GSMem Startup Project

Andrew Kwong
University of Michigan

Kevin Fu
University of Michigan

Abstract

Security conscious companies and government agencies may choose to carry out extremely security critical computations on air gapped machines, which are physically disconnected from the internet. Despite taking such measures, recent history has shown that such machines can still be compromised. To address the problem of exfiltrating data in such circumstances, a recent study [2] demonstrated how an infected, air-gapped machine can modulate arbitrary binary at GSM frequencies by utilizing the electro-magnetic radiation (EMR) given off by memory-related CPU instructions. A nearby cell phone with a baseband rootkit can then demodulate the signal and transmit or store the exfiltrated data for the attacker.

The purpose of this paper is to verify the claims made in [2], regarding the feasability of using signals transmitted at GSM frequencies as a covert channel. In our studies, we found that a possibly shielded laptop could indeed transmit arbitrary binary, that could then be received by a cell phone with a modified baseband firmware. While we were unable to produce signals of the same strength as claimed by [2], we nonetheless succeeded in developing a proof of concept that confirms the feasability of the proposed covert channel.

1 Introduction

An organization may choose to use an air-gapped machine to store or operate on sensitive data, in the hopes that complete physical separation from public networks will eliminate all vectors of infection. Stuxnet [1], however is just one recent example of this assumption being violated. The virus, likely a product of the joint efforts of Israeli and United States intelligence agencies, propogated via USB to air-gapped machines that controlled Iranian nuclear refineries.

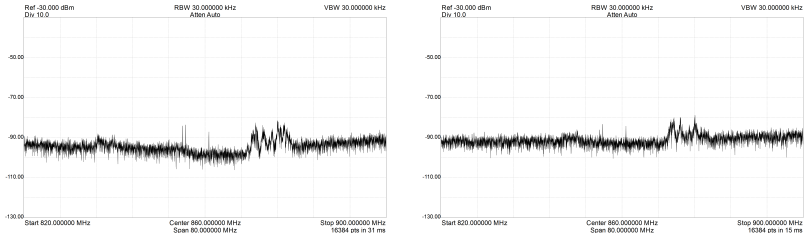
Once such a machine has been compromised, there is still the issue of data exfiltration; cryptographic keys and other sensitive secrets may be present. To address this task, researchers [2] devised a method by which the infected, air-gapped machine can transmit arbitrary binary via the EMR produced by the memory bus. Since most memory buses operate within the GSM frequency range, a baseband rootkit in a nearby phone can demodulate the signal and then transmit the secrets to the attacker. It should be noted that the proximity of a nearby phone is indeed a realistic scenario; in Lockheed Martin's instructions [3] to facility visitors, it is stated that, "*Because ATL is a secure facility, the following items are not allowed to our floor of the building: cameras (film, video, digital), imaging equpiment, tape recorders, sound recording devices. Cell phones are allowed, but camera/recording features may not be used*".

Unfortunately for researchers investigating the internal workings of baseband chips, the baseband industry is extremely secretive and refuses to release any source code. Because of this, the authors in [2] opted to use an older, feature phone that the opensource community has developed baseband firmware for; as such, it is highly likely that modern phones and dedicated radio receivers can implement the proposed covert channel more effectively.

In this paper, our primary contribution is the verification that a nearby GSM phone can pick up the proposed covert channel from an air-gapped, transmitting machine. The rest of this paper is organized as follows: Sections 2 and 3 will describe the implementations of the transmitter and receiver, respectively. Section 4 will describe the experimental setup. Section 5 will evaluate our results. Section 6 concludes.

2 Transmitter

To implement the transmitter, we used the computer's memory bus as an unintentional antenna. When data travels across the wires, EMR is emitted at approxi-



(a) Frequency domain from casual emissions. (b) Frequency domain while transmitting high.

Figure 1: Power measurements taken just above the laptop

mately the frequency of the bus's clock (800MHz for our setup). We made use of Intel's Streaming SIMD Extensions 2(SSE2) instruction set to maximize this effect. In particular, we used the _mm_stream_si128 intrinsic, which corresponds to the MOVNTDQ (move non-temporal double quadword) SSE2 instruction; this instruction moves 128 bits from memory to the CPU's SSE registers, while ignoring caches, allowing us to fully utilize the memory IO bus.

If we repeatedly execute this instruction in a loop, the result is an increase in power around the 800MHz band. Figure 1 compares the EMR from regular activity to the EMR measured while the machine is transmitting. Upon visual inspection of the radiation's fallout among the GSM-850 band, we decided that 860MHz is a good carrier frequency for modulating our signal at, given the nearly 10 dBm difference.

Building upon this phenomena, we modulate arbitrary binary using the binary amplitude shift keying (B-ASK) scheme. To transmit a 1, we fully utilize the memory bus for t seconds; to transmit a 0, we do nothing for t seconds.

3 Receiver

Due to the baseband industry's largely successful efforts at security through obscurity, we resorted to using the Motorola C118 feature phone for our receiver; this is because the OsmocomBB(Open-Source MObile COMmunication BaseBand) project has developed the only open source GSM baseband implementation, for certain compatible phones. As such, osmocombb was the only way by which we could modify the source code for a baseband chip.

The baseband firmware was modified so that the RTOS's main event loop called an additional function with each iteration. In this additional function, power measurements at 860MHz are taken and stored in a ring buffer. Once averages for transmission at B-ASK 1 and B-ASK 0 levels are found, the baseband rootkit can then demodulate the signal by sampling the DSP.

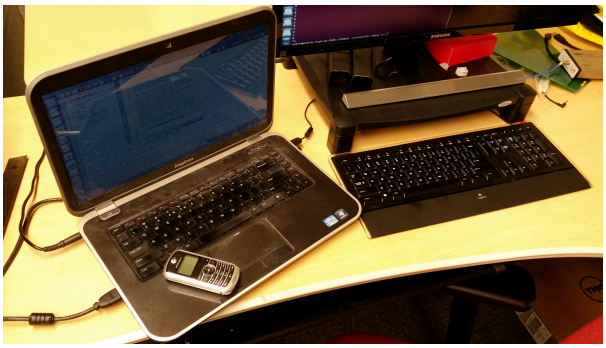


Figure 2: Experimental setup.

If samples are taken at a higher rate, the receiver can be made more resilient to noise; due to limitations in the processing power of such an old feature phone, however, we found that using t=2 seconds and sampling 8 times per period resulted in a reliable channel with an acceptable throughput of 0.5 bits/second. This allows us to exfiltrate an AES-256 key in just under 9 minutes.

4 Experimental Setup

Our receiver can receive files transmitted through GSMem [2] when the phone is sitting right on top of the transmitting laptop, as shown in Figure 2.

The laptop used is an Inspiron 15R 5520, which has DDR3 memory with a bus clock of 800MHz. The phone is the Motorola C118 phone, which has a Calypso chipset.

All experiments were conducted in a graduate student office, exposed to multiple sources of noise and background radiation. There were numerous desktop machines in use within a 5m radius.

5 Evaluation

In this section we will evaluate whether or not the covert channel proposed in [2] works as claimed. In particular,

we will examine the covert channel's range and throughput.

Guri's [2] study claims that they achieved power level differences of 1 dBm at ranges of up to 110cm. As the graph to the right demonstrates, we were unable to achieve the same range as Guri et al. At ranges of just over 4 cm, the lines already converge, and there is no discernible signal.

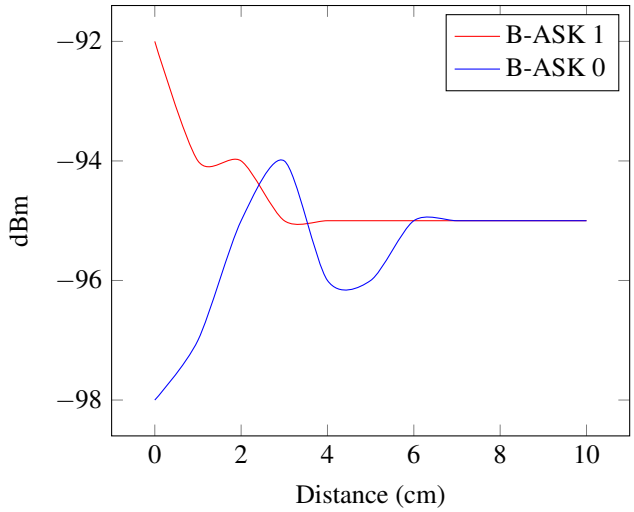
We attribute this discrepancy to one major factor: shielding. In contrast to the Guri et al., who transmitted from various workstations, we only transmitted from the Inspiron 15R 5520 laptop. While some of their cases had sides made primarily of plastic, doing little to attenuate EMR, our laptop was likely shielded by a metal case, and we were simply observing EMR leakage from the keyboard. This hypothesis is supported by our observation that the strength of the signal varied dramatically as the phone was moved to different locations on top of the laptop.

Regarding throughput, at very close proximities we were able to produce comparable results. While Guri et al. were able to transmit up to 2 bits/second, we were able to achieve a throughput of 0.5 bits/second. There are a few factors that are likely culprits.

For one, we used the Motorola C118, as opposed to the C123, to implement our receiver. The C118 is an older model, with reduced computational power, and possibly a less sensitive/accurate DSP. Another is that the weakened signal compounds that fact that the channel becomes more susceptible to noise at higher bit rates; as the period shortens, shorter durations of interference can flip bits. Finally, it also quite possible that Guri et al. simply implemented more effective noise filtering in the phone's baseband.

6 Conclusion

Despite being unable to perfectly replicate GSMem's [2] results, we did manage to accomplish what we set out to do. That is, we modified a phone's firmware, and successfully transfered files via the GSMem covert channel to the phone. Though the signal to noise ratio was only sufficient when the phone was lying directly on top of the laptop, we deem this a satisfactory proof of concept that verifies the feasability of using GSMem to exfiltrate data from air-gapped machines.



References

[1] Falliere, N., O Murchu, L., and Chien, E. W32.stuxnet dossier. Tech. rep., Symantec Corporation, February 2011.

[2] Guri, M., Kachlon, A., Hasson, O., Kedma, G., Mirsky, Y., and Elovici, Y. Gsmem; data exfiltration from air-gapped computers over gsm frequencies. In *Proceedings of Usenix Security Symposium 2015* (Aug. 2015).

[3] L, M. Important information. `http://www.lockheedmartin.com/us/atl/maps/cherryhill/information.html`.