An AWKward Modem

5 Lines to Scream in Silence

Remember *WarGames*, where a teenager dials into a military supercomputer via an acoustic coupler — a device that converts data into sound over a telephone handset? Can we revive this technique to exfiltrate data from a locked-down Unix system?

Figure 1: The acoustic coupler in *WarGames*

1 Acoustic Modems

The Bell 103 protocol was introduced by AT&T in 1962. Each byte is framed as: `0 BYTE_LSB 1`. For example `*` (ASCII 42) becomes `0 01010100 1`. The modem transmits each `1` (`MARK`) with a 1270 Hz tone and each `0` (`SPACE`) with a 1070 Hz tone. At 300 baud (300 symbols per second) and 10 bits per byte, Bell 103 transfers 30 bytes/s. Not fast, but enough to exfiltrate an SSH private key in under a minute.

2 Create WAV Files With AWK

With no Internet access, no compiler, and no install rights, an attacker can still turn to `awk`, a POSIX-standard utility, available on virtually every Unix system since the 1970s. `awk` processes text files line by line, with optional BEGIN and END blocks. Perfect for encoding Bell 103 tones into a WAV file![1]

`bell103.awk`[2] comes fully commented, but the compact 5-line version (433 bytes) below is all you need — ready to type or copy-paste[3].

The BEGIN block outputs the WAV header, followed by a MARK carrier for synchronization. The main block turns each character of input into 10 tones. The END block writes a final MARK carrier.

3 Decode the Signal

Implementing a Bell 103 decoder is relatively straightforward. For each bit, the Goertzel algorithm measures which of the two frequencies has more energy. We can also use an existing decoder such as `minimodem`[4].

```
minimodem --rx 300 -M 1270 -S 1070 -f out.wav
```

We can even decode live from the microphone. In quiet environments and with good hardware, the error rate is near zero at short distances[5].

```
# On macOS: brew install sox minimodem
rec -q -c 1 -r 48000 -t wav - \
  | minimodem --rx 300 -M 1270 -S 1070 -f -
```

```
# On Linux
arecord -q -c 1 -r 48000 -f S16_LE -t wav - \
  | minimodem --rx 300 -M 1270 -S 1070 -f -
```

4 Ultrasonic Exfiltration

While standard audio hardware operates up to 20 kHz, the upper limit of human hearing tends to drop with age, often to 15–16 kHz around age 45. By shifting into this near-ultrasonic range, data can flow inaudibly.

To enable this mode, set M to 17500 and S to 15000 in the awk script and in `minimodem` flags. Interestingly, the iOS Voice Memo app in lossless mode faithfully captures these frequencies. A nearby iPhone could record this silent transmission for later decoding.

```
minimodem --rx 300 -M 17500 -S 15000 -f out.wav
```

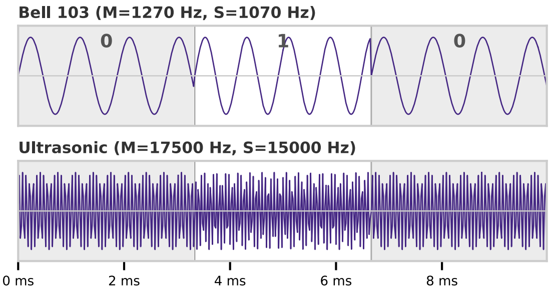

Figure 2: Frequency Shift Keying Waveforms

```
# b.awk - an AWKward Modem - Nicolas Seriot 2026 seriot.ch
# Usage: echo "Secret" | LC_ALL=C awk -f b.awk > out.wav
BEGIN{M=1270;S=1070;R=300;H=2^16;FS="";for(i=256;i--;)o[sprintf("%c",i)]=i;printf"RIFF";P(H*H-1);
printf"WAVEfmt ";P(16);P(65537);P(48e3);P(96e3);O(2);O(16);printf"data";P(H*H-1);B(1)}
function P(v){O(v);O(v/H)}function O(v){printf"%c%c",v%256,v/256%256}function B(b,j)
{for(j=48e3/R;j--;){p+=6.283*(b?M:S)/48e3;O((sin(p)*3e4+H)%H)}}function W(c,b){B(0);
for(b=8;b--;){B(c%2);c=int(c/2)}B(1)}{for(i=0;i++<NF;)W(o[$i]);W(10)}END{B(1)}
```

[1] `awk` works with text and gets confused by NUL characters. Encode binary files in Base64 before transmission.
[2] `bell103.awk` `https://gist.github.com/nst/73ba26cff092cecdac0a851c56ef0243`
[3] On Linux, `awk` may be a symlink to `gawk`. If so, prefix `LC_ALL=C` to prevent `gawk` from UTF-8 encoding binary output.
[4] Kamal Mostafa `http://www.whence.com/minimodem/`
[5] Depending on your hardware, you may need to decrease the baud rate from 300 to 100 or lower for reliable data transmission.