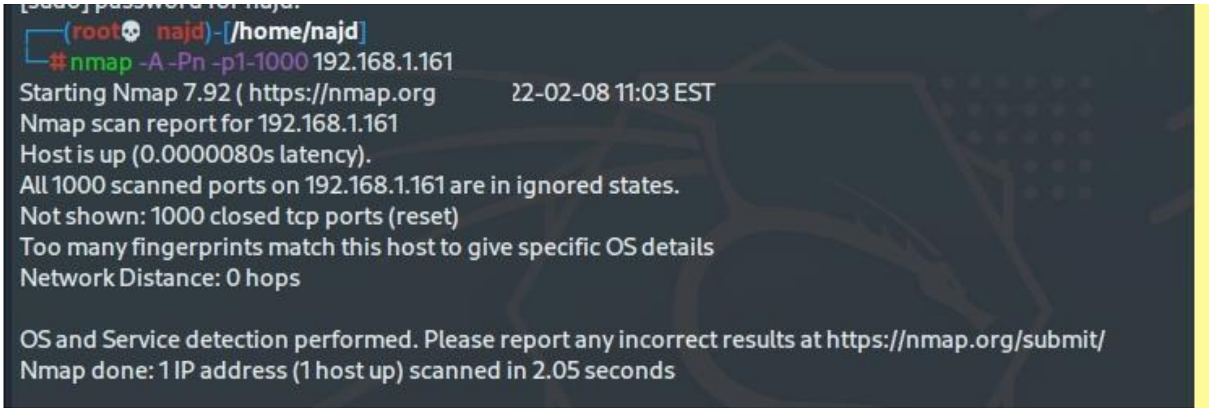
Lab 10-2-2: Testing and Monitoring our SCADA HoneyPot

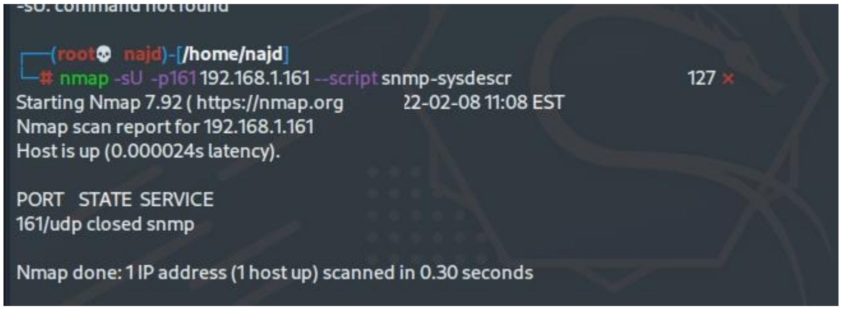
Lab Requirement

- Ubuntu Machine (Honeypot already setup)
- Kali Linux Machine

Step #1 Scan with nmap

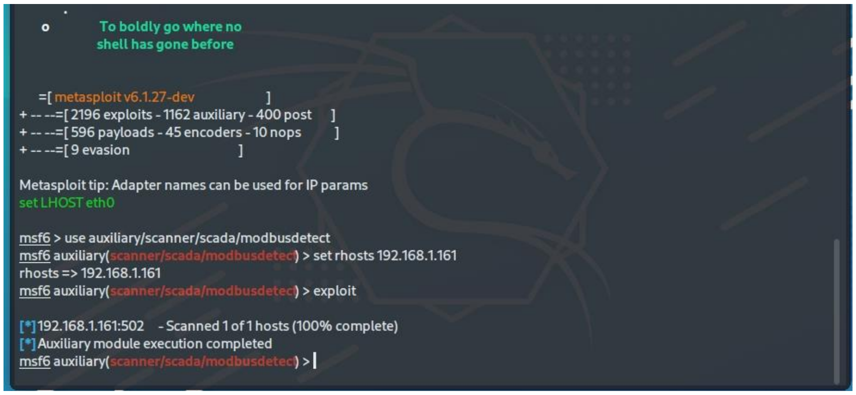


```
[sudo] password for najd:
┌──(root💀 najd)-[/home/najd]
└─# nmap -A -Pn -p1-1000 192.168.1.161
Starting Nmap 7.92 ( https://nmap.org          22-02-08 11:03 EST
Nmap scan report for 192.168.1.161
Host is up (0.0000080s latency).
All 1000 scanned ports on 192.168.1.161 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
Too many fingerprints match this host to give specific OS details
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 2.05 seconds
```

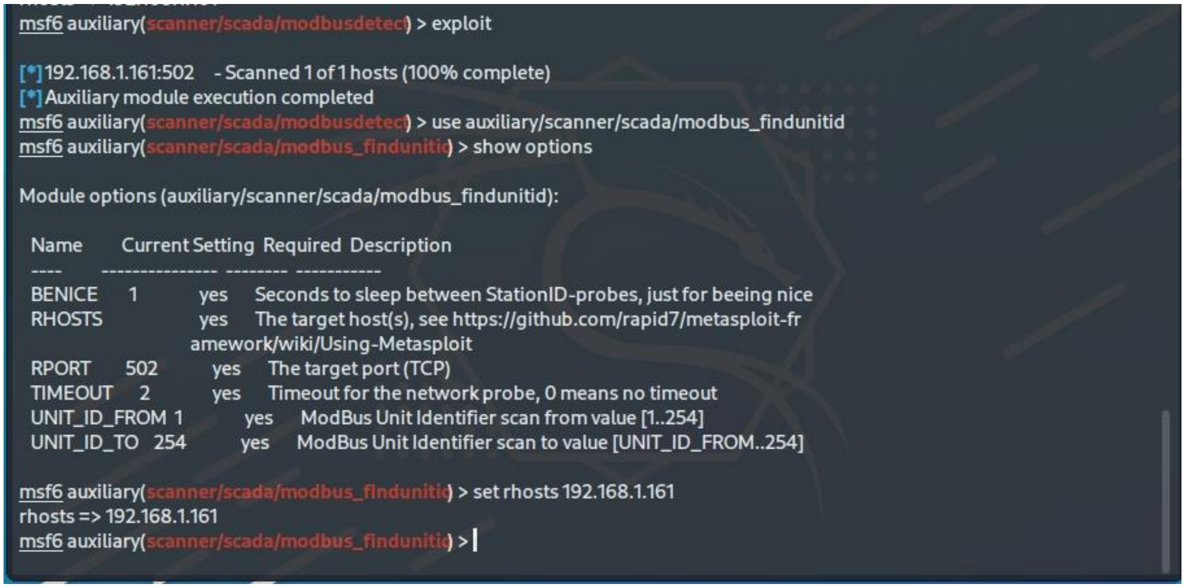


```
-sU: command not found

┌──(root💀 najd)-[/home/najd]
└─# nmap -sU -p161 192.168.1.161 --script snmp-sysdescr                 127 ✕
Starting Nmap 7.92 ( https://nmap.org          22-02-08 11:08 EST
Nmap scan report for 192.168.1.161
Host is up (0.000024s latency).

PORT   STATE  SERVICE
161/udp closed snmp

Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
```

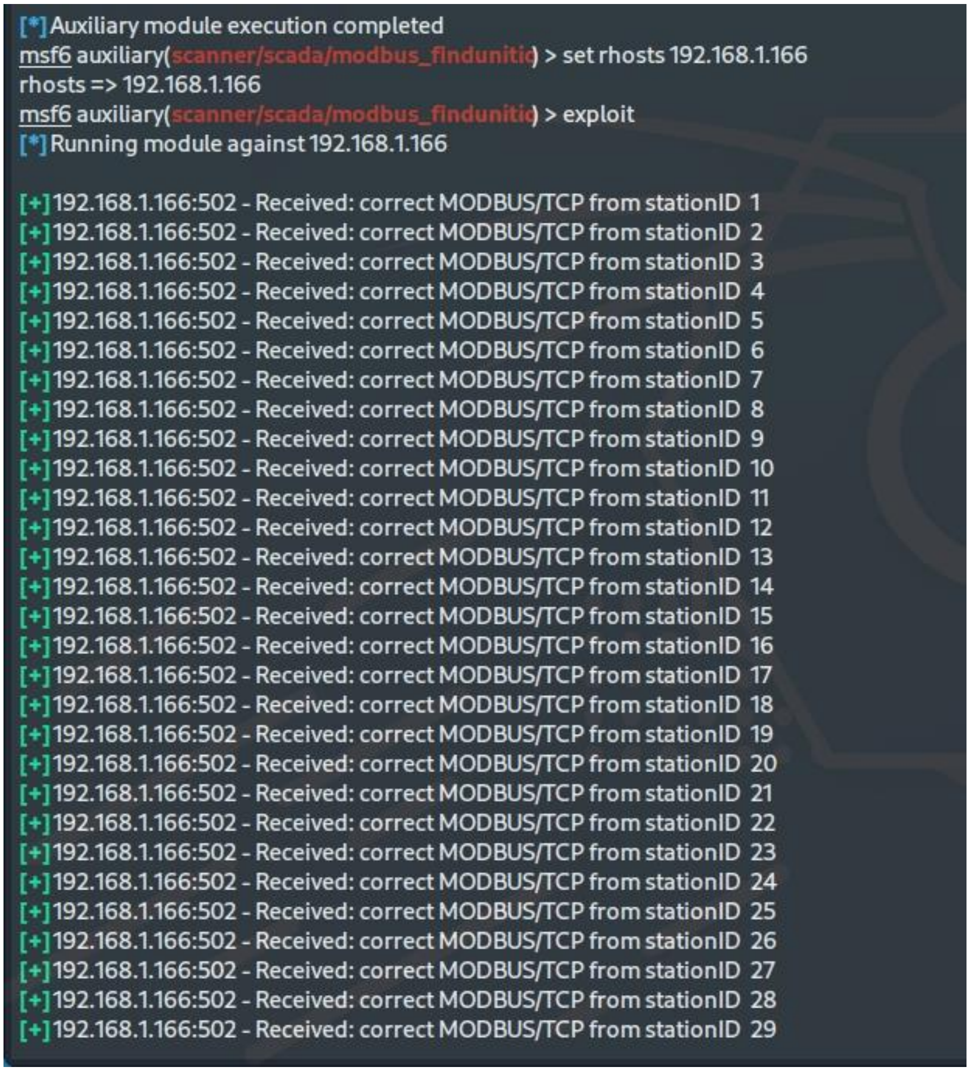
Step #2 Metasploit Scan on the Honeypot



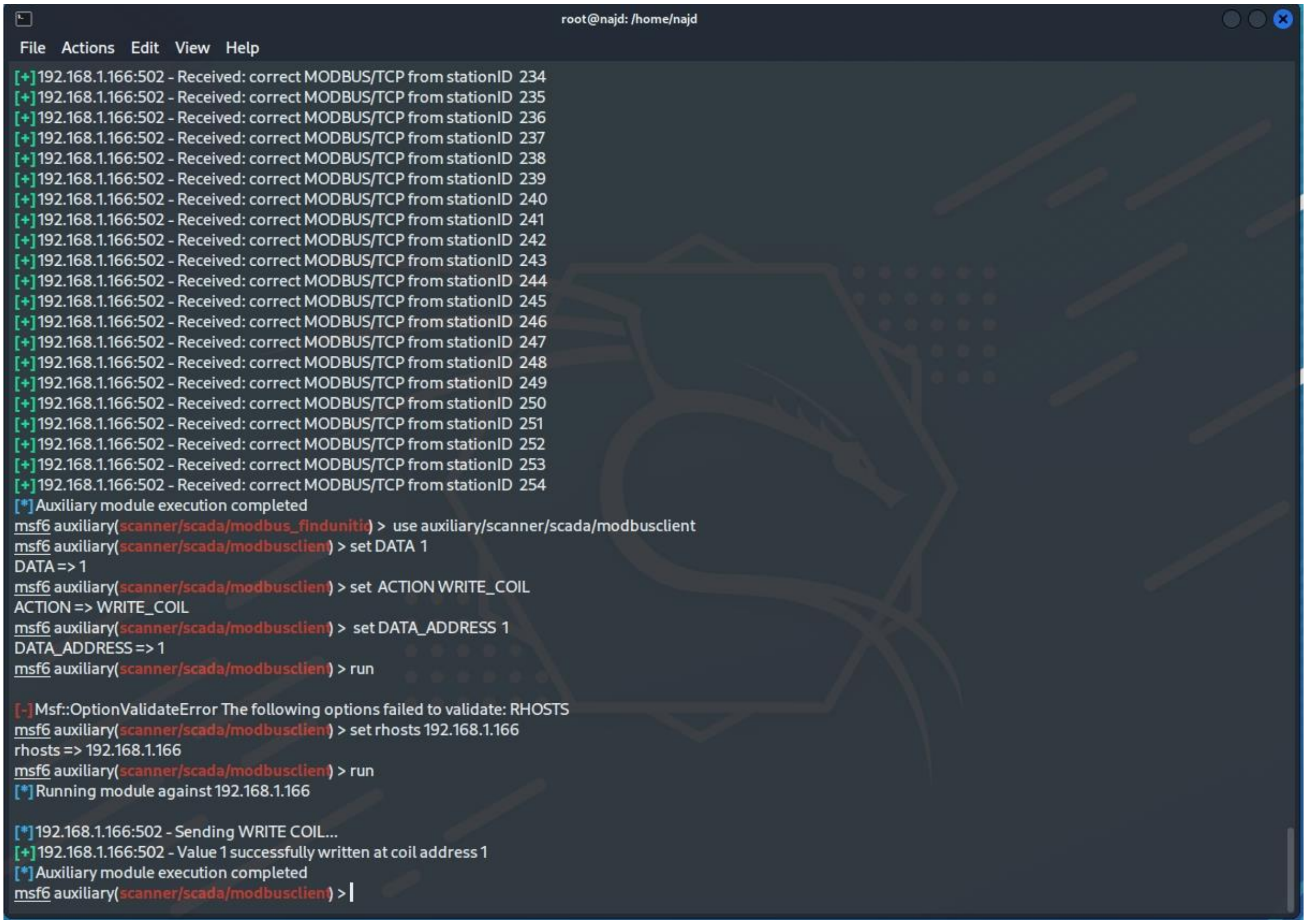
```
   o          To boldly go where no
             shell has gone before


       =[ metasploit v6.1.27-dev                 ]
+ -- --=[ 2196 exploits - 1162 auxiliary - 400 post     ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops         ]
+ -- --=[ 9 evasion                              ]

Metasploit tip: Adapter names can be used for IP params
set LHOST eth0

msf6 > use auxiliary/scanner/scada/modbusdetect
msf6 auxiliary(scanner/scada/modbusdetect) > set rhosts 192.168.1.161
rhosts => 192.168.1.161
msf6 auxiliary(scanner/scada/modbusdetect) > exploit

[*] 192.168.1.161:502    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusdetect) >
```



```
msf6 auxiliary(scanner/scada/modbusdetect) > exploit

[*] 192.168.1.161:502    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusdetect) > use auxiliary/scanner/scada/modbus_findunitid
msf6 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), see https://github.com/rapid7/metasploit-fr
                                            amework/wiki/Using-Metasploit
   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]

msf6 auxiliary(scanner/scada/modbus_findunitid) > set rhosts 192.168.1.161
rhosts => 192.168.1.161
msf6 auxiliary(scanner/scada/modbus_findunitid) >
```



```
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbus_findunitid) > set rhosts 192.168.1.166
rhosts => 192.168.1.166
msf6 auxiliary(scanner/scada/modbus_findunitid) > exploit
[*] Running module against 192.168.1.166

[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  1
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  2
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  3
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  4
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  5
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  6
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  7
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  8
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  9
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  10
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  11
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  12
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  13
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  14
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  15
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  16
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  17
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  18
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  19
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  20
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  21
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  22
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  23
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  24
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  25
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  26
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  27
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  28
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID  29
```



```
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 234
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 235
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 236
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 237
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 238
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 239
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 240
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 241
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 242
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 243
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 244
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 245
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 246
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 247
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 248
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 249
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 250
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 251
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 252
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 253
[+] 192.168.1.166:502 - Received: correct MODBUS/TCP from stationID 254
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbus_findunitid) > use auxiliary/scanner/scada/modbusclient
msf6 auxiliary(scanner/scada/modbusclient) > set DATA 1
DATA => 1
msf6 auxiliary(scanner/scada/modbusclient) > set ACTION WRITE_COIL
ACTION => WRITE_COIL
msf6 auxiliary(scanner/scada/modbusclient) > set DATA_ADDRESS 1
DATA_ADDRESS => 1
msf6 auxiliary(scanner/scada/modbusclient) > run

[-] Msf::OptionValidateError The following options failed to validate: RHOSTS
msf6 auxiliary(scanner/scada/modbusclient) > set rhosts 192.168.1.166
rhosts => 192.168.1.166
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.166

[*] 192.168.1.166:502 - Sending WRITE COIL...
[+] 192.168.1.166:502 - Value 1 successfully written at coil address 1
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) >
```