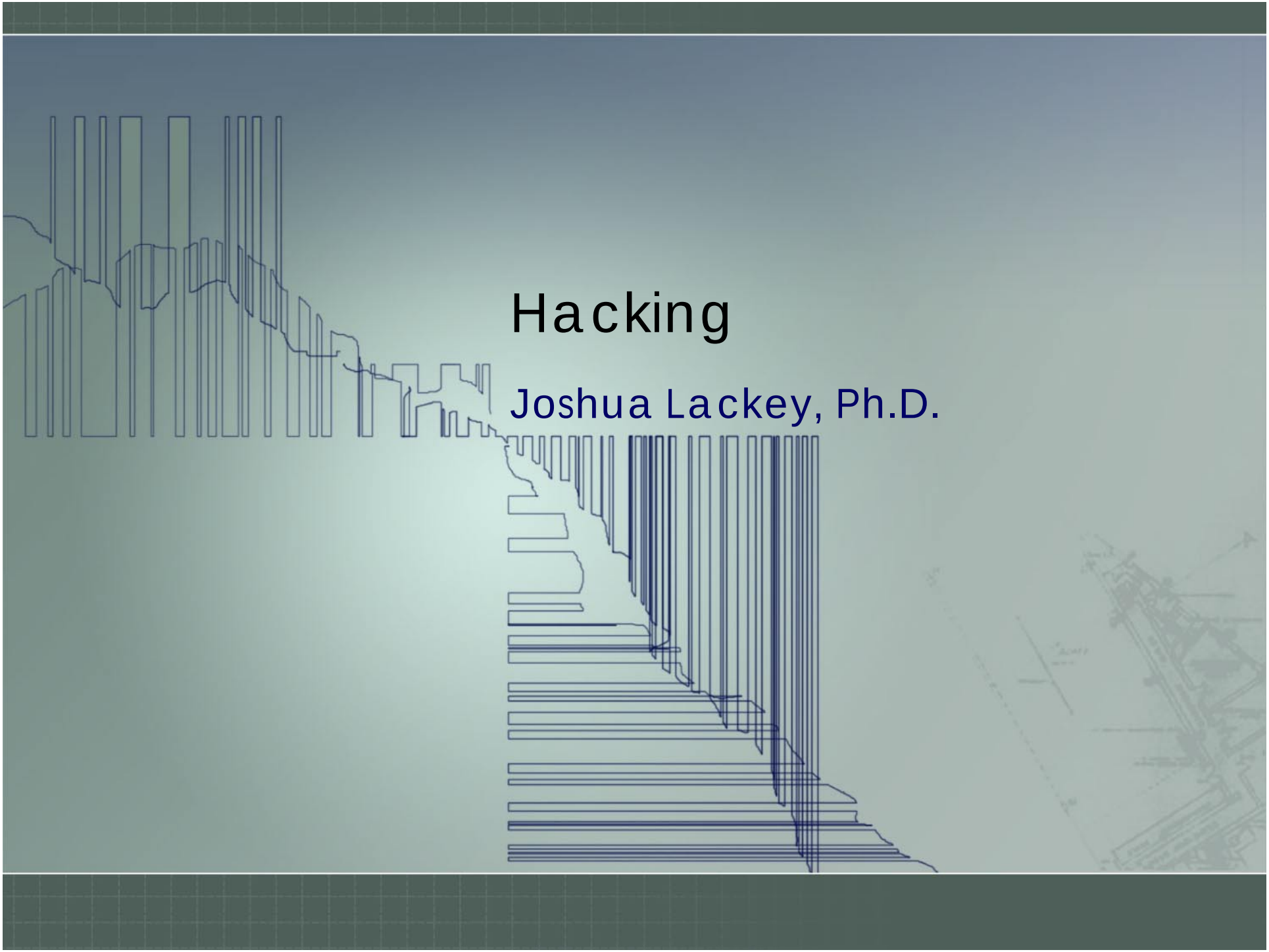

Hacking
Joshua Lackey, Ph.D.

Background

- Ph.D., Mathematics. University of Oregon. 1995 – 2000
- Senior Ethical Hacker. IBM Global Services. 1999 – 2005
- Security Software Developer. Microsoft SWI Attack Team. 2005 –

Hacking as a White Hat

Requirements

- Technical Talk
- One 50 minute lecture

Personal Requirements

- Not boring

Question

Why would anyone spend $1.5k – $2k per day for a penetration test?

Answer

- Cost/benefit
- Risk analysis
  - how?
- Example
  - an MSRC bulletin costs between $100k and $200k.
  - design review, threat model review, history of product/feature, training statistics feed into the risk analysis.
  - this determines if more work must be performed.

Answer

The goal of any penetration test or ethical hack is to determine the ***truth***.

Truth

Is what we believe, what we have been told actually true?

Is what we designed, what we implemented secure?

Truth

- Adversarial Situations
  - "of course we did this securely"
- Acquisitions
  - quality analysis
  - unknown environment
- Talent
  - "never even thought of that"

Truth

The best plans include security analysis in all phases of development.

- Design
  - Penetration testing during design phase provides feedback before implementation.
  - The worst flaws are design flaws.
- Implementation
  - Software developers who understand how to write secure code.

Truth

Does it really cost $1.5k – $2k per day per penetration tester?

For top-level penetration testers, these are the standard security consultant's fees.

The main reason is that the talent required is not so common.

Examples from work.

Problem:

- I cannot discuss any of my good examples.

Examples

Examples from my research.

- 802.11 Fragmentation Attack
- VW Key Fob
- GSM

Examples

- Most of what I'm going to speak about is works-in-progress.
- There will be a lot of questions and very few answers.

802.11 Fragmentation Attack

(This is finished research.)

Serious Design Flaw – trying to gauge how much this cost is difficult. (Especially since most people/companies haven't addressed this...)

Would have been extremely difficult to find in design phase anyway. (Although possible.)

Best previous attack:

Weaknesses in the Key Scheduling Algorithm of RC4. Fluhrer, Mantin, Shamir.

- Vendors countered by not using weak IVs.
- Unfortunately, this was not enough. (Although many thought it was.)

802.11 Fragmentation Attack

A vulnerability exists in the IEEE 802.11 protocol which allows an attacker the ability to transmit WEP encrypted packets without knowing the encryption key.

This vulnerability allows an attacker to decrypt packets as well.

This was disclosed to CERT on September 16, 2003.

RC4 Encryption

If we denote by $E_k(P)$ the encryption of the plain-text message P by the RC4 encryption method with key k, we have

$$E_k(P) = X + P$$

Where X is the pseudo-random bit-stream generated by the RC4 PRGA with key k .

And thus

$$E_k(P) + P = X$$

Logical Link Control Packets

The most common LLC/SNAP packet seen on an 802.11 network is the Ethernet type LLC with IP.

Explicitly, this packet consists of the following eight bytes.

```
P' = { 0xaa, 0xaa, 0x03, 0x00, 0x00, 0x00, 0x08, 0x00 }
```

Logical Link Control Packets

Each encrypted packet on an 802.11 network is encapsulated in a logical-link control packet.

That is, each packet P is the concatenation of P', given above, and some P''.

$$P = P' \; P''$$

Logical Link Control Packets

By the above comments on RC4, we can find the first eight bytes of the pseudo-random bit-stream X' generated by the key used to encrypt this packet,

$$X' = E_k(P') + P'$$

Because we know the plain-text P', we can encrypt any arbitrary eight bytes with key k. We have, for any eight byte text Q,

$$E_k(Q) = X' + Q$$

802.11 Fragmentation

Section 9.4 of the 1999 IEEE 802.11 protocol specification provides a method to fragment packets when needed. Moreover, each fragment is encrypted individually.

802.11 Fragmentation Attack

By transmitting packets in fragments, an attacker can inject arbitrary packets into a WEP encrypted 802.11 wireless network.

802.11 Fragmentation Attack Example

Capture a packet, including the 802.11 headers, off a WEP encrypted network.

```
08 41 02 01 00 04 5a 37 ee 75 00 0e 35 ea 75 17
00 00 24 50 da 11 00 01 55 f9 47 00 db 76 e1 66
14 cf 05 c5 51 06 95 41 70 06 2d 4f 96 0e 0a 01
3c 6f fc bd 38 a2 21 02 33 0c 50 f1 e9 ae a4 8a
5e 16 49 41
```

802.11 Fragmentation Attack Example

If we parse the 802.11 header, we find this packet contains the following.

```
type: data frame, data only
to_ds: 1, from_ds: 1, more_frag: 0,
retry: 0, pwr_mgt: 0, more_data: 0,
wep: 1, order: 0
dur: 102
a1:       00-04-5A-37-EE-75
a2:       00-0E-35-EA-75-17
a3:       00-00-24-50-DA-11
seq:      frag = 00, num = 0010
data:
      55 f9 47 00 db 76 e1 66 14 cf 05 c5 51 06 95 41
      70 06 2d 4f 96 0e 0a 01 3c 6f fc bd 38 a2 21 02
      33 0c 50 f1 e9 ae a4 8a 5e 16 49 41
```

802.11 Fragmentation Attack Example

The first 10 encrypted data bytes are:

```
db 76 e1 66 14 cf 05 c5 51 06
```

Assuming that we have a IPv4 packet with a Ethertype LLC/SNAP header, the plain-text data is:

```
aa aa 03 00 00 00 08 00 45 00
```

Therefore the first ten bytes of the pseudo-random bit-stream are derived as follows.

```
  db 76 e1 66 14 cf 05 c5 51 06
+ aa aa 03 00 00 00 08 00 45 00
------------------------------
  71 dc e2 66 14 cf 0d c5 14 06
```

802.11 Fragmentation Attack Example

Suppose we wish to transmit an ICMP echo request.

```
45 00 00 2c 7a 0f 00 00 ff 01 33 b9 01 02 03 04  E..,z.....3.....
0a 01 00 02 08 00 6d 81 5d 02 2f 96 69 6e 6a 65  ......m.]./.inje
63 74 65 64 20 70 61 63 6b 65 74 00              cted packet.
```

802.11 Fragmentation Attack Example

Break this packet into fragments.

```
fragment 0:
     data: aa aa 03 00 00 00
     crc : f2 bb 67 21

fragment 1:
     data: 08 00 45 00 00 2c
     crc : 22 e7 83 c3

fragment 2:
     data: 25 4c 00 00 ff 01
     crc : 8a 4d 83 9f

fragment 3:
     data: 88 7c 0a 01 00 02
     crc : a7 d1 72 ff
[…]
```

802.11 Fragmentation Attack Example

For each piece of fragmented data, encrypt with the pseudo-random bit stream and attach an 802.11 header.

```
fragment 0:
        type: data frame, data only
        to_ds: 1, from_ds: 0, more_frag: 1,
        retry: 0, pwr_mgt: 0, more_data: 0,
        wep: 1, order: 0
        dur: 0
        a1:         00-04-5A-37-EE-75
        a2:         00-0E-35-EA-75-17
        a3:         00-00-24-50-DA-11
        seq:        frag = 00, num = 0024
        data:
                55 f9 47 00 db 76 e1 66 14 cf ff 7e 73 27
```

802.11 Fragmentation Attack Example

Continue.

```
fragment 1:
        type: data frame, data only
        to_ds: 1, from_ds: 0, more_frag: 1,
        retry: 0, pwr_mgt: 0, more_data: 0,
        wep: 1, order: 0
        dur: 0
        a1:         00-04-5A-37-EE-75
        a2:         00-0E-35-EA-75-17
        a3:         00-00-24-50-DA-11
        seq:        frag = 01, num = 0024
        data:
                55 f9 47 00 79 dc a7 66 14 e3 2f 22 97 c5
```

802.11 Fragmentation Attack Example

Now transmit the fragments.

The access point will decrypt each fragment and combine them into a single decrypted packet and forward it to the destination.

802.11 Fragmentation Attack Example

I omitted quite a few details, but this is the attack. It has been verified to work against all tested access points. Understandable as all this is specified in the protocol.

For an excellent write-up of this attack, see Andrea Bittau's paper. (Better version that I co-authored is coming soon.)

http://www.toorcon.org/2005/slides/abittau/paper.pdf

Now to talk about some research that isn't finished.

But first, a small aside.

Once upon a time, radio was for hardware geeks.

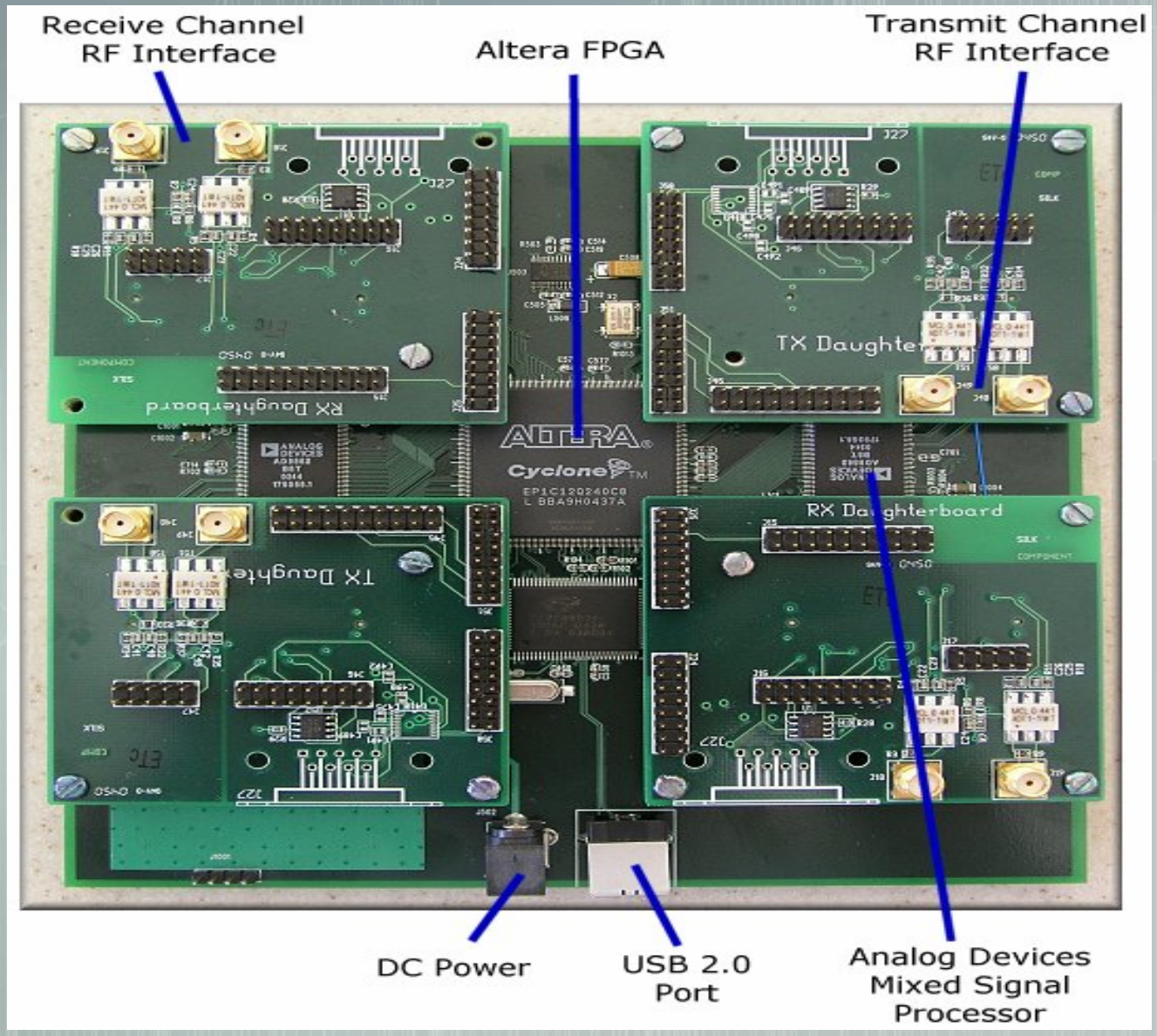
- Expensive equipment.
- For digital signals, *very expensive* equipment.
  - And sometimes not available to the general public.
- Of course custom hardware was always an option.

Software Radio

- Now we have inexpensive “front end” hardware.
- Uses your computer as the “back end” processor.
  - Every signal is now only a matter of software.
  - Free and increasingly full-featured SDR libraries.
- USRP
  - The Universal Software Radio Peripheral.

http://www.ettus.com

USRP


Receive Channel
RF Interface
Altera FPGA
Transmit Channel
RF Interface
RX Daughterboard
TX Daughterboard
ALTERA
Cyclone
DC Power
USB 2.0
Port
Analog Devices
Mixed Signal
Processor

USRP

- Two A/D D/A converters
  - A/D @ 64Msamples/sec
  - D/A @ 128Msamples/sec
- Altera FPGA
  - Field Programmable Gate Array
- Daughterboard interfaces
  - For RF integration
    - BasicRX and BasicTX – direct interface to AD/DA
    - TVRX – cable TV tuner interface
    - DBSRX – satellite TV tuner interface

Daughterboards

- BasicRX @ 64Msamples/sec
  - Receive frequencies up to 32MHz
    - Broadcast AM
    - Shortwave
  - Aliased frequencies with decreased signal strength.
    - Not so good for digital.
- BasicTX @ 128Msamples/sec
  - Transmit frequencies up to 64MHz

Daughterboards

- TVRX – cable TV tuner
  - Receive frequencies from 50MHz to 900MHz
    - Broadcast FM
    - Police (analog and digital)
    - Analog cellular phones (AMPS)
    - Digital mobile phones
      - DAMPS
      - GSM
      - iDEN
    - Etc, etc, etc.

Daughterboards

- DBSRX – satellite TV tuner
  - Receive frequencies from 800MHz to 2.5(+)GHz
    - GSM
    - CDMA
    - Bluetooth
    - 802.11
    - Hydrogen (Radio Astronomy)
    - Etc, etc, etc.

Software Radio

- Cheap hardware.
- Easily available.
- Highly flexible.

Examining the security of complex wireless protocols is now possible for the independent researcher.

We will gradually see more and more of wireless protocol vulnerabilities announced.

Examples

Volkswagen Key Fob

Every day I unlock my car with a radio. How secure is this? I'm sure if we asked we would be told that, "of course this is secure."

What is the truth?

What is the

- Algorithm?
- Quality of PRNG? (If used.)

First step is to gather data.

Examples

Volkswagen Key Fob

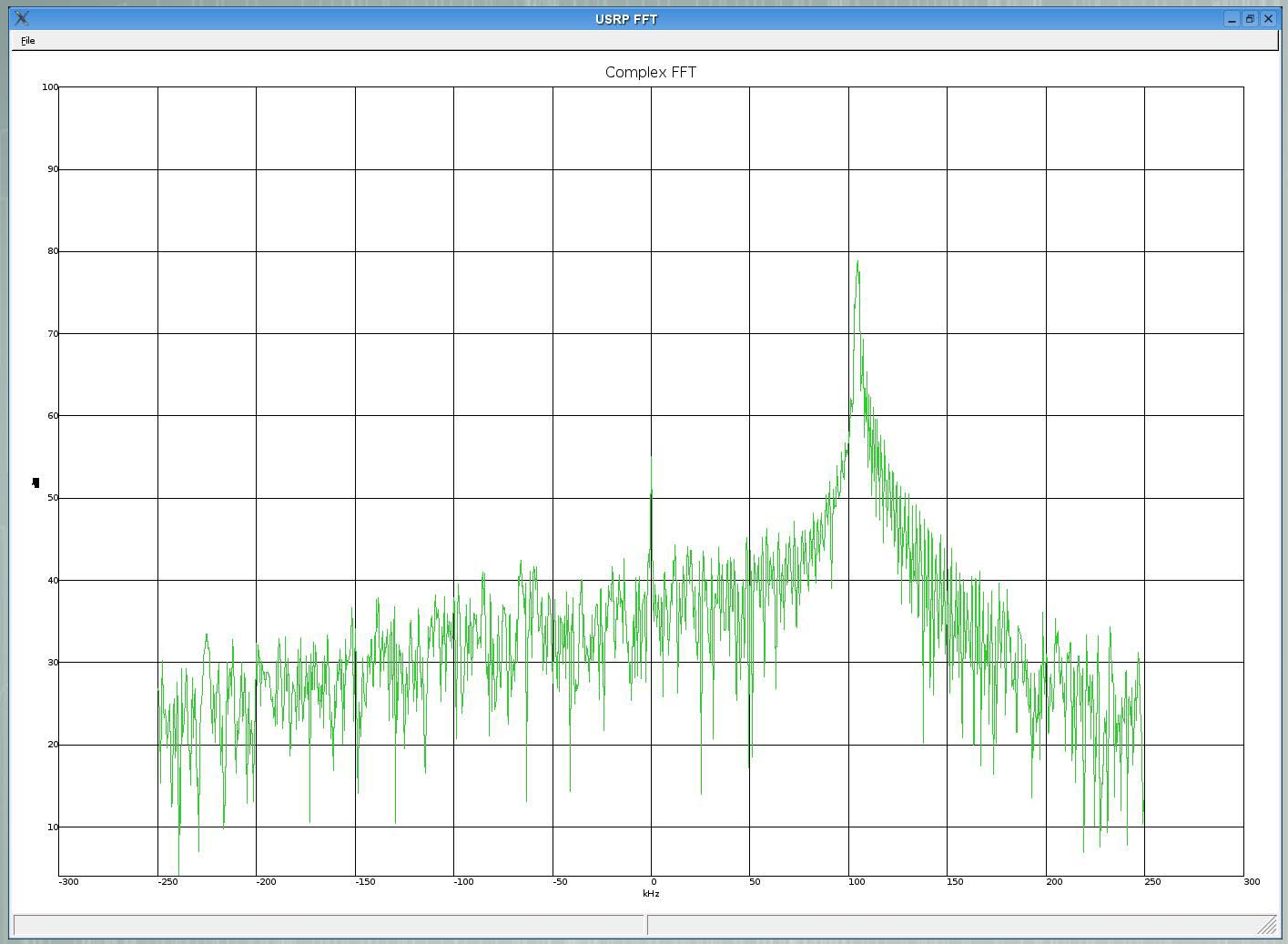
Find key fob transmit frequency

- FFT signal search
- Frequency grabber
- FCC ID Search:

https://gullfoss2.fcc.gov/prod/oet/cf/eas/reports/GenericSearch.cfm

VW Key Fob

- FFT Signal Search



VW Key Fob

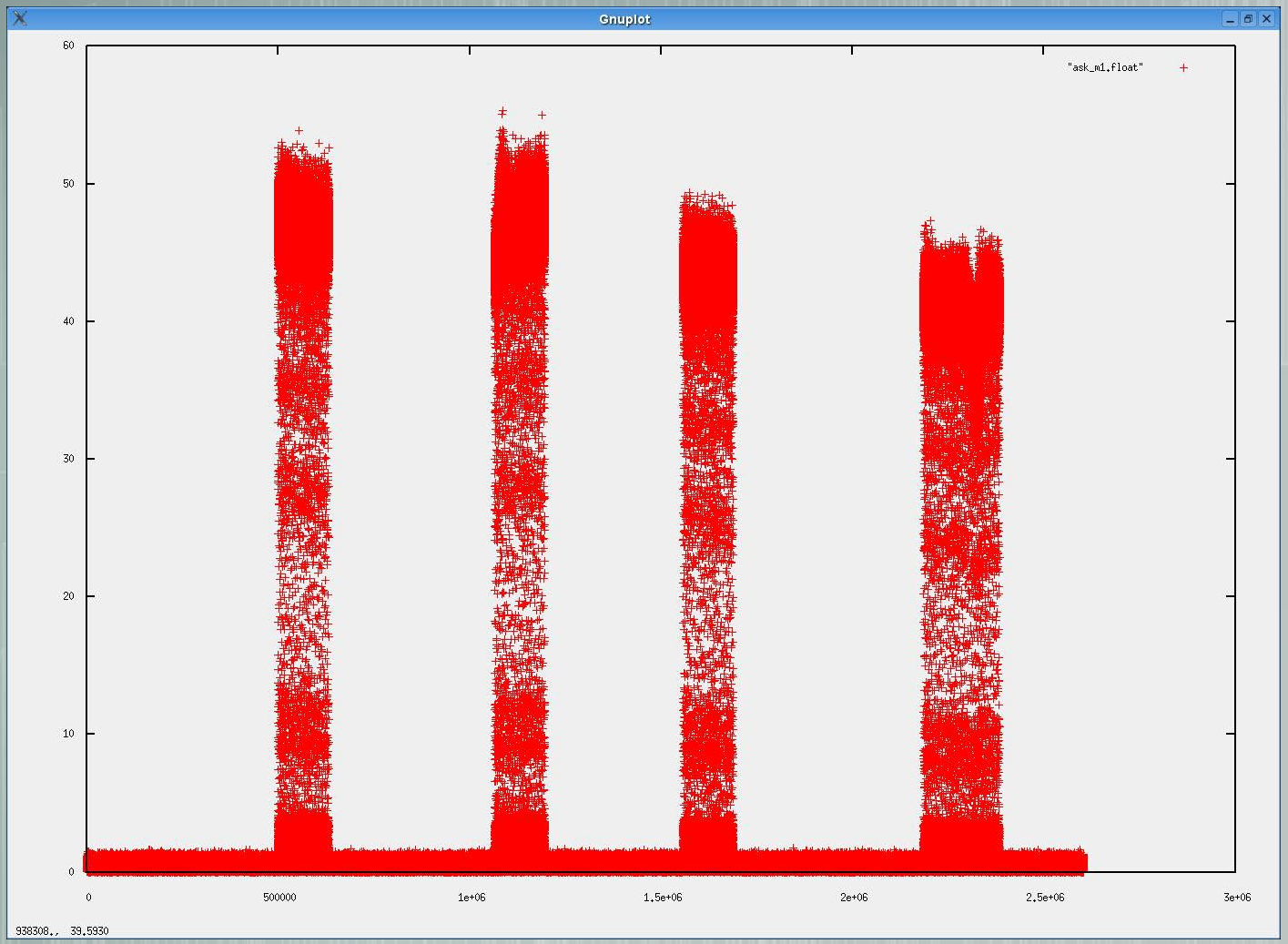
- FCC ID Search
  - Get FCC ID from device.
  - Grantee code is first three letters.
    - VW Key Fob: NBG
  - Frequency is 315MHz.
  - Modulation type is A1D
    - Amplitude modulation data transmission, double sideband, without using a modulating subcarrier.

VW Key Fob

- Modulation
  - FCC gave us modulation.
  - Can recognize different modulation types from FFT and raw signal.
    - Estimate bandwidth. Filter. Examine closely.

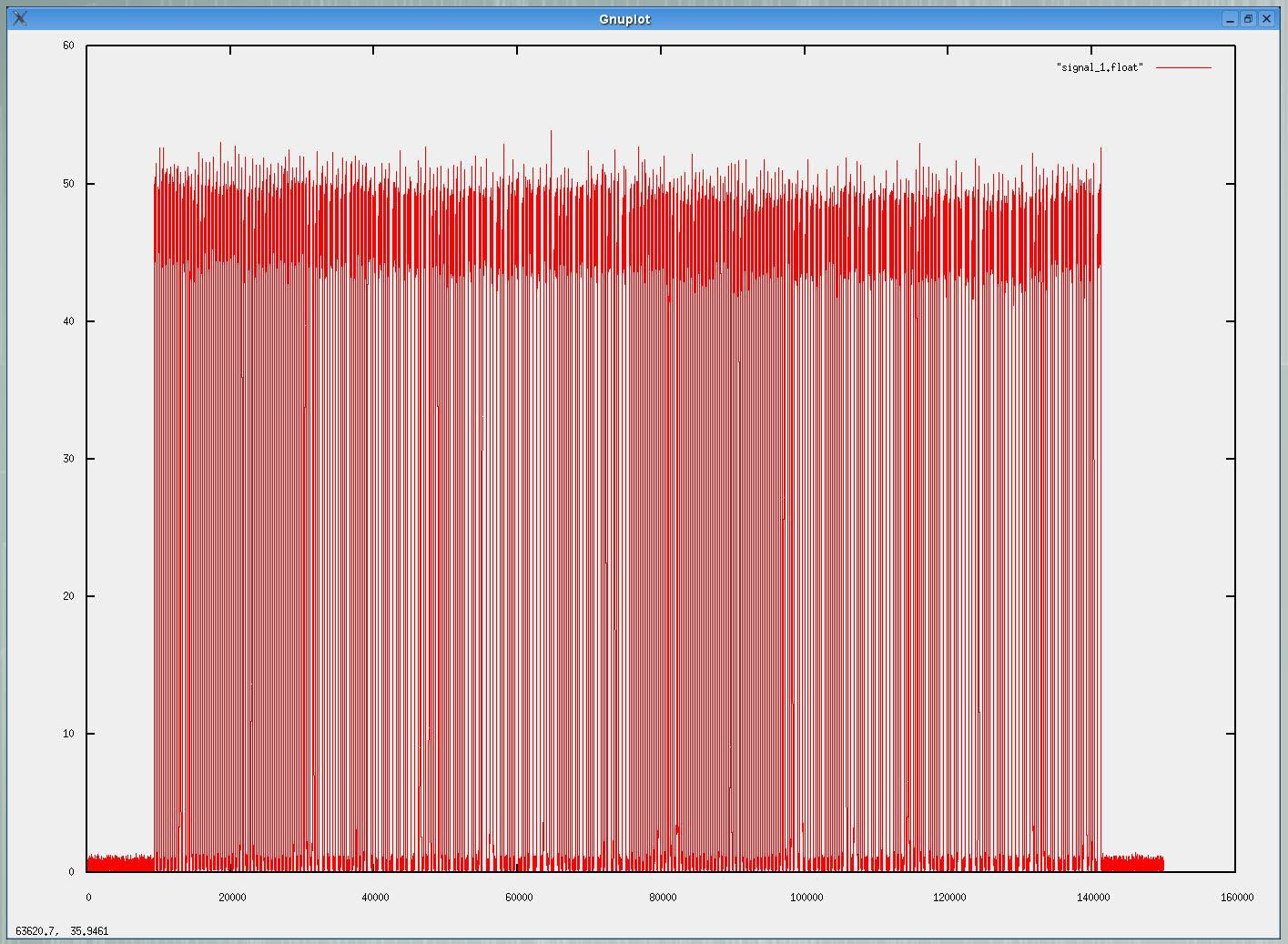
Capture signal (amplitude demod)

VW Key Fob



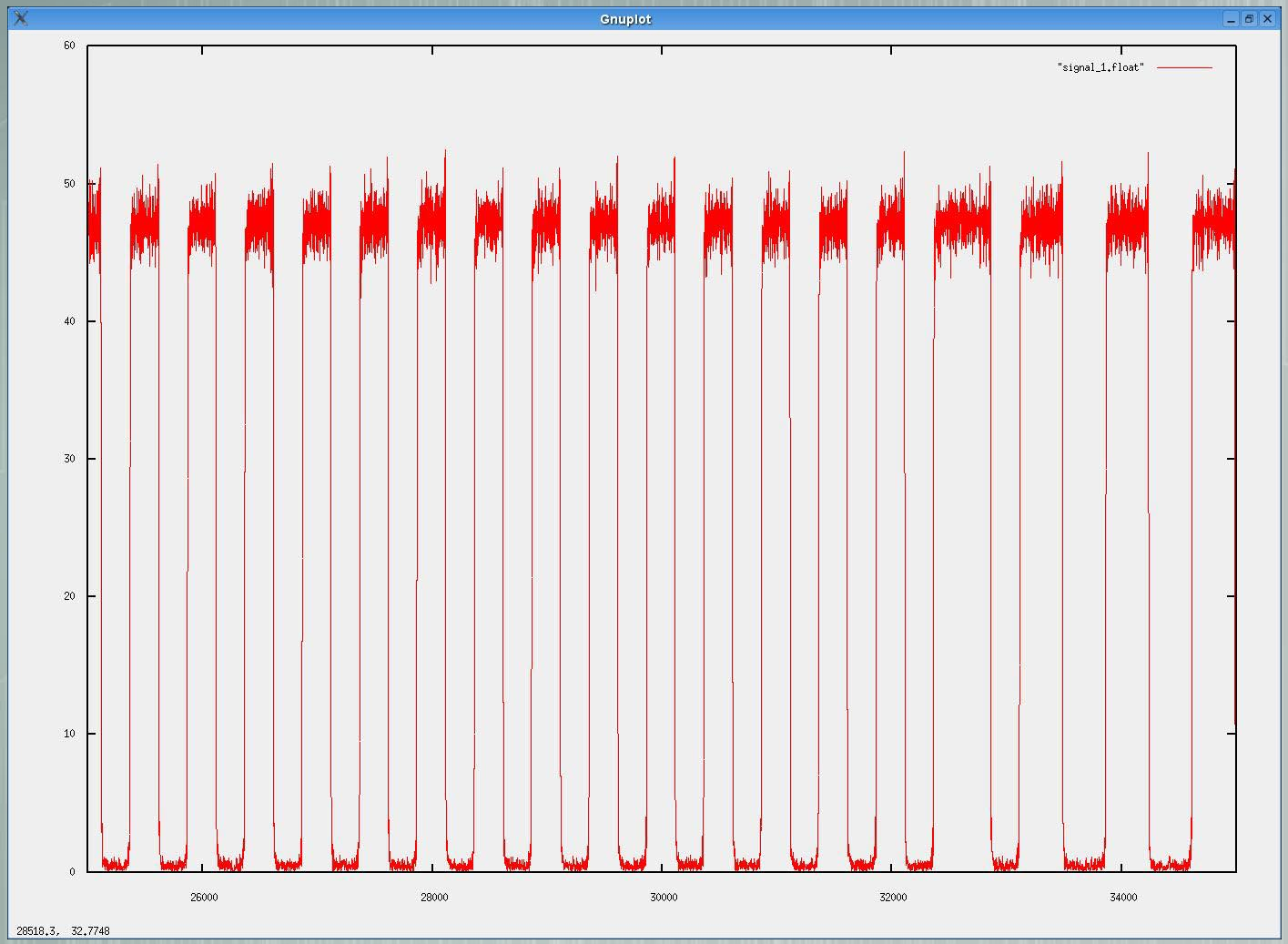
VW Key Fob

First Signal



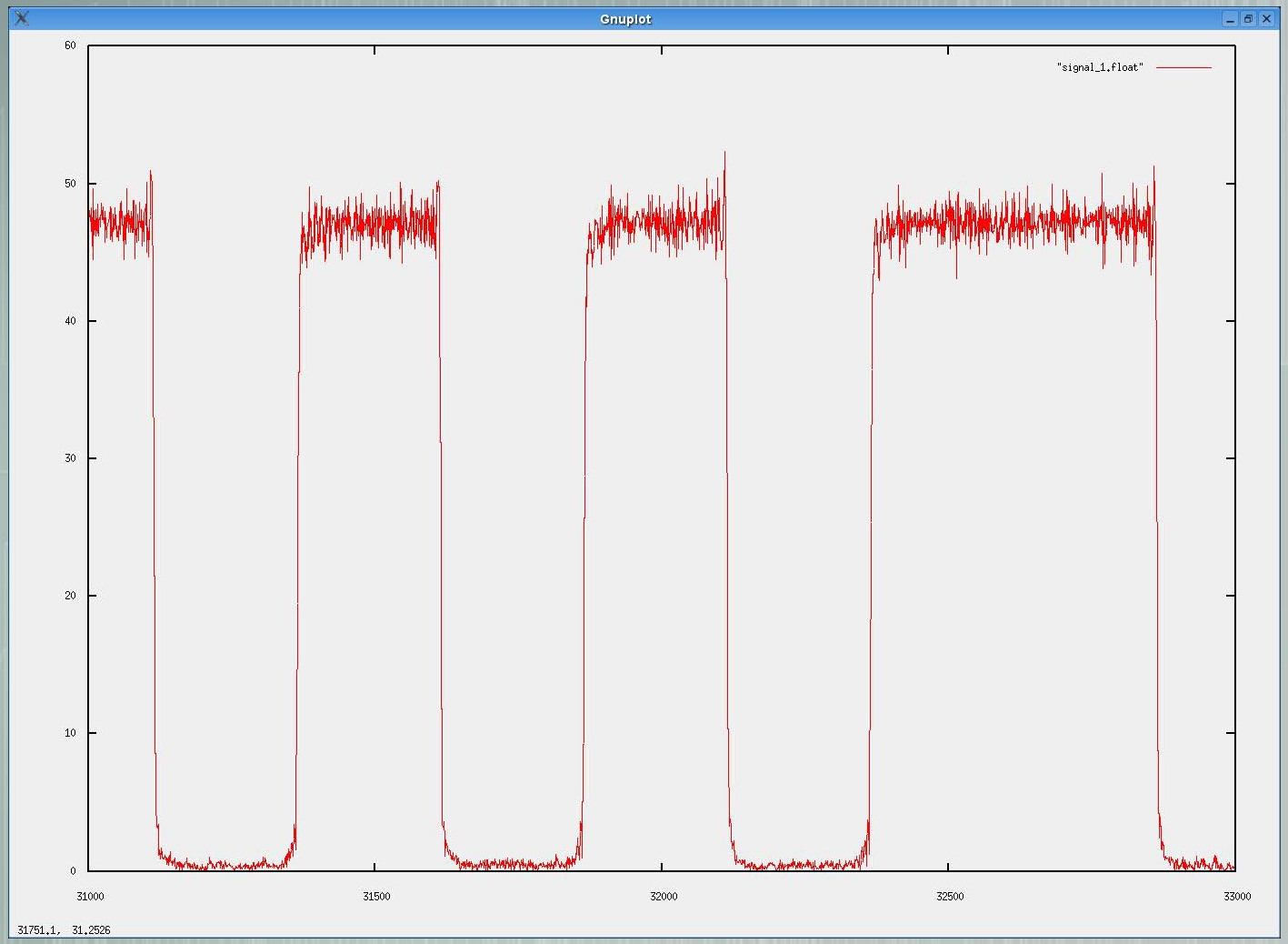
VW Key Fob

Samples 25000 – 35000 of first signal



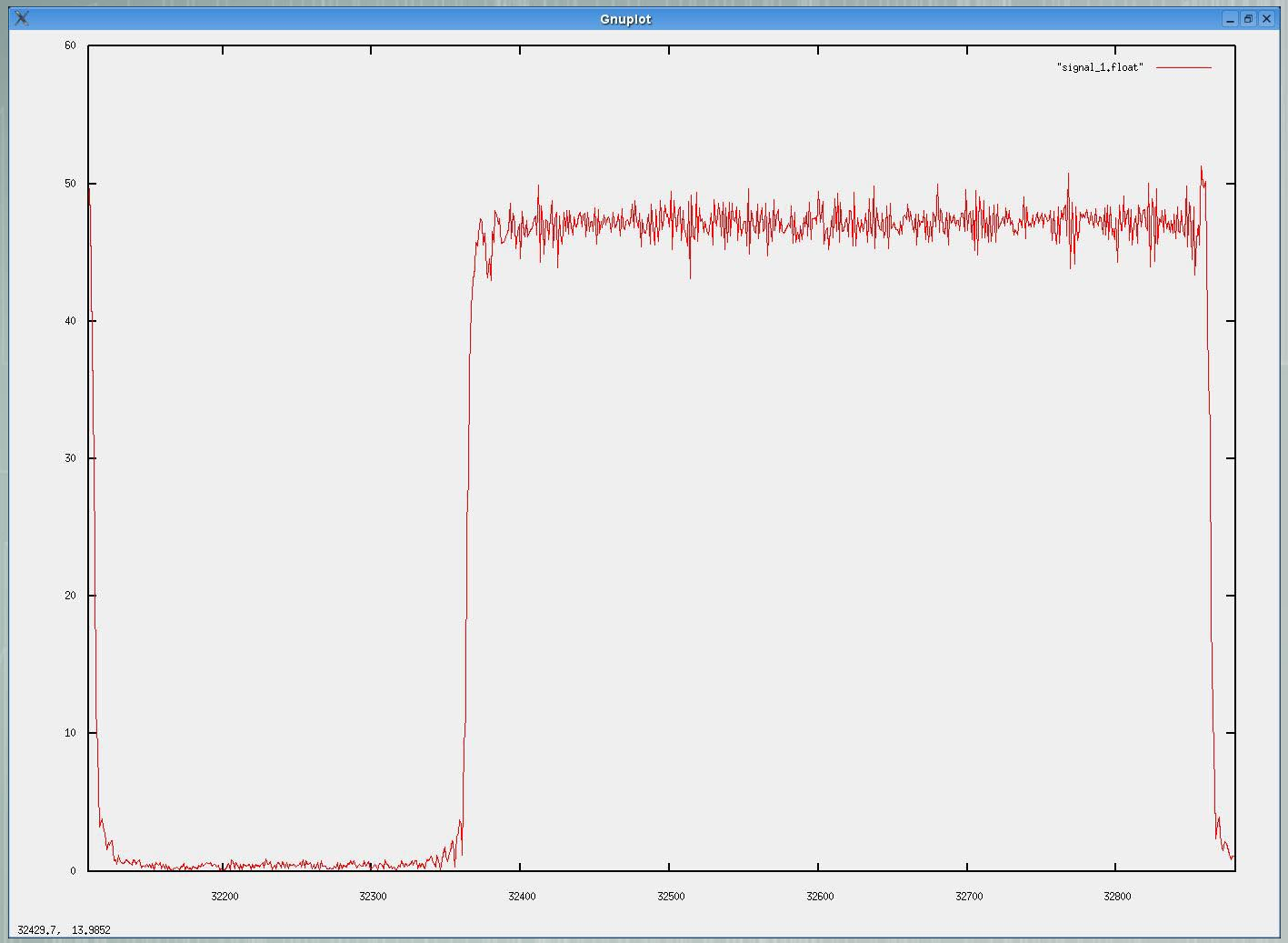
VW Key Fob

Samples 31000 – 33000 of first signal



VW Key Fob

Samples 32110 – 32880 of first signal



VW Key Fob

Sampled at 500kHz

- Holds low for 250 samples – .5ms
- Holds high for 500 samples – 1ms

For initial purposes

- Symbol length is .5ms
- Low is 0
- High is 1

So the sample we were looking at was

011

VW Key Fob

Demod

- transmit Frequency
- signal bandwidth
- guess at symbol modulation

Now just write some software!

Examined ~100 examples. No repeats although there are definite patterns.

Next steps

- Probably easiest thing to do would be to examine the demoded data for statistical patterns. (Diehard)
- See Bindview paper on strange attractors in TCP sequence numbers.

http://www.bindview.com/Services/Razor/Papers/2001/tcpseq.cfm

GSM

Documentation is very good and design flaws can be identified there. What about implementation flaws?

- What encryption does my phone use?
  - A5/1 and A5/2 hacked.
  - Actually, how do I know I'm using any encryption at all?
- How about random numbers?
- How about man-in-the-middle attacks?
  - Requires transmit and so probably illegal to test.

Next

GSM

Find transmit frequency.

- FFT Signal Search
  - just knowing the bandwidth is actually good enough
  - FCC Search for towers in your area
  - Documentation

http://www.3gpp.org/specs/numbering.htm

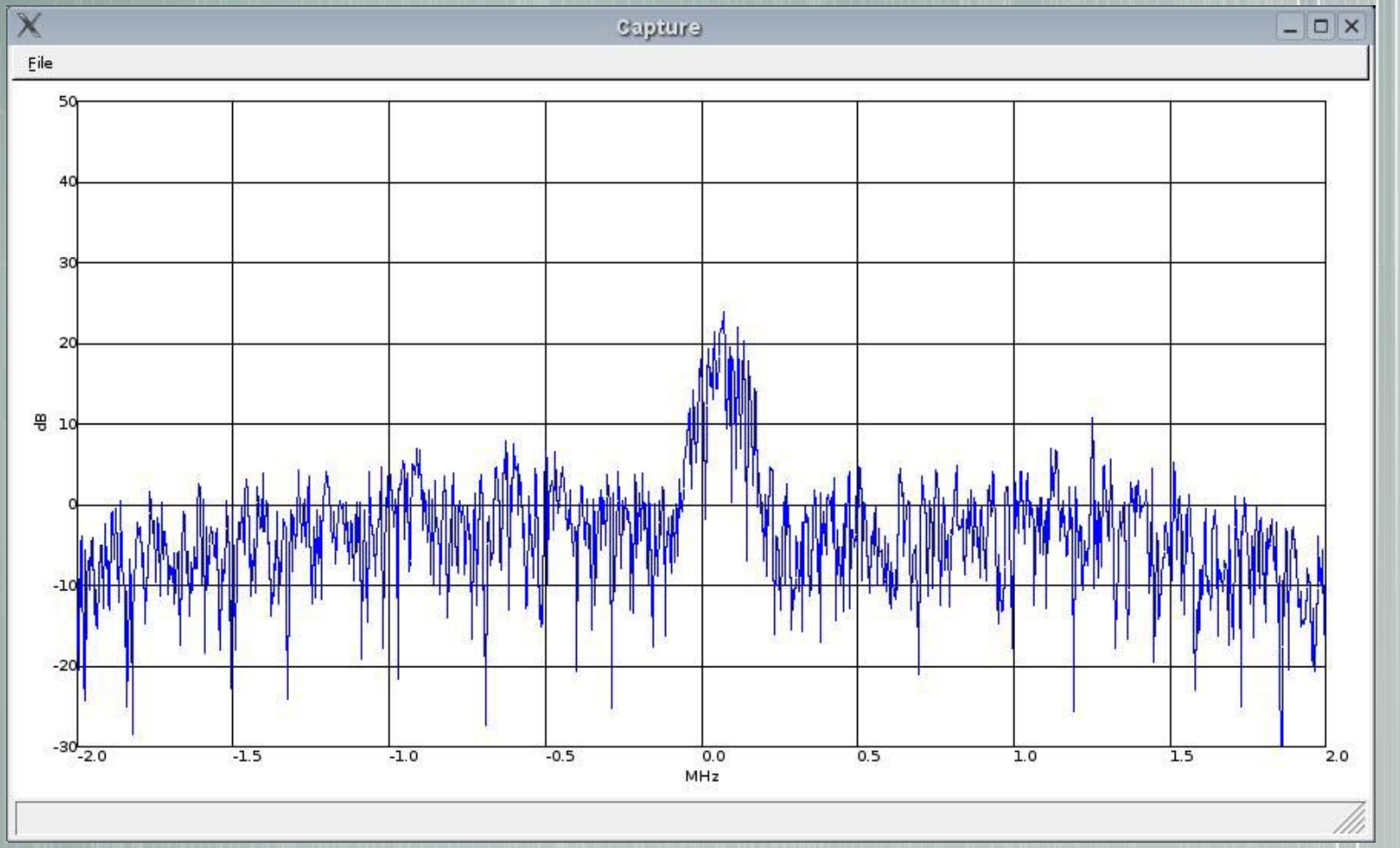
GSM

Turns out I have two strong signals reachable from my computer room.

One at 1.9474GHz and the other at 1.9468GHz.

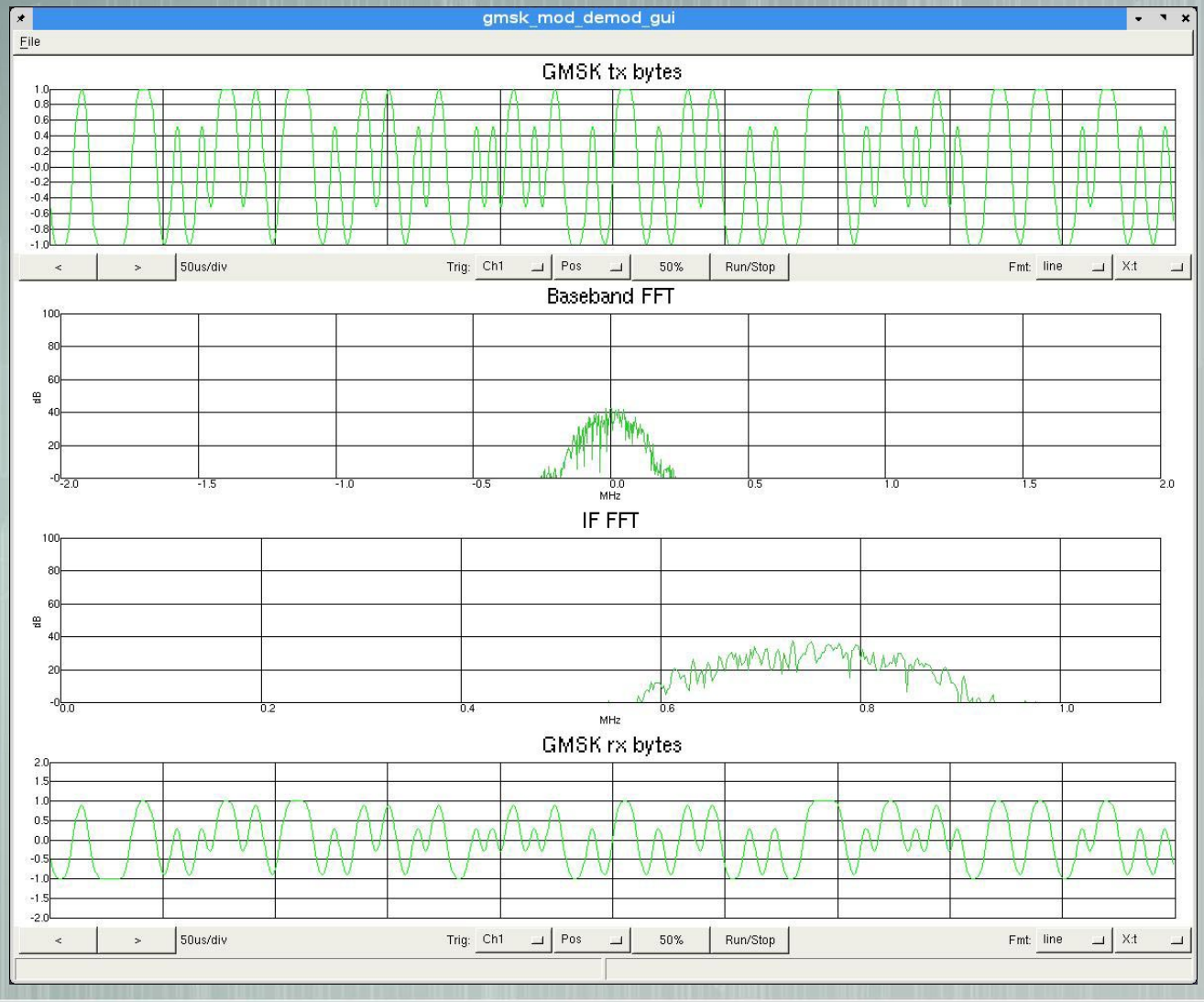
GSM

GSM Tower at 1.9474GHz with 1MHz DBS filter.



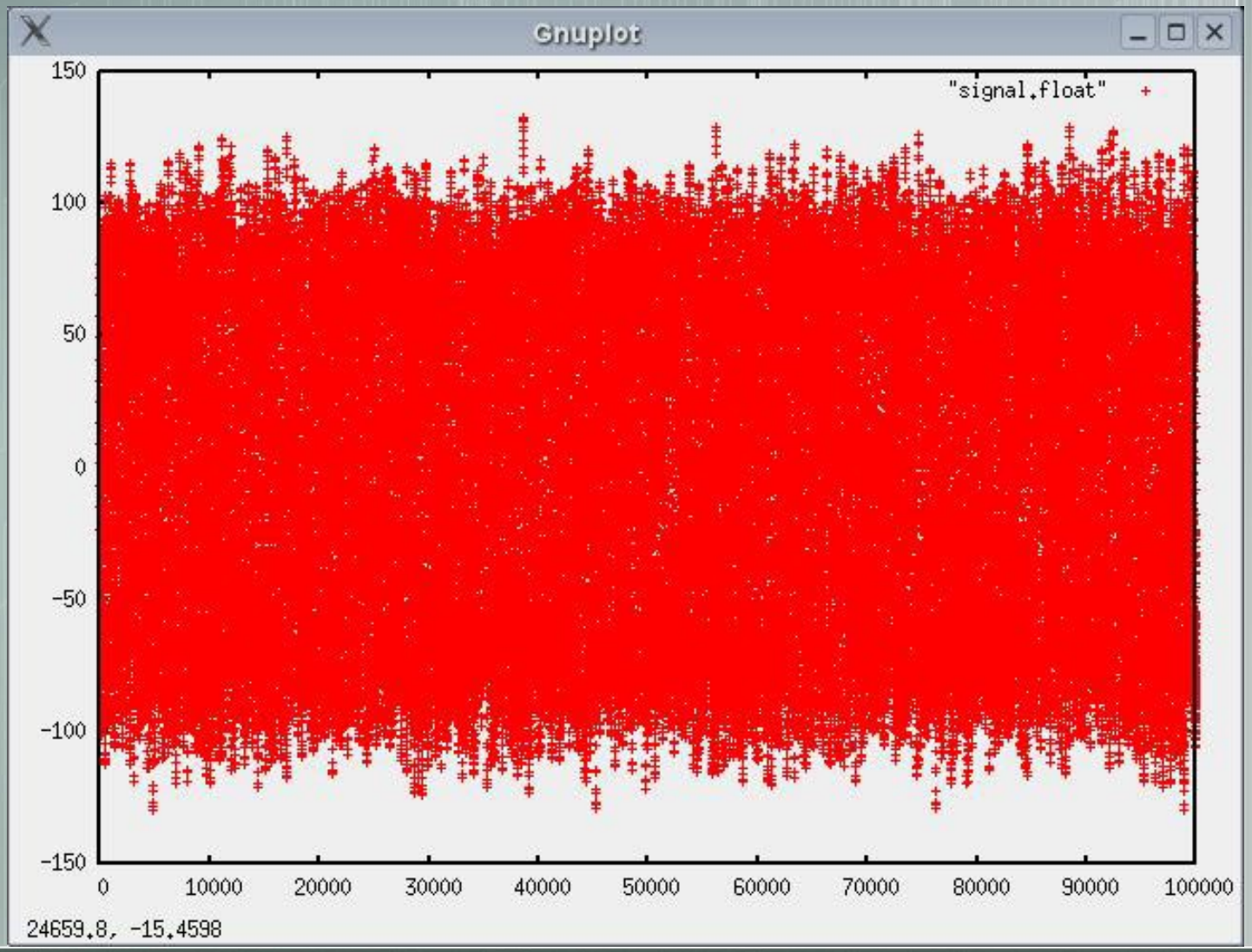
Modulation type is GMSK (or 8PSK)

GSM



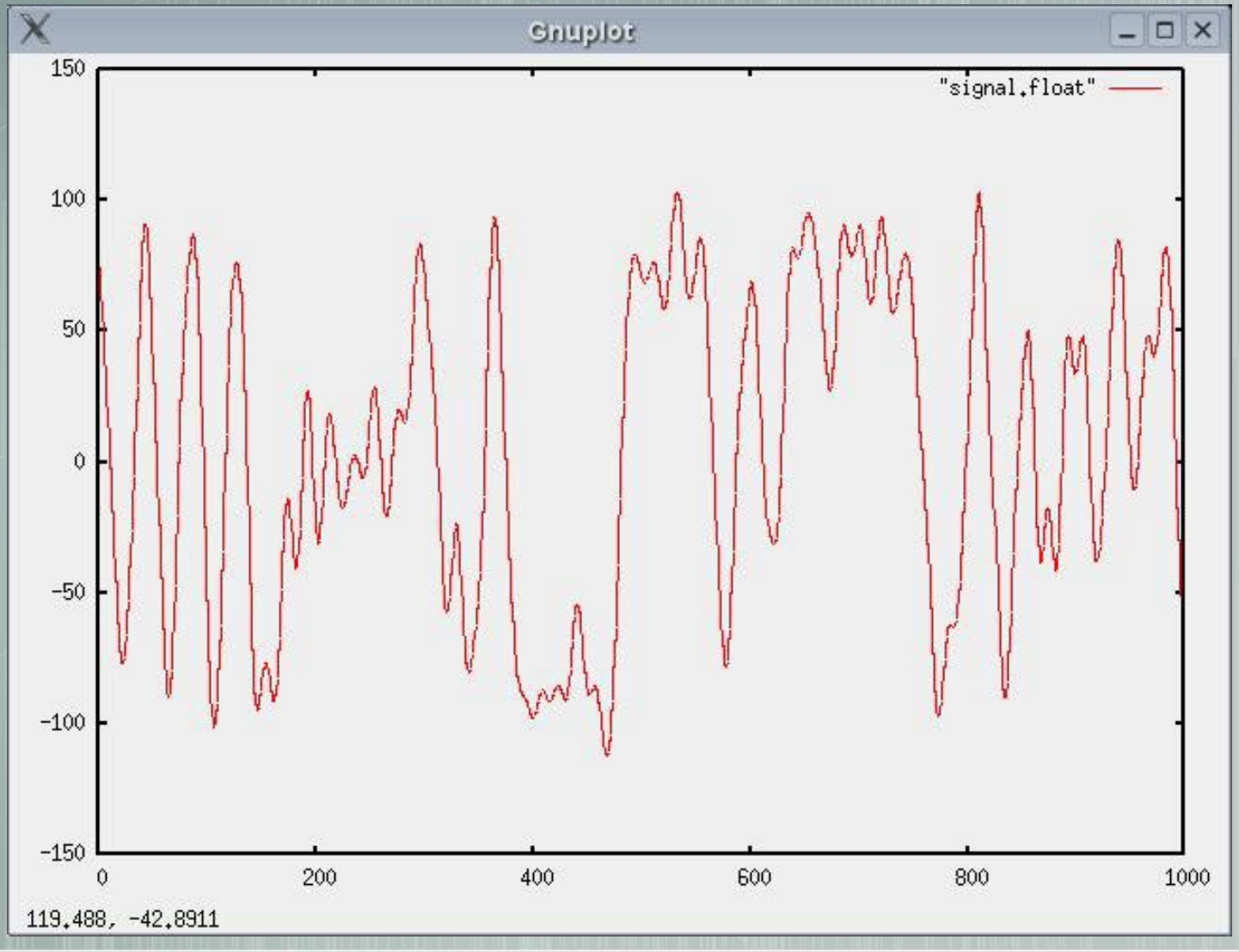
GSM

- Capture signal (Samples 2M – 3M @4Msps.)



GSM

- Samples 2M – 2.001M



GSM

Now, write some software!

- BCCH (SCCH)
  - FIRE parity
  - convolutional encoder / Viterbi decoder
  - block interleaving
  - map on burst
  - content parsing
- Next channel

Conclusion

- Penetration testing can be useful – cost/benefit. Costs for mistakes can be very high.
- Software radio is cool.
- More info? Ideas? Send me email:

jl@thre.at