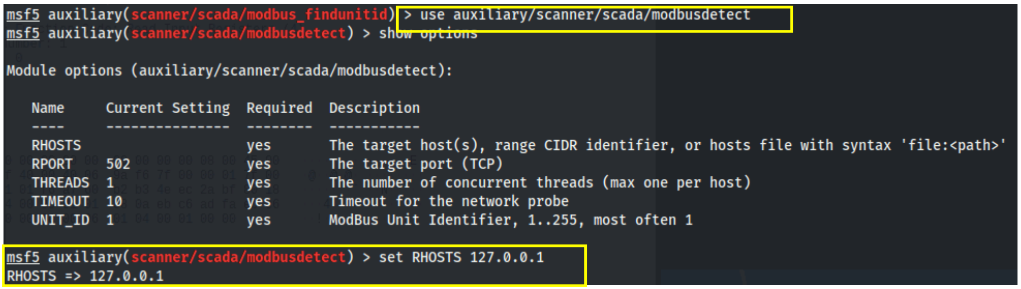
The responses are shown in the red box in the screenshot below. The responses indicate that there was an “illegal function (1)” which probably means that there was an error finding the function or it was not supported.



```
modbus

No.   Time          Source      Destination  Protocol  Length Info
 1448 36.331470088  127.0.0.1   127.0.0.1    Modbus…     78    Query: Trans:
 1450 36.337426732  127.0.0.1   127.0.0.1    Modbus…     75 Response: Trans:
 1502 37.540898308  127.0.0.1   127.0.0.1    Modbus…     78    Query: Trans:
 1640 40.746255623  127.0.0.1   127.0.0.1    Modbus…     78    Query: Trans:

Frame 1450: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 502, Dst Port: 41185, Seq: 1, Ack: 13, Len: 9
Modbus/TCP
Function 4:  Read Input Registers.  Exception: Illegal function
   .000 0100 = Function Code: Read Input Registers (4)
   Exception Code: Illegal function (1)
```

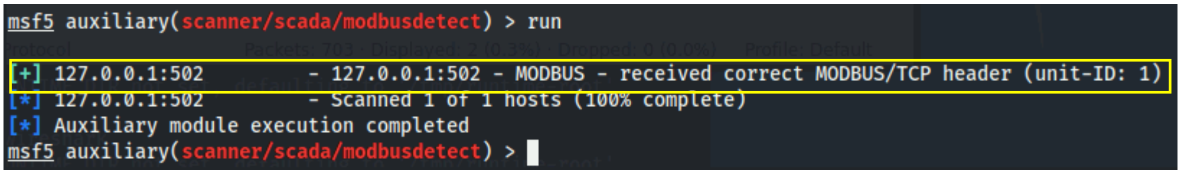
2. Use the module `modbusdetect`, and answer the following questions.
   a. Briefly explain what does this module do, what is the result running this module, and how would you interpret the result.

The yellow squared box in the screenshot are the steps I used to use the modbusdetect module and set the RHOST to the target ID or IP address of the Target host.



```
msf5 auxiliary(scanner/scada/modbus_findunitid) > use auxiliary/scanner/scada/modbusdetect
msf5 auxiliary(scanner/scada/modbusdetect) > show options

Module options (auxiliary/scanner/scada/modbusdetect):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT    502              yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)
   TIMEOUT  10               yes       Timeout for the network probe
   UNIT_ID  1                yes       ModBus Unit Identifier, 1..255, most often 1

msf5 auxiliary(scanner/scada/modbusdetect) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
```

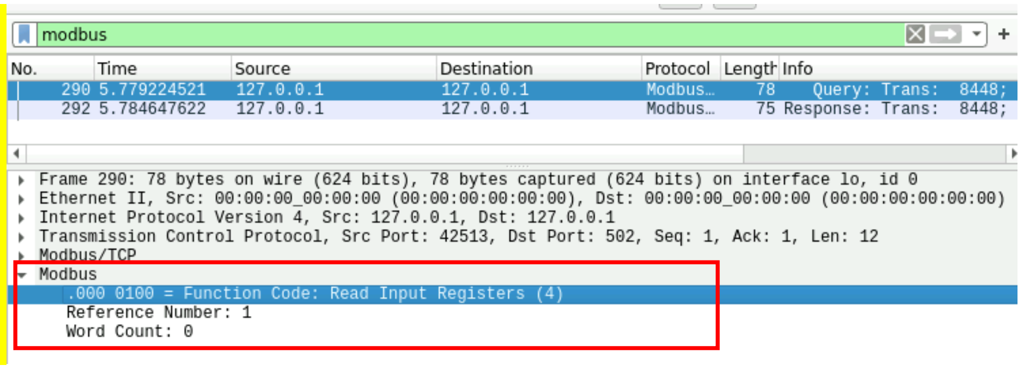
This module is the modbus version scanner where it scans the target ID to see if it is running on Modbus and the result shows that the target host is running on Modbus in the screenshot below.



```
msf5 auxiliary(scanner/scada/modbusdetect) > run

[+] 127.0.0.1:502         - 127.0.0.1:502 - MODBUS - received correct MODBUS/TCP header (unit-ID: 1)
[*] 127.0.0.1:502         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/scada/modbusdetect) >
```

   b. What Modbus message/messages are send by this module to ModbusPal? What is the function code of those messages, and what is the intention of this function code?

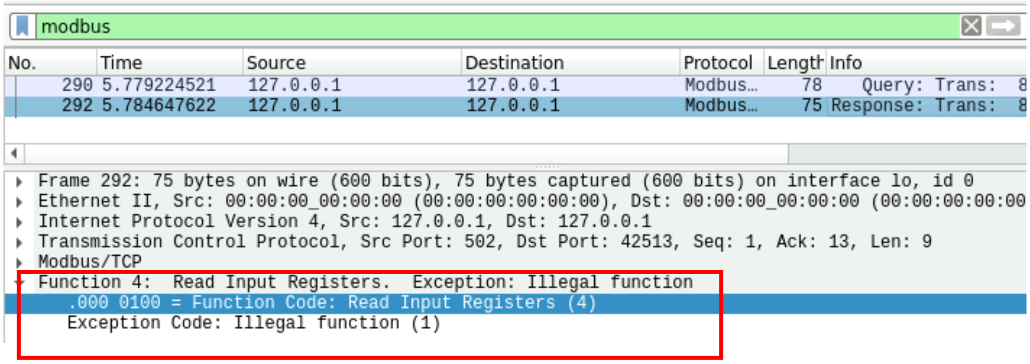
The messages that are sent by the module to modbuspal are in the red box in the screenshot below. The function code of the message is “Read Input Registers (4)” which means that it reads the value given to the registers in the slave.



```
modbus
No.   Time          Source      Destination  Protocol  Length Info
  290 5.779224521   127.0.0.1   127.0.0.1    Modbus…     78    Query: Trans:  8448;
  292 5.784647622   127.0.0.1   127.0.0.1    Modbus…     75 Response: Trans:  8448;

▸ Frame 290: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface lo, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 42513, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
▸ Modbus/TCP
▾ Modbus
    .000 0100 = Function Code: Read Input Registers (4)
    Reference Number: 1
    Word Count: 0
```

c. What are the responses from ModbusPal, and what those responses indicate?

The responses are shown in the red box in the screenshot below. The responses indicate that there was an “illegal function (1)” which probably means that there was an error finding the function or it was not supported.

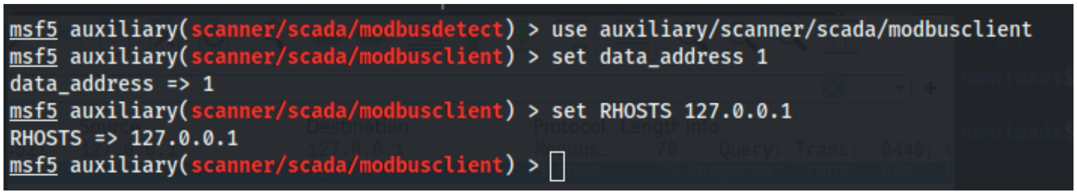


```
modbus
No.   Time          Source      Destination  Protocol  Length Info
  290 5.779224521   127.0.0.1   127.0.0.1    Modbus…     78    Query: Trans:  8
  292 5.784647622   127.0.0.1   127.0.0.1    Modbus…     75 Response: Trans:  8

▸ Frame 292: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface lo, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 502, Dst Port: 42513, Seq: 1, Ack: 13, Len: 9
▸ Modbus/TCP
▾ Function 4:  Read Input Registers.  Exception: Illegal function
    .000 0100 = Function Code: Read Input Registers (4)
    Exception Code: Illegal function (1)
```

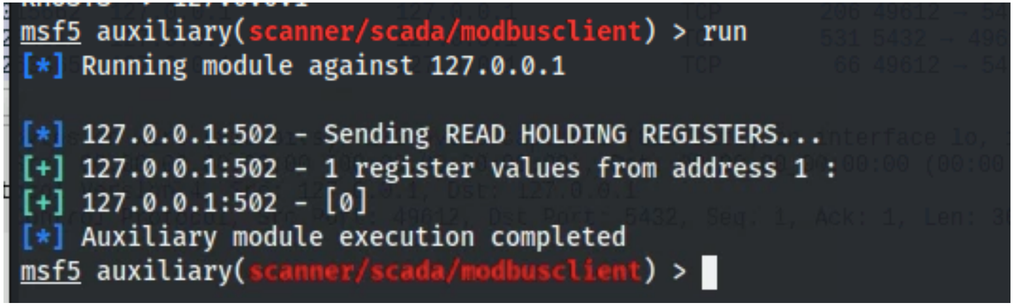
3. Use the module `modbusclient`, and answer the following questions.

   a. Set the parameter `data_address` to 1, and run the module.

   In the screenshot below are the steps I used for modbusclient. I set the data address to 1 and I set the target host IP address as well using RHOSTS. This module is used to attack the Target host to read and write the register or coils on a slave.



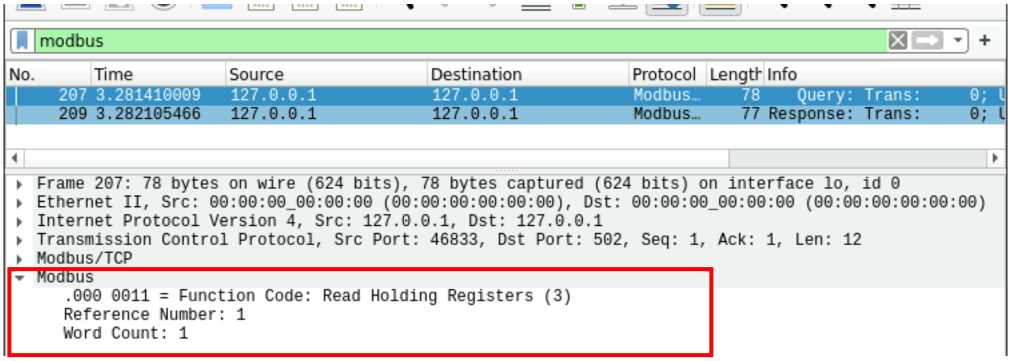
```
msf5 auxiliary(scanner/scada/modbusdetect) > use auxiliary/scanner/scada/modbusclient
msf5 auxiliary(scanner/scada/modbusclient) > set data_address 1
data_address => 1
msf5 auxiliary(scanner/scada/modbusclient) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf5 auxiliary(scanner/scada/modbusclient) >
```



```
msf5 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 127.0.0.1

[*] 127.0.0.1:502 - Sending READ HOLDING REGISTERS...
[+] 127.0.0.1:502 - 1 register values from address 1 :
[+] 127.0.0.1:502 - [0]
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/scada/modbusclient) >
```

i. What Modbus message/messages are send by this module to ModbusPal? What is the function code of those messages, and what is the intention of this function code?

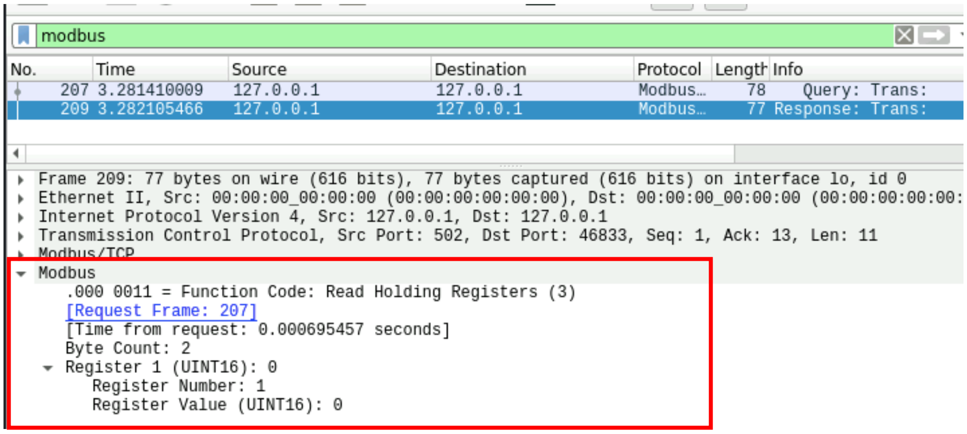
The messages are in the red box in the screenshot below. The function code is “Read Holding Registers (3)” which means that reads the content of the registers.



```
modbus

No.   Time         Source      Destination  Protocol  Length Info
  207 3.281410009  127.0.0.1   127.0.0.1    Modbus…     78     Query: Trans:     0;
  209 3.282105466  127.0.0.1   127.0.0.1    Modbus…     77 Response: Trans:     0;

▸ Frame 207: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface lo, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 46833, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
▸ Modbus/TCP
▾ Modbus
    .000 0011 = Function Code: Read Holding Registers (3)
    Reference Number: 1
    Word Count: 1
```

ii. What are the responses from ModbusPal, and what those responses indicate?

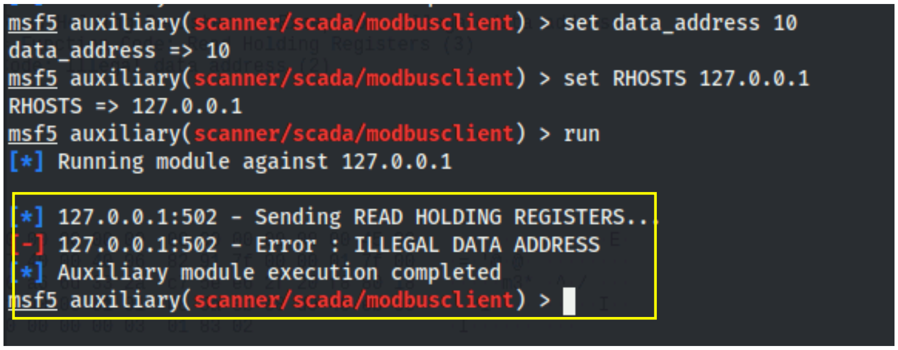
The response is in the red box in the screenshot below. The responses indicate that the registers were read and the byte count is 2 along with the register number which is 1 and the value which is 0.



```
modbus

No.   Time         Source      Destination  Protocol  Length Info
  207 3.281410009  127.0.0.1   127.0.0.1    Modbus…     78     Query: Trans:
  209 3.282105466  127.0.0.1   127.0.0.1    Modbus…     77 Response: Trans:

▸ Frame 209: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface lo, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 502, Dst Port: 46833, Seq: 1, Ack: 13, Len: 11
▸ Modbus/TCP
▾ Modbus
    .000 0011 = Function Code: Read Holding Registers (3)
    [Request Frame: 207]
    [Time from request: 0.000695457 seconds]
    Byte Count: 2
  ▾ Register 1 (UINT16): 0
      Register Number: 1
      Register Value (UINT16): 0
```

b. Set the parameter data_address to 10, and run the module again.

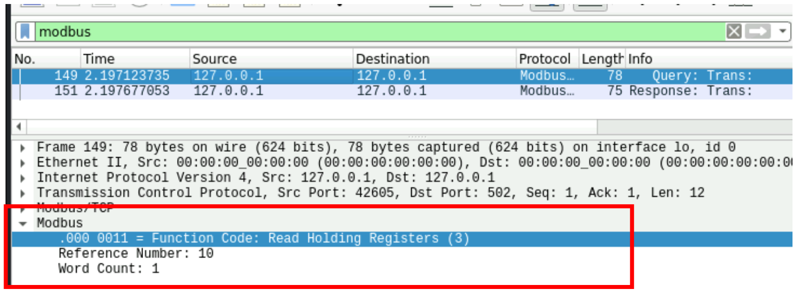
I used the steps in the screenshot below to set the data address to 10 and set the RHOSTS to the target IP address. The result gave me an error because there is no 10 data address.



```
msf5 auxiliary(scanner/scada/modbusclient) > set data_address 10
data_address => 10
msf5 auxiliary(scanner/scada/modbusclient) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf5 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 127.0.0.1

[*] 127.0.0.1:502 - Sending READ HOLDING REGISTERS...
[-] 127.0.0.1:502 - Error : ILLEGAL DATA ADDRESS
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/scada/modbusclient) > 
```

i. What Modbus message/messages are send by this module to ModbusPal? What is the function code of those messages, and what is the intention of this function code?

The messages are in the red box in the screenshot below. The function code is "Read Holding Registers (3)" which means that reads the content of the registers.



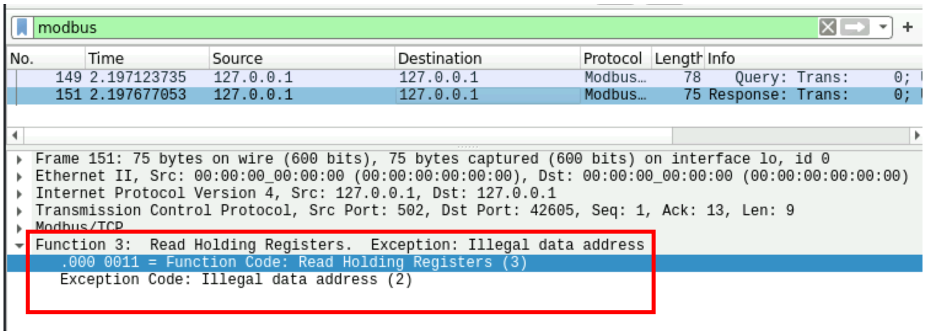
modbus

```
No.   Time         Source      Destination  Protocol  Length Info
  149 2.197123735  127.0.0.1   127.0.0.1    Modbus…     78     Query: Trans:
  151 2.197677053  127.0.0.1   127.0.0.1    Modbus…     75 Response: Trans:
```

```
▸ Frame 149: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface lo, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:0
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 42605, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
▸ Modbus/TCP
▾ Modbus
    .000 0011 = Function Code: Read Holding Registers (3)
    Reference Number: 10
    Word Count: 1
```

ii. What are the responses from ModbusPal, and what those responses indicate?

The responses are in the red box in the screenshot below. The responses indicate that the data address 10 cannot be read because it is not the correct values registered in the slave to be read.



modbus

```
No.   Time         Source      Destination  Protocol  Length Info
  149 2.197123735  127.0.0.1   127.0.0.1    Modbus…     78     Query: Trans:      0;
  151 2.197677053  127.0.0.1   127.0.0.1    Modbus…     75 Response: Trans:      0;
```

```
▸ Frame 151: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface lo, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 502, Dst Port: 42605, Seq: 1, Ack: 13, Len: 9
▸ Modbus/TCP
▾ Function 3:  Read Holding Registers.  Exception: Illegal data address
    .000 0011 = Function Code: Read Holding Registers (3)
    Exception Code: Illegal data address (2)
```