

# ModScan

A SCADA MODBUS Network Scanner

Mark Bristow

[mark.bristow@gmail.com](mailto:mark.bristow@gmail.com)

# Agenda

- Brief introduction to SCADA Systems
- The MODBUS Protocol
- MODBUS TCP
- ModScan Demonstration
- ModScan Project Information
- Q&A

# Disclaimer

- The material in this presentation is to be used for authorized security scanning/auditing
- If you do something stupid with the information I present here, don't blame me

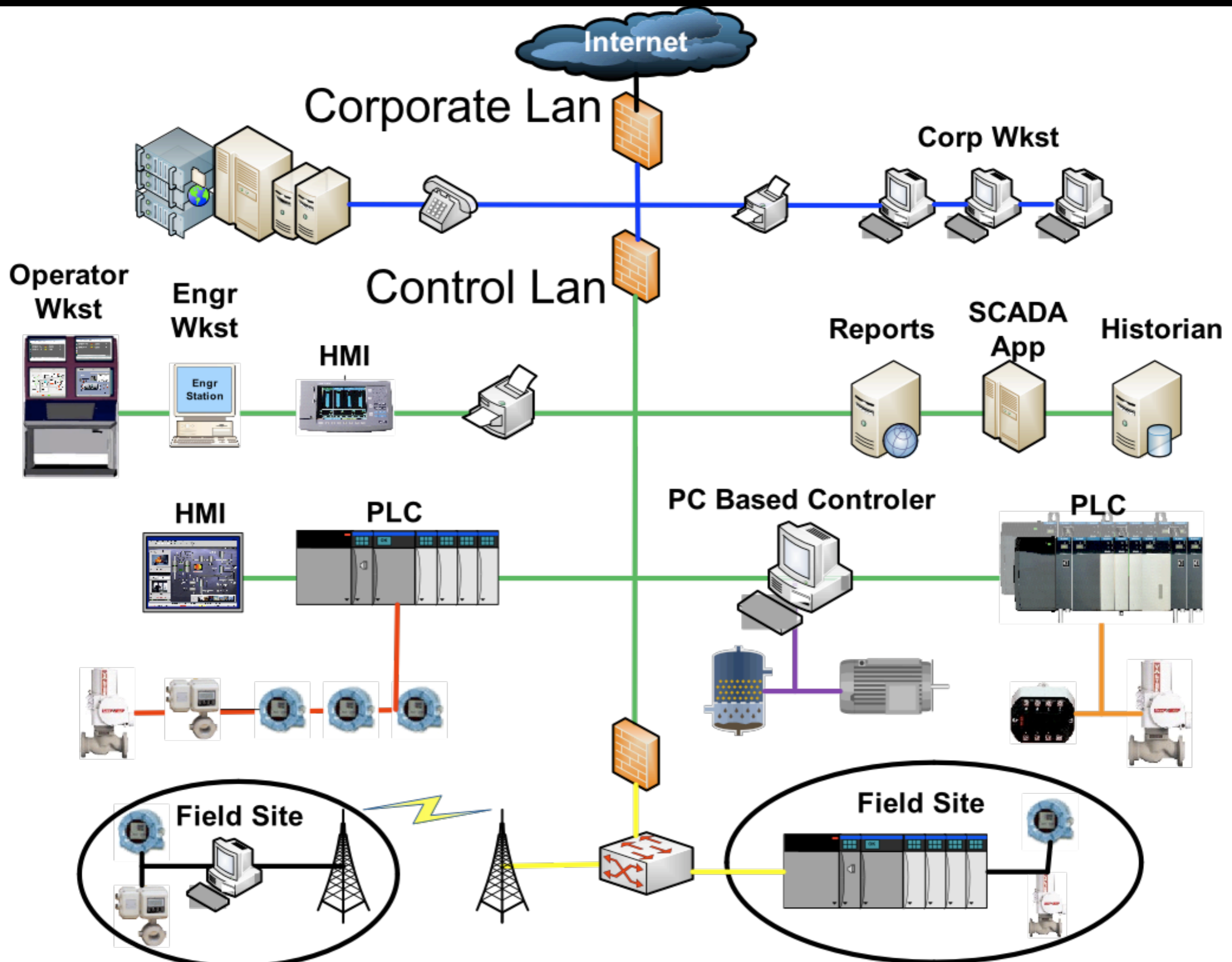
# What is SCADA?

- Supervisory Control And Data Acquisition is a system that centrally gathers data in real time from local and remote locations in order to control equipment and conditions.
- Commonly also referred to as Industrial Control Systems (ICS), which is not accurate but close

# Where is SCADA?

- Power Generation/Transmission
- Water Treatment/Distribution
- Pipelines
- Traffic Control Systems
- Manufacturing Facilities
- National Infrastructure
- Communications

# SCADA Architecture



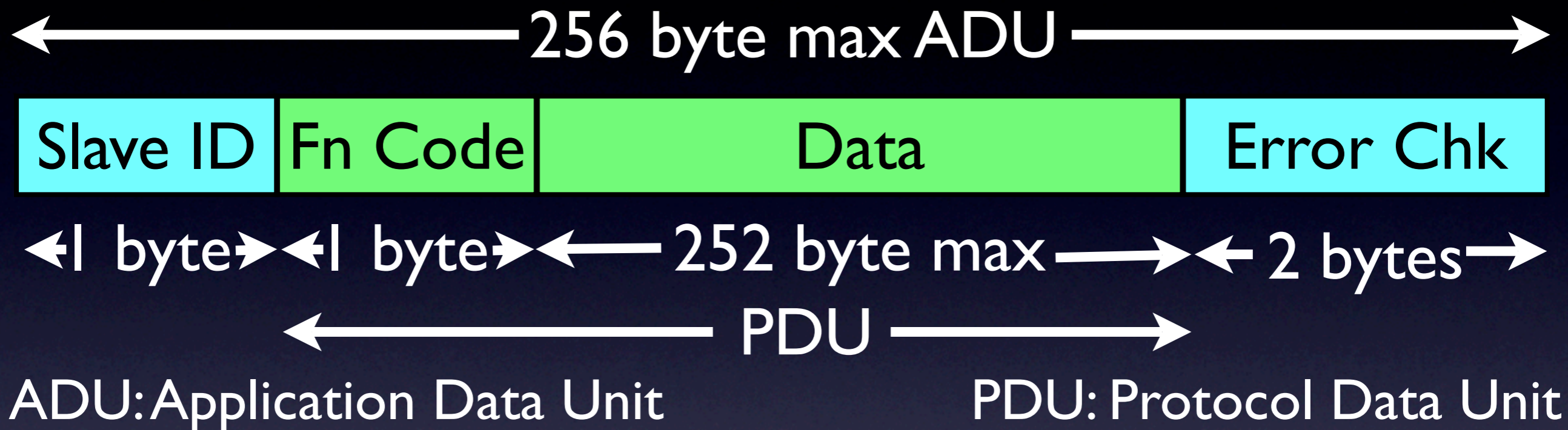
# What is ModScan?

- ModScan is a tool to detect open MODBUS/TCP ports and identify device Slave IDs associated with IP addresses
- ModScan is designed for an administrator or security auditor to be able to accurately reconnoiter a MODBUS/TCP network

# The MODBUS Protocol

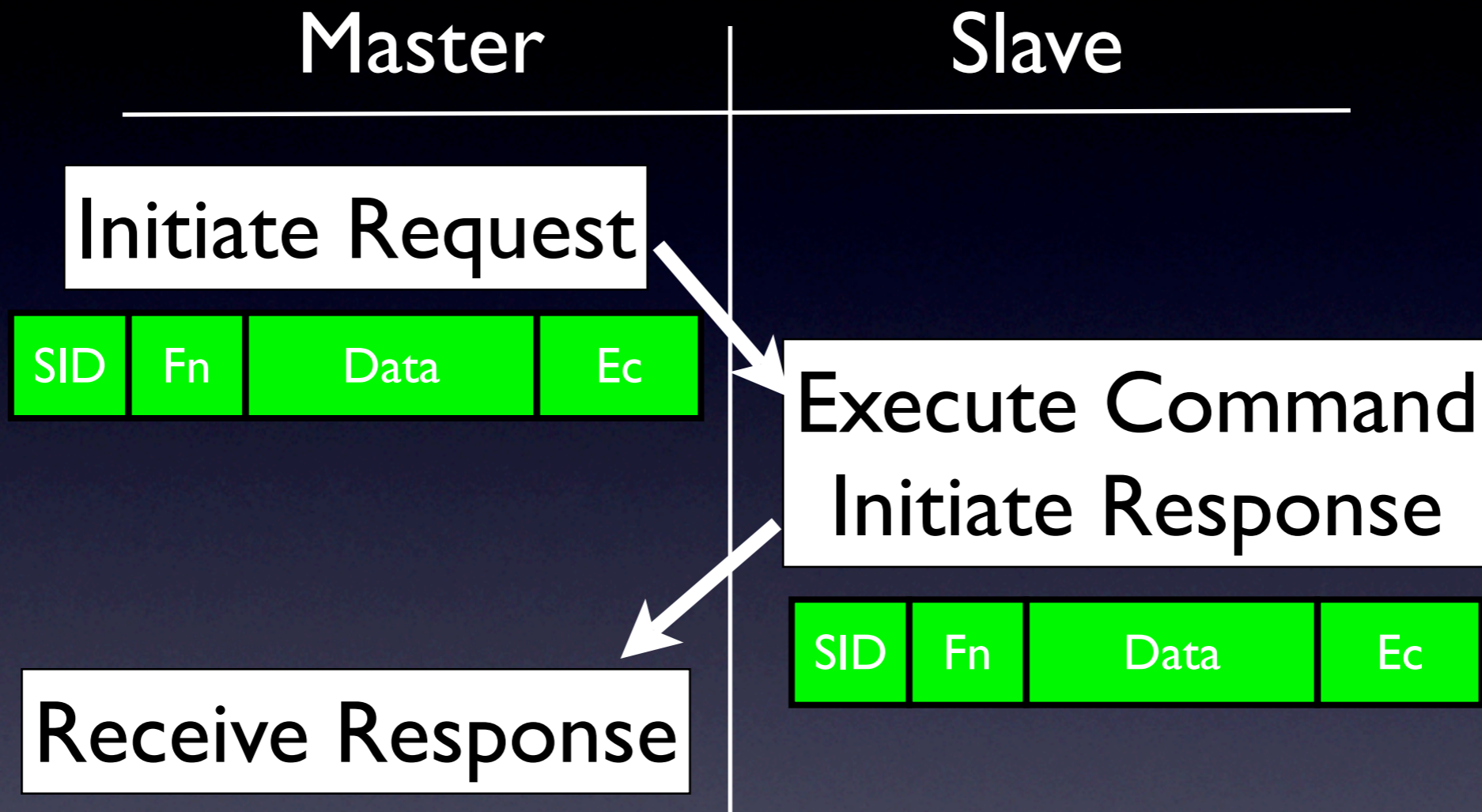
- About the Protocol
  - Developed in 1979 by Modicon
  - Free and Open Source
  - The most common protocol found in SCADA and ICS networks
  - Default port 503
- Flavors
  - Modbus RTU - Compact Binary
  - Modbus ASCII - Human readable

# MODBUS Packet Construction



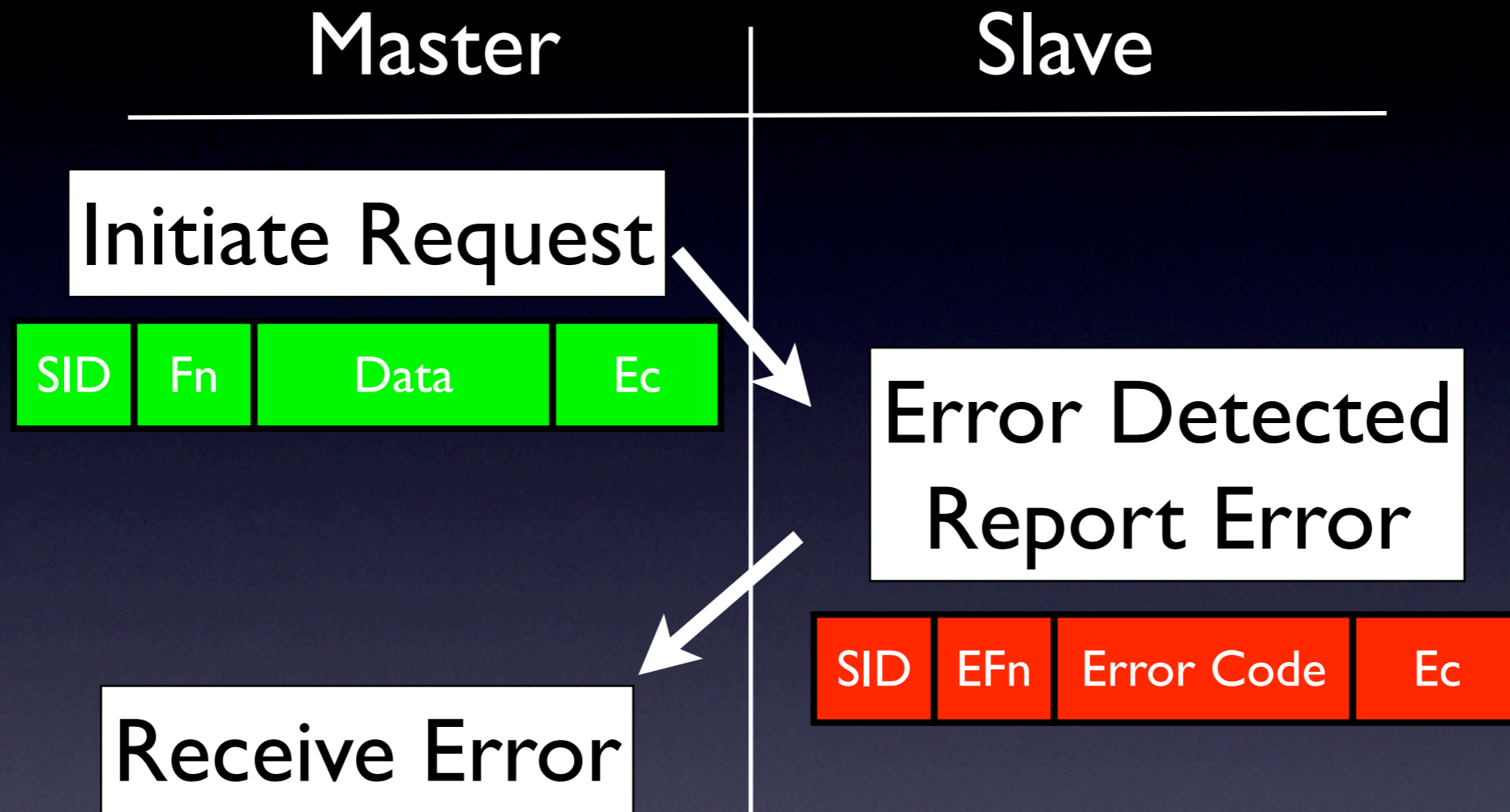
- Valid Function codes are 1-127
- 256 byte maximum packet size
- Big-Endian encoding
- Error Check is CRC/LRC

# Typical Communication



- Modbus is a Master/Slave Serial Protocol
- Only Masters can initiate conversation

# Error Communication



- Error Function =  $0x80 + \text{Function Code}$
- Error Codes defined in specification

# Function Codes

Function Code	Description
01	Read Coils
02	Read Discretes
03	Read Holding Registers
04	Read Input Registers
05	Write Coil
06	Write Register
07	Read Exception Status
08	Diagnostics
0B	Get Comm Event Counter
0C	Get Comm Event Log
0F	Write Multiple Coils
10	Write Multiple Registers
11	Report Slave ID
14	Read File Record
15	Write File Record
16	Mask Write Register
17	Read/Write Multiple Registers
18	Read FIFO Que

# Diagnostic Codes

Function Code	Description
00	Return Query Data
01	Restart Communication
02	Return Diagnostic Register
03	Change ASCII Input Delimiter
04	Force Listen Only Mode
05-09	Reserved
0A	Clear Counters and Diagnostic Reg.
0B	Return Bus Message Count
0C	Return Bus Communication Error Count
0D	Return Bus Exception Error Count
0E	Return Slave Message Count
0F	Return Slave No Response Count
10	Return Slave NAK Count
11	Return Slave Busy Count
12	Return Bus Character Overrun Count
13	Reserved
14	Clear Overrun Counter and Flag
16+	Reserved

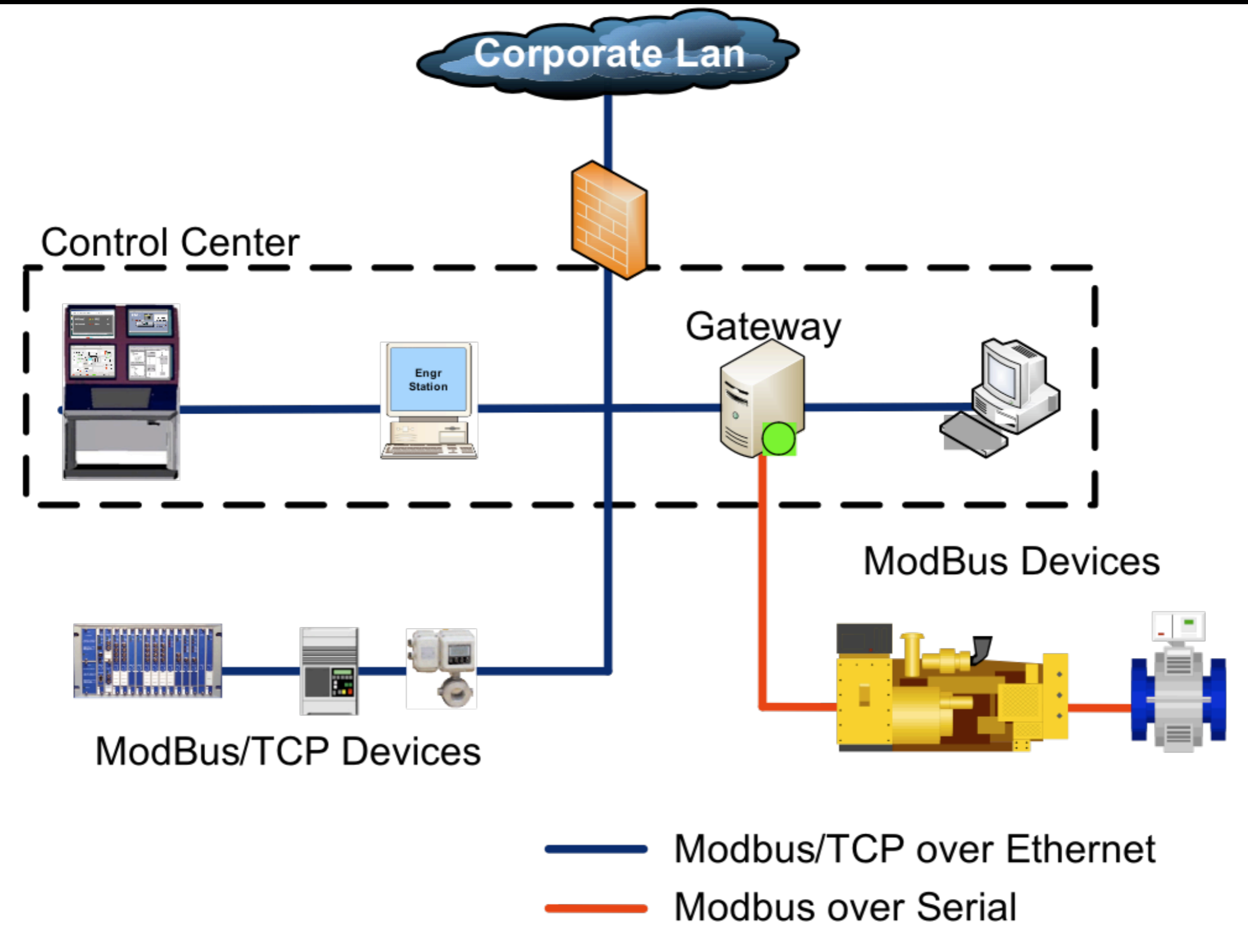
# Modbus Notes

- Addressing
  - Valid Slave IDs 1-247
  - Slave ID must be unique per bus
  - Masters do not have to have an address
  - Slaves will error when improperly addressed
- Communication
  - One request on the line at a time
  - Masters must wait for responses

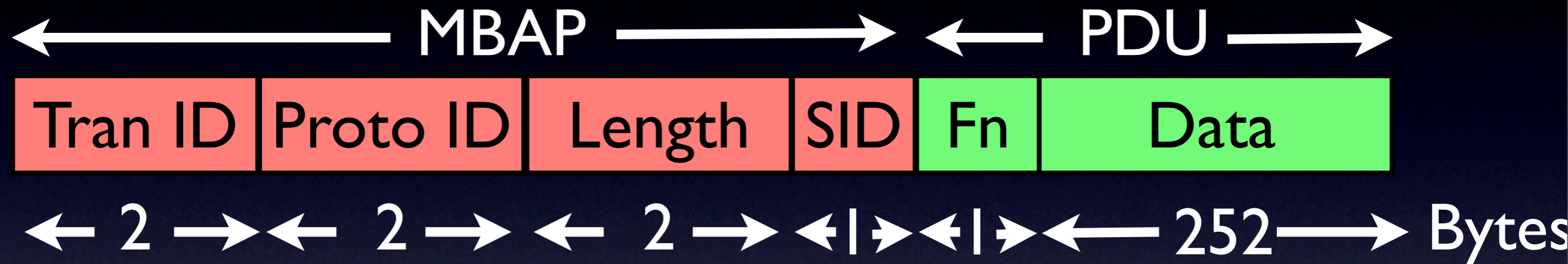
# ModBus/TCP

- ModBus protocol wrapped in TCP  
Goodness
- Checksum dropped
- Introduces Gateway device to ModBus
- Port 502 is reserved for Modbus/TCP
- No additional inherent security measures

# ModBus/TCP Architecture



# ModBus/TCP Packet



- MBAP: MODBUS Application Protocol Header
- PDU remains the same from the MODBUS spec
- Protocol ID is always 0x0000
- Big-Endian encoding

# Example Request

0000	02	00	00	00	45	00	00	40	79	51	40	00	40	06	00	00	....E..@ yQ@. @. . .
0010	7f	00	00	01	7f	00	00	01	ce	6f	01	f6	05	c4	86	3b	..... .0..... ;
0020	45	10	cf	83	80	18	ff	ff	fe	34	00	00	01	01	08	0a	E..... .4.....
0030	11	5b	88	4f	11	5b	87	d1	00	00	00	00	06	01	08		. [.0. [.. ..... ...S
0040	00	00	05	39													

Diag Code    Data                    Tran ID    Proto ID    Len    SID    FN

- Request sent by Master
- Request is to Slave 01 Fn 8 Diagnostics
- Diagnostic code 00 for "Return Query Data"

# Example Response

0000	02	00	00	00	45	00	00	40	3d	c5	40	00	40	06	00	00	....E..@ =.@. @...
0010	7f	00	00	01	7f	00	00	01	01	f6	c0	04	30	ce	2e	c4	..... 0...
0020	41	8b	4e	ef	80	18	ff	ff	fe	34	00	00	01	01	08	0a	A.N..... .4.....
0030	21	b9	fe	12	21	b9	fe	12	00	00	00	00	00	06	01	08	!...!... .....
0040	00	00	05	39													...9

Diag Code    Data                      Tran ID    Proto ID    Len    SID    FN

- Response sent by slave
- Request is to Slave 01 Fn 8 Diagnostics
- Diagnostic code 00 for "Return Query Data"
- Identical to Request

# Error Request

```
0000  02 00 00 00 45 00 00 3e 14 ec 40 00 40 06 00 00  ....E..> ..@.@...
0010  7f 00 00 01 7f 00 00 01 c0 08 01 f6 68 4e c3 f8  ....hN..
0020  5c f0 89 dc 80 18 ff ff fe 32 00 00 01 01 08 0a  \......2.....
0030  21 ba 60 c4 21 ba 60 c4 00 00 00 00 00 02 01 08  !.!.!.!.....
0040  00 ff
```

Tran ID Proto ID Len SID FN

Bad Diag Code

- Request is to Slave 01 Fn 8 Diagnostics
- Diagnostic code FF sent

# Error Response

```
0000  02 00 00 00 45 00 00 3d f0 fa 40 00 40 06 00 00  ....E..= ..@.@...
0010  7f 00 00 01 7f 00 00 01 01 f6 c0 08 5c f0 89 dc  ....\...
0020  68 4e c4 02 80 18 ff ff fe 31 00 00 01 01 08 0a  hN..... .1.....
0030  21 ba 60 c4 21 ba 60 c4 00 00 00 00 00 03 01 88  !.~!.~. ....
0040  03
```

03

00 00 00 00 00 03 01 88

Tran ID Proto ID Len SID FN

Error Code (data)

- Function code is 0x88 or 0x08 + 0x80
- Specific Error codes are returned in data field
- Error 0x03 is Illegal data value

# Errors are the Key

- When an improper SID is sent
  - The slave will not respond
  - The slave will respond with FN+0x80
- When a proper SID is sent
  - The slave will respond with a valid response
- This forms the basis for mapping

# ModScan

- Modscan Scans the IP range provided for open 503 ports
- When an open port is found it finds the SID via brute force
- By default it stops after first discovered SID
- Output in “IP:Port\tSID” format

# Options

- `-p` PORT (502)
- `-t` TIMEOUT socket timeout (100 mills)
- `-a` `--aggressive` Aggressive Mode
- `-f` FUNCTION MODBUS Function Code (17)
- `--data` Data for use with `-f`
- `-v`, `-d` Verbose, Debug

# ModScan Demonstration

- Scanning our sample network
- A look at a pcap
- Demo of additional Options

# ModScan Project

- <http://modscan.googlecode.com>
- Uses
  - Security Network Enumeration
  - IDS/Network Monitoring Test
  - Asset Management
  - Bulk Commands

# Known Issues

- Really, Really Noisy
- Brute forcing all ports is inefficient
- Does not interpret Error Codes
  - Can generate false negatives
- Does not calculate Length
- TCP Checksum not properly calculated

# Planned Enhancements

- Interpret Error Codes
- Implement with SCAPY
- Additional Protocol Support
- Device Fingerprinting
  
- Anything cool someone suggests....

Questions?

# References

- <http://en.wikipedia.org/wiki/SCADA>
- <http://www.modbus.org/specs.php>
- <http://www.wingpath.co.uk>

# Contact Information

**Mark Bristow**

[mark.bristow@gmail.com](mailto:mark.bristow@gmail.com)

[modscan.googlecode.com](http://modscan.googlecode.com)