Paging Dr. Brackett: How Software Defined Radio Makes RF Espionage Trivial

Ashton Knight

December 12, 2018

Abstract

It is no secret that the POCSAG paging protocol was not designed with security in mind. In fact, having been first proposed in 1975, it predates modern ideas of security by decades. But with the introduction of inexpensive software defined radios, the cost and knowledge barriers that once made spying on these transmissions difficult have been all but demolished, and the information is available to anyone with twenty dollars and a free afternoon.

1 Introduction

Pagers are one of the longest running commonly used radio frequency devices, and have been an indispensable tool for emergency response teams like police forces, hospitals, and firefighters for over 70 years. The first telephone pager was implemented in 1949 for the Jewish Hospital in New York City, but similar rad systems had been in use by police forces as early as 1929. These early systems only had the capability to cause the desired pager in the system to ring at a few set tones, that the holder of the pager could interpret.

Most pager systems work by addressing the desired pager with its assigned "tone," a frequency or set of frequencies that uniquely identify a pager. These would be transmitted to all pagers in the network, and when a pager received its identifier it would wake up. It later iterations of paging protocols, the pagers had the ability to read numerical characters on an LCD display after being addressed, instead of only beeping. This way, the recipient could receive a "call back" phone number to call for further instruction. This

progressed to support alphabet characters as well, giving the capability to transmit brief messages over the protocol.

The most common paging protocol, POCSAG, stands for Post Office Code Standardization Advisory Group, and devised by a group of the same name led by the British Post Office in 1975. Below you can see a transmission using this protocol. Labeled is the address tone, followed by digital data transmitted to the pager. There is no encryption on the data, and it can be decoded using POCSAG protocol regardless of if the device decoding it was the intended recipient or not.

This paper is also concerned with another technology, software defined radio [1], or SDR. A software defined radio operates as a traditional radio would, except the functions normally done with analog components would be done in software. Making a good SDR receiver is difficult, as the receiver must sample the entire RF spectrum in its range before the software filters it. If this can be done cheaply, then you would have a very versatile broadband radio receiver on your hands, and the ability to process the signals very easily as they are already digital.

2 Method

The method to decode pager messages over the air using a software defined radio is relatively simple. You will need a few things:

SDR Reciever These can be acquired rather cheaply on-line, the one used here is called RTL-SDR[1] and cost about $20. A more expensive receiver may have a larger range and the ability to transmit as well as receive, but the

Linux Operating System Linux distributions are free, and can be installed on a VM if one does not have a device. Similar SDR software is available on Windows as well, but here we use a laptop running Ubuntu natively.

GQRX This SDR software[2] is available free for any Linux system, and allows you to tune your receiver, choose a demodulating method, and stream the audio signal.

[1] www.rtl-sdr.com
[2] gqrx.dk

Multimon-NG This is a free pager decoder[3] available for Linux. It will decode an already demodulated pager signal and output the messages to the terminal.

The receiver and GQRX were set up according to the instruction on the rtl-sdr website, and the search for POCSAG signals could begin. Finding them is as simple as looking up common pager channel frequencies, tuning to those ranges, and looking for a pattern that resembles the spectrogram in figure 1. The signal is demodulated as a narrow band FM signal, and GQRX can pipe the audio to a localhost port over UDP. Then, using a simple script attached to the appendix here, a netcat listener can be set up to pipe the audio to Multimon-NG, which could then decode the signal and read all messages in that channel in plaintext.

An example of the whole set up is shown in figure 2. In the GQRX window, one can see the spectrogram of the RF spectrum near our tuned frequency, and the pattern from figure 1 is visible here. The window in the upper right is the waveform received by multimon-ng, and the TTY in the lower right containers the output from the decoder. Figure 3 shows a close up of the most concerning find in the decoded messages: a first name, last name, and date of birth. The unholy trinity of personally identifiable information. Pieces of this are redacted as to not violate any laws by sharing this paper.

3 To The Community

So, why should someone be concerned about this? If these channels are used responsibly, there should be no sensitive information communicated across them. The typical use case of these channel are usually to provide a phone number to call back, or direct a doctor to head to a certain room. But, alas, this is the real world. In just a few minutes of tuning to a local hospital, this experiment read, among other things, patients first and last names, sometimes with date of birth, and specifics of their medical conditions. Not only is this information personally identifying, but the medical information could put these transmissions in violation of HIPAA. This is a very big deal, and not something to be taken lightly. Perhaps action has not been taken on this yet because of the cost and expertise required to snoop on the RF band in days past, but today the barrier is access to $20 dollars and the ability to

[3] tools.kali.org/wireless-attacks/multimon-ng

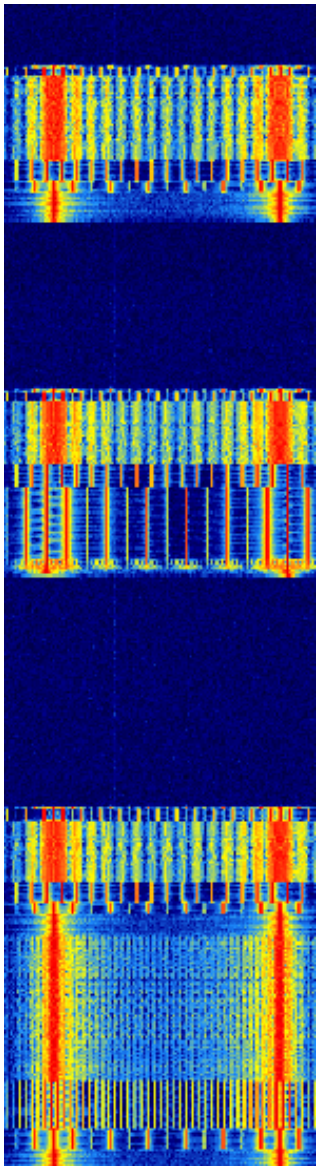

Figure 1: POCSAG spectrogram

read instructions, and now even the argument of security through obscurity is rather weak.

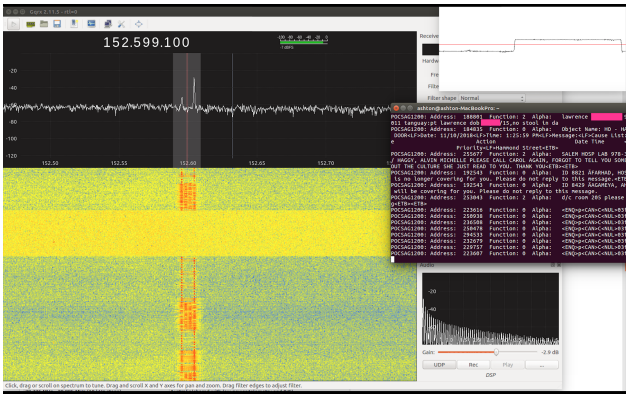

Figure 2: Screenshot of pager decoding setup

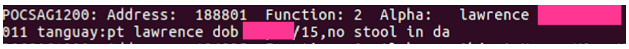

Figure 3: Name and date of birth transmitted in the clear

4 Action Items

It would probably be unwise to ban pager systems all together [2][3], or even ditch the protocol. Pagers have been in use for a long time, and offer unparalleled robustness and throughput for the conveying of critical information over wide areas. However, the transmission of such sensitive data in the clear is no longer acceptable. Luckily, if the medical community has the motivation and resources to change this practice or update their infrastructure, the technical solutions should not be too difficult.

Pager systems have the unique advantage of using identical physical devices distributed from a central location, and only having one direction of communication. Hypothetically, private keys for a basic encryption scheme over an alphanumeric carrier could be paired with pager like devices before distribution, allowing secure encryption without the need to define a key exchange protocol. A less costly but also more fallible solution would be to instruct hospital workers to keep sensitive information off of the pager channels. This is easier said than done, but the messages containing sensitive information were rare enough that it is not unreasonable to think that they could be communicated over a more secure channel.

5 Conclusion

So what is the lesson learned here? Its not that pagers are insecure. It is well known that pagers transmit date without encryption [4], and a setup similar to this experiment were the subject of an art exhibit [5] in the past year, First, the availability and afford-ability of the SDR setup should turn heads as to what other information could be left in the clear in the RF spectrum. The digitization of this equipment mean that snooping the the spectrum is no longer just for hardcore amateur radio enthusiasts, and that the data is easier to save and process than ever before.

Second, and most important, is to ask the question: why would something like this still be left in the clear? Likely, the answer lies with the biggest challenge in security, which is humans. No matter how good the security community is at finding vulnerabilities, or thinking of clever ways to secure systems, nothing will ever change until the people in charge can be convinced of their importance. Hospitals are staffed by busy people, and often have tight budgets for infrastructure. If nobody is convinced that it is critical that a system like this is updated to keep their patients information secure, it will never happen. And this is not the fault of the employees of the hospital, but rather the responsibility of those who know how high the stakes are to communicate the severity to those who are at risk.

A Appendix

Listing 1: script used to pipe signal to multimon-ng

```
#!/bin/bash

nc -l -u 7355 |
sox -t raw -esigned-integer -b16 -r 48000 \
        - -esigned-integer -b16 -r 22050 -t raw - |
multimon-ng -t raw -a SCOPE -a POCSAG512 \
        -a POCSAG1200 -a POCSAG2400 -f alpha -
```

References

[1] A. A. Abidi, "The Path to the Software-Defined Radio Receiver," *IEEE Journal of Solid-State Circuits*, vol. 42, pp. 954–966, May 2007.

[2] L. Powers, "Why the pager isn't dead yet," *CBC*, Jan. 2015.

[3] P. Association, "Old technology: NHS uses 10% of world's pagers at annual cost of 6.6m," *The Guardian*, Sept. 2017.

[4] S. Hilt and P. Lin, "Leaking beeps: A closer look at it systems that leak pages," *Trend Micro*, Sept. 2016.

[5] "Art Installation Eavesdrops on Hospital Pagers with a HackRF," Dec. 2017. Available at https://www.rtl-sdr.com/art-installation-eavesdrops-on-hospital-pagers-with-a-hackrf.