Reverse Engineering GoodWe

Danny Robson

GoodWe Wifi Protocols

Adventures in Frustration: Part 1 of N

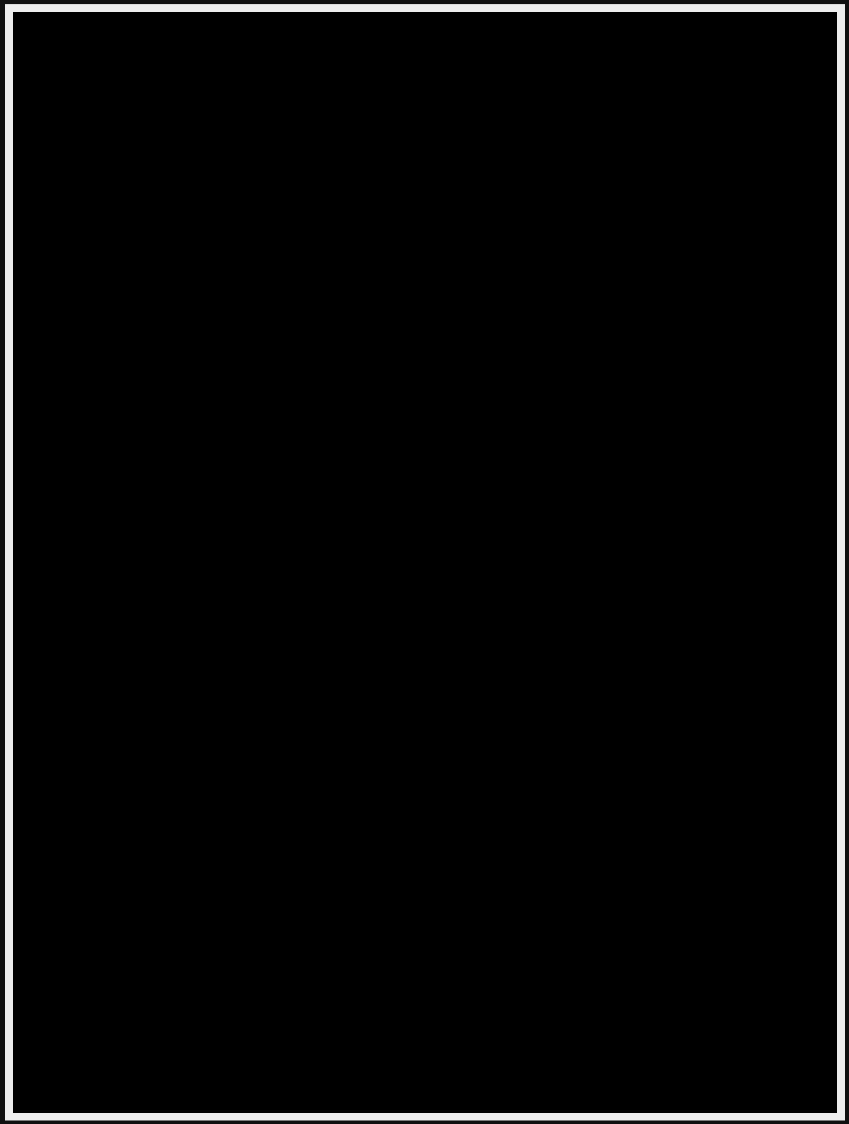
Caveats

To avoid disappointment: I haven’t cracked it yet.

But it’s an interesting journey so far.

Inverters

Power Monitor

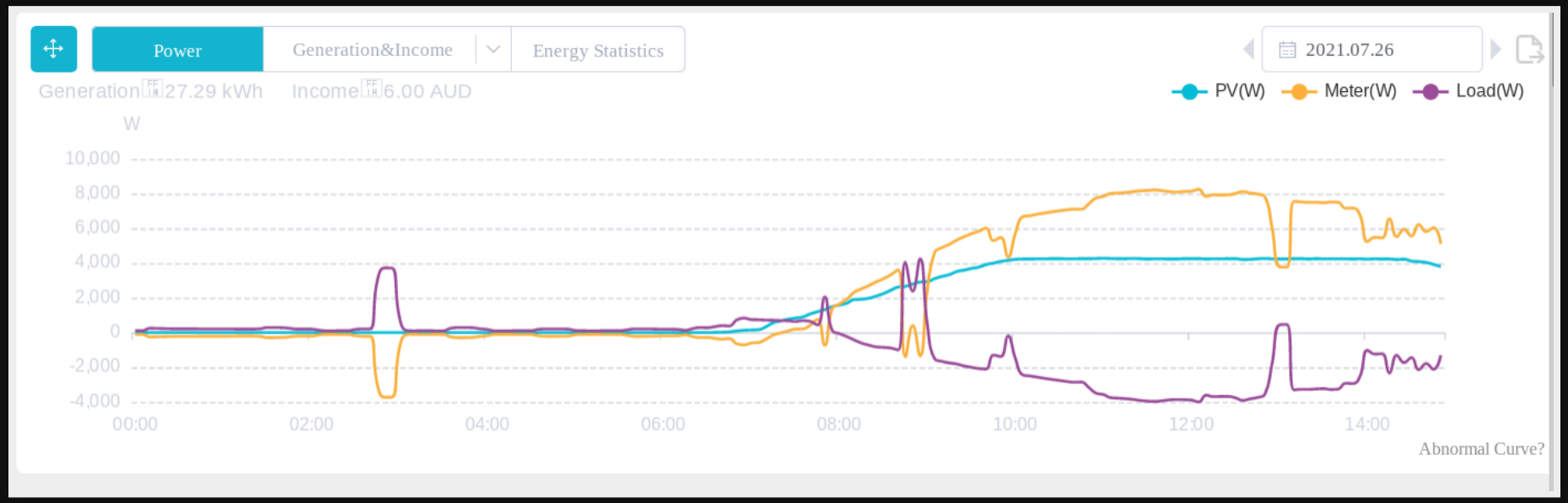


Hardware

- 2x inverters
- 1x energy logger

All connected to our `IOT' WiFi network...

Software


Power
Generation&Income
Energy Statistics
2021.07.26
Generation 27.29 kWh
Income 6.00 AUD
PV(W)
Meter(W)
Load(W)
W
10,000
8,000
6,000
4,000
2,000
0
-2,000
-4,000
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
Abnormal Curve?

System

- Hardware requires an internet connection.
- Stores all data elsewhere.
- The data is visualised incorrectly.

And it’s me

What about self-hosting..?

HomeAssistant

mletenay/home-assistant-goodwe-inverter

direct queries for older models

TimSoethout/goodwe-sems-home-assistant

scrapes the portal

mletenay can detect the devices but can’t query them.

ModBus

- AA55C07F0102000241
- 7F03753100280409
- 197d0001000dff045e5030303656465 7f6e60d

ModBus

Consists of: read/write, slaveid, address, length.

It worked for someone else. I just need to guess the correct parameters.

And I *do* know how to write a for loop...

ModBus

It’s a bust.

Only the energy monitor responds.

And it’s always returns a zero to one specific battery query...

SolarMan/Omnik

- Broadcast "WIFIKIT-214028-READ" on UDP 48899
- Receive a response with the serial number
- Send a request with the serial number and get a status result

SolarMan/Omnik

It’s a bust.

GoodWe won’t respond to anything beyond the broadcast...

Getting frustrated...

Start over

It’s on my network: what if I pretend to be a GoodWe server?

MITM

The devices use DHCP: what if I set the default route to a Pi and log everything?

dnsmasq

```
dhcp-range=tag:iot,192.168.1.225,192.168.1.254,1h
dhcp-host=xx:xx:xx:xx:xx:xx,set:iot
dhcp-option=tag:iot,option:router,192.168.1.100
dhcp-option=tag:iot,option:dns-server,192.168.1.100
```

nft

```
table ip nat {
  chain postrouting {
    type nat hook postrouting priority 100;
    ip saddr 192.168.1.0/24 oif "eth0" snat to 192.168.1.100;
  }
}
```

tcpdump

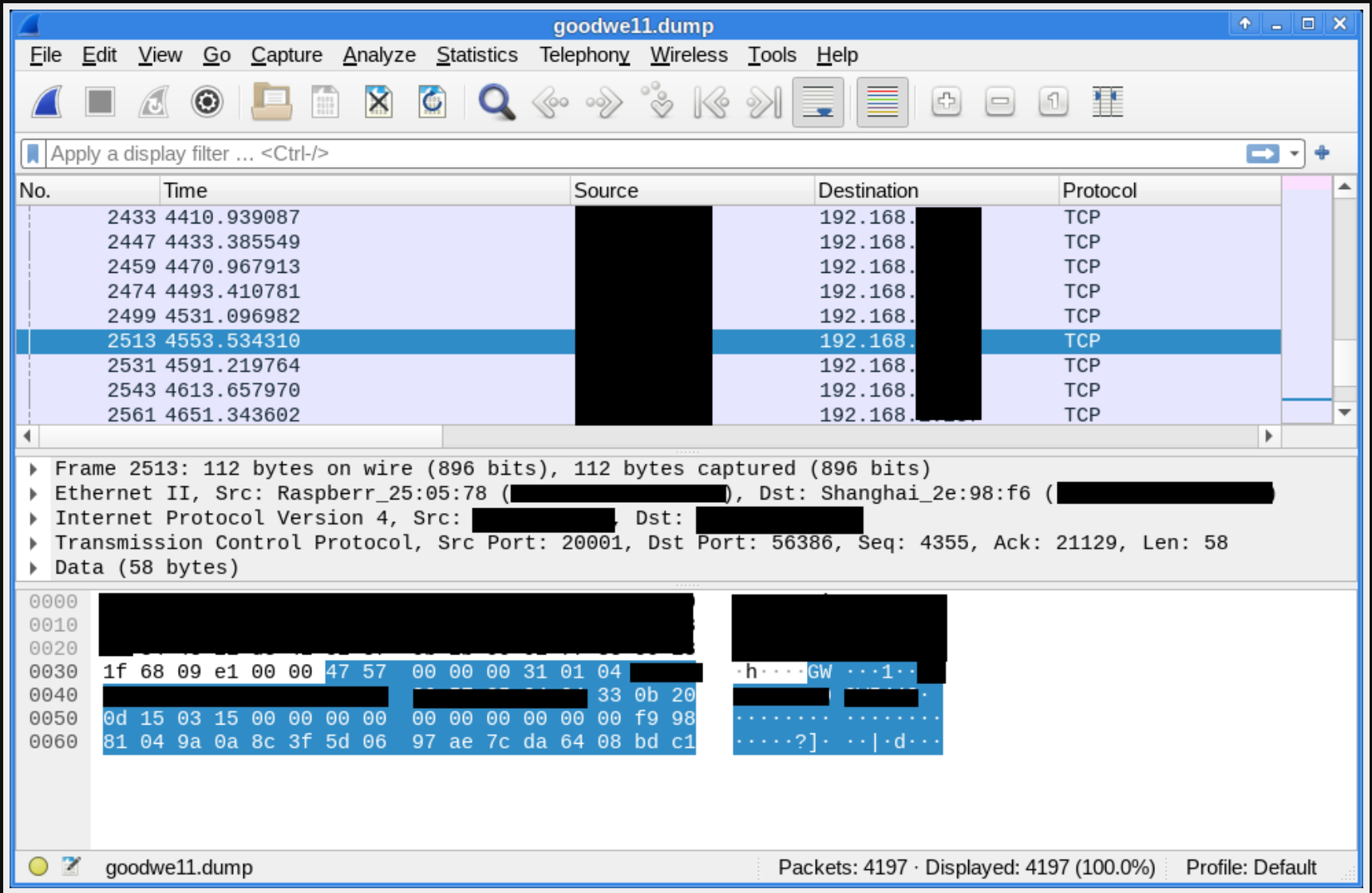
```
tcpdump "ether host xx:xx:xx:xx:xx:xx" -w goodwe.dump
```

Wireshark

Often the best first analysis is to:

1. sort by source IP
2. then sort by time (or size then time)
3. hold the 'down arrow' key
4. defocus your eyes
5. watch for patterns

Wireshark


goodwe11.dump
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
No. Time Source Destination Protocol
2433 4410.939087 192.168. TCP
2447 4433.385549 192.168. TCP
2459 4470.967913 192.168. TCP
2474 4493.410781 192.168. TCP
2499 4531.096982 192.168. TCP
2513 4553.534310 192.168. TCP
2531 4591.219764 192.168. TCP
2543 4613.657970 192.168. TCP
2561 4651.343602 192.168. TCP
Frame 2513: 112 bytes on wire (896 bits), 112 bytes captured (896 bits)
Ethernet II, Src: Raspberr_25:05:78 ( ), Dst: Shanghai_2e:98:f6 ( )
Internet Protocol Version 4, Src: , Dst:
Transmission Control Protocol, Src Port: 20001, Dst Port: 56386, Seq: 4355, Ack: 21129, Len: 58
Data (58 bytes)
0000
0010
0020
0030 1f 68 09 e1 00 00 47 57 00 00 00 31 01 04 ·h····GW ···1··
0040 33 0b 20 ·
0050 0d 15 03 15 00 00 00 00 00 00 00 00 00 00 f9 98 ····· ·····
0060 81 04 9a 0a 8c 3f 5d 06 97 ae 7c da 64 08 bd c1 ·····?]· ··|·d···
goodwe11.dump
Packets: 4197 · Displayed: 4197 (100.0%)
Profile: Default

Wireshark

Looks like it’s sending HTTP POST to a specific host every minute after sun up.

Packet analysis

Nothing obvious. So let’s write some code.

It’s likely reporting a series of numbers that change just a little each minute.

Try diffing bytes between packets

Packet analysis

That’s weird… it looks random…

Oh no…

Luck "saves" the day

- Months of extended internet outages at our place
- After a reconnect *many* buffered messages were sent at once
- If they were sent during the same second the first 256bits of "random" were identical

"Luck" "saves" the day

- Probably encrypted with AES256 using the time as a nonce.
- *This* is why you don’t write your own crypto.

Start over

It’s on my network, so: what can *I* do to the devices locally…

nmap

```
Nmap scan report for HF-A21 (xx.xx.xx.xx)
Host is up (0.011s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
23/tcp open  telnet
80/tcp open  http
MAC Address: xx:xx:xx:xx:xx:xx (Shanghai High-Flying Electronics
```

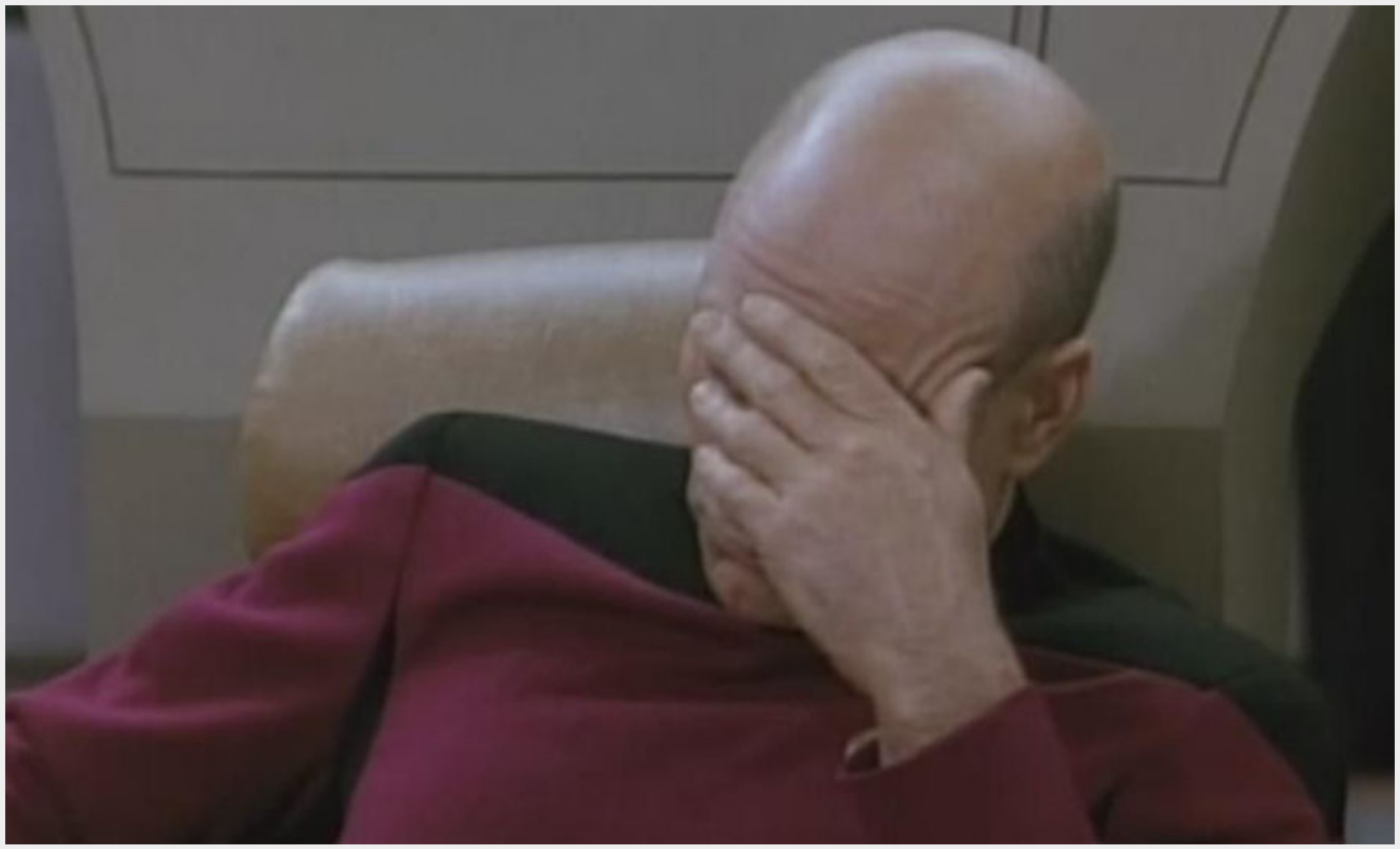
telnet

```
danny@sybil ✓ ~ $ telnet
telnet> open 192.168.1.xx
Trying 192.168.1.xx...
Connected to 192.168.1.xx.
Login as:
```

telnet

```
danny@sybil ✓ ~ $ telnet
telnet> open 192.168.1.238
Trying 192.168.1.238...
Connected to 192.168.1.238.
Login as:admin
Password:admin
MCMD>
```

telnet



telnet

```
CFG> cd mft
MFT>bootcfg
MHW_VER=x.x.x.x
HW_SN=xxxxxx
HW_APP=x
HW_VLAN=x
HW_INTF=x
...
```

telnet

- Not much that looks immediately useful...
- But I *can* see log statements when some queries are sent to the device

telnet

```
OS> help spi
SPI <rd/wr/er> <addr> [len]
```

telnet

So, we can fetch 4 bytes at an arbitrary address.

And I know `for` loops...

100 lines of Python later

```
./read.py ${ip} > spi.img
```

binwalk

```
danny@sybil ✓ ~/src/goodwe $ binwalk spi.img
DECIMAL       HEXADECIMAL     DESCRIPTION

71744         0x11840         U-Boot version string, "U-Boot 1.1.
327680        0x50000         uImage header, header size: 64 byte
337024        0x52480         LZMA compressed data, properties: 0
```

strings

A really useful tool that lists printable strings in a binary file.

```
strings spi.img | grep goodwe | less
```

ftp

- Don’t encode your FTP site and password in plaintext.
- Turns out it has multiple gigabytes of firmeware images

Ghidra

- The NSA released an open source reverse engineering / decompilation tool
- I have some (many) firmware images
- ...

Ghidra

CodeBrowser(2): goodwe-ns:/spi.dat

File Edit Analysis Graph Navigation Search Select Tools Window Help

Program Trees

- spi.dat
  - b0000000
  - dup

Program Tree

Symbol Tree

- Imports
- Exports
- Functions
  - FUN_b0000...
  - FUN_b0001...
    - FUN_b000126c
    - FUN_b0001878
  - FUN_b0002...
    - FUN_b000269c
    - FUN_b00027c8

Filter:

Data Type Manager

- Data Types
  - BuiltInTypes
  - spi.dat
  - generic_clib

Filter:

Listing: spi.dat

```
b00027ac ff    ??    FFh
b00027ad ff    ??    FFh
b00027ae 02    ??    02h
b00027af 24    ??    24h    $
b00027b0 ff    ??    FFh
b00027b1 00    ??    00h
b00027b2 e2    ??    E2h
b00027b3 30    ??    30h    0
b00027b4 00    ??    00h
b00027b5 00    ??    00h
b00027b6 c2    ??    C2h
b00027b7 ac    ??    ACh
b00027b8 21    ??    21h    !
b00027b9 10    ??    10h
b00027ba 00    ??    00h
b00027bb 00    ??    00h
b00027bc 18    ??    18h
b00027bd 00    ??    00h
b00027be bf    ??    BFh
b00027bf 8f    ??    8Fh
b00027c0 08    ??    08h
b00027c1 00    ??    00h
b00027c2 e0    ??    E0h
b00027c3 03    ??    03h
b00027c4 20    ??    20h
b00027c5 00    ??    00h
b00027c6 bd    ??    BDh
b00027c7 27    ??    27h    '

                    **********************************************
                    *                          FUNCTION
                    **********************************************
                    undefined FUN_b00027c8()
                       assume gp = 0x151f0
                       assume t9 = 0x27c8
          undefined          v0:1          <RETURN>
```

Decompile: FUN_b00027c8 - (spi.dat)

```
/* WARNING: Globals starting with '_' overlap smaller symbols at

undefined4 FUN_b00027c8(int param_1,undefined4 param_2,int param_

{
  undefined4 uVar1;
  char cVar2;
  int iVar3;
  undefined4 local_18 [2];

  cVar2 = **(char **)(param_4 + 4);
  if ((cVar2 == 'r') || (cVar2 == 'R')) {
    if (param_3 == 3) {
      uVar1 = (*_DAT_000152a4)(*(undefined4 *)(param_4 + 8),0,10)
      iVar3 = 0;
      local_18[0] = 0;
      goto LAB_b00028b8;
    }
    cVar2 = **(char **)(param_4 + 4);
  }
  if (((cVar2 != 'w') && (cVar2 != 'W')) || (param_3 != 4)) {
    (*_DAT_000155c8)(_DAT_000151fc + 0x249c,*(undefined4 *)(param_
    return 1;
  }
  uVar1 = (*_DAT_000152a4)(*(undefined4 *)(param_4 + 8),0,10);
  local_18[0] = (*_DAT_000152a4)(*(undefined4 *)(param_4 + 0xc),0
  iVar3 = 1;
LAB_b00028b8:
  (*_DAT_000154d8)(iVar3,uVar1,local_18);
  if (iVar3 == 0) {
    (*_DAT_000155c8)(_DAT_000151fc + 0x24a8,uVar1,local_18[0]);
  }
  return 0;
}
```

Console - Scripting

b00027c8 | FUN_b00027c8 | lui gp,0x1

Start over

...

Future work

- Analyse the firmware
- Extract the encryption keys
- ...
- (Unlikely to profit)

Thanks