INDUSTRIAL ECHOES WRITEUP

Category: Forensics
Points: 300
CTF: NICCTF 2026
By: lelkiramkeel

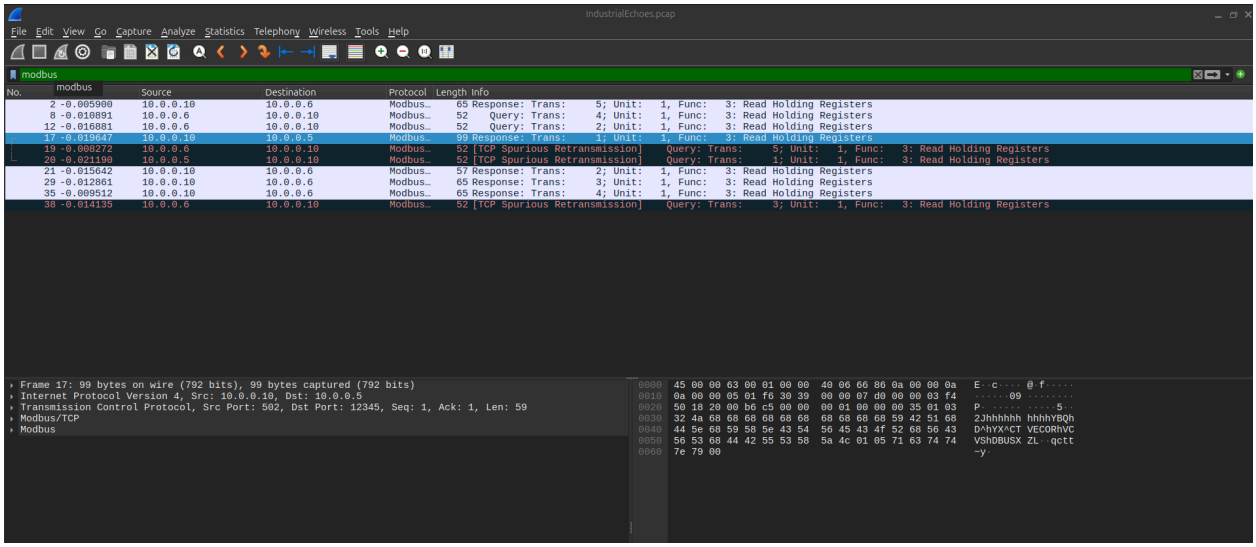
Challenge Description:

#modbus #ot #tcp

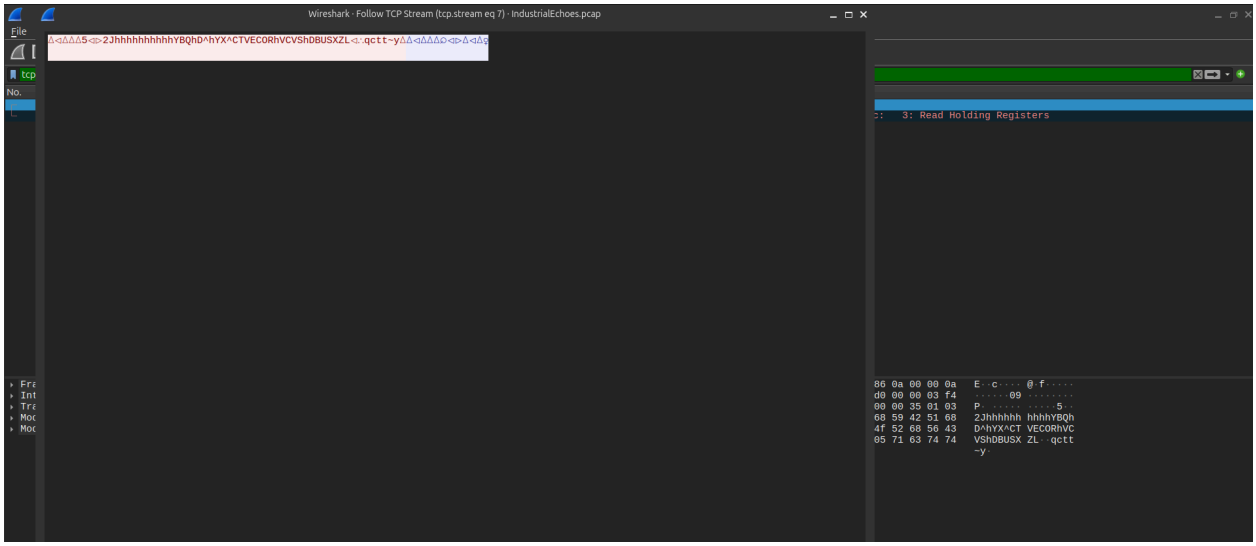
Our monitoring team captured a segment of traffic from an industrial automation network during a brief window of suspicious activity. The snapshot contains a mix of operational chatter, supervisory polling, and some unexpected protocol interactions.
However, analysts suspect that something more was transmitted during these exchanges, hidden in plain sight among legitimate industrial communications.
Your task is simple: recover any meaningful evidence that might confirm the presence of exfiltrated or encoded control messages.

Solution:

From the desc two clues appear, modbus and encoded.
Opening the file IndustrialEchoes.pcap file in wireshark, and using the filter modbus we can see the following captured packets.



One transaction stoods out, due to it's larger length compared to other transactions.
Following it's tcp stream, one can clearly see encoded text. After some research, the text is



clearly XOR.
The following python script is used to bruteforce XOR:

```
# XOR bruteforce

def xor_decrypt(data, key):
    return "".join(chr(b ^ key) for b in data)

payload =
".....5..2JhhhhhhhhhhYBQhD^hYX^CTVECORhVCVShDBUSXZL..qctt~y............."
data = payload.encode()
for key in range(256):
    decoded = xor_decrypt(data, key)

    if "NICCTF" in decoded or "FTCCIN" in decoded:
        print(f'{key}: {decoded}')
```

The flag is printed out, clearly showing it is reversed.
The following bash script generates the flag:

```
#!/usr/bin/env bash

# This is a bash script to obtain the flag of the challenge
IndustrialEchoes1
# It gets the flag
# print it out
```

```
# store it in a flag.txt
#

if [ -f "xor_bruteforce.py" ]; then
   #echo "file: xor_bruteforce.py exists"
   flag=$(python3 xor_bruteforce.py | rev)
   echo ${flag}

   echo ${flag} > flag.txt
else
   echo "ERROR: file: xor_bruteforce.py doesnot exist"
fi
echo "done"
```

The flag is stored in a flag.txt file.
NICCTF{modbus_data_extraction_is_fun_____}

Conclusion
The challenge necessitated industrial protocol analysis and the application of multi-layer obfuscation techniques.

Successful resolution required a thorough comprehension of network packet structure and the intricacies of XOR encryption, specifically the methodology for its decryption.