Passive and Active Attacks on Audience Response Systems using Software Defined Radios

Khai T. Phan Ryan Ewing David Starobinski and Liangxiao Xin

Boston University, Boston, Massachusetts 02115, USA,
{kphan95, rjewing, staro, xlx}@bu.edu

Abstract. Audience response systems, also known as *clickers*, are used at many academic institutions to offer active learning environments. Since these systems are used to administer graded assignments, and sometimes even exams, it is crucial to assess their security. Our work seeks to exploit and document potential vulnerabilities of clickers. For this purpose, we use software defined radios to perform jamming, sniffing and spoofing attacks on an audience response system in production, which provide different possible methods of cheating. The results of our study demonstrate that clickers are easily exploitable. We build a prototype and show that it is practically possible to covertly steal or forge answers of a peer or even an entire classroom, with high levels of confidence. Additionally, we find that the receivers software of the system lacks protection against unexpected answers, which allows our spoofer to submit any ASCII character and opens the receiver up to possible fuzzing attacks. As a result of this study, we discourage using clickers for high-stake assessments, unless they provide proper security protection.

1 Introduction

Many institutions employ Turning Technologies' Response Cards [7], also known as *clickers*, to create active learning environments and encourage students' participation in their classes. Clickers are wireless devices that let instructors poll students for purposes such as taking attendance, and administering quizzes and/or surveys. Research has shown that such a learning tool can greatly improve students' learning abilities and engagement with material if the clickers are used effectively [4–6].

While many universities limit their use of these clickers to attendance monitoring and in-class polls, some educational institutions go so far as to administer clicker-based exams. University of Maryland of Baltimore County shows evidence of having administered these types of exams in the past. A post on the university's Division of Information Technology page includes a quotation of a student expressing favor for these exams, commenting "I liked taking the exam on the clickers because we had our own exam booklet in front of us and could go at our own pace. I also liked getting my grade back right away." [1]

The popularity of clickers raises the question of whether these devices are actually secure. In particular, since clickers transmit over radio frequencies, is it possible for a student or another party to block, eavesdrop, or change answers submitted by other students?

In this paper, we answer this question in the affirmative. We build a prototype of a fake receiver (sniffer) and a fake clicker (spoofer) using the HackRF One software defined radio platform [2]. Using information provided by the sniffer or the functionality of a spoofer, a student can cheat in various ways, e.g., by finding out the most commonly submitted answer, looking at the answer submitted by a particular student (assuming the clickerID of that student is known), or by altering the answer submitted by other students. Furthermore, we uncover new information about the TurningPoint receiver and polling software that could lead to additional vulnerabilities in the form of fuzzing. Specifically, we find these technologies do not fully sanitize user input, allowing our spoofer to submit unexpected answers to polls.

2 The Tools

2.1 The HackRF One

The specific software defined radio used in this project to assess the security of Turning Technologies' Response Cards is Scott Gadgets' HackRF One. The HackRF One [2] is a hardware device able to capture radio signals via an antenna and stream the signal data captured through USB into another device, oftentimes a computer operating on a Linux-based operating system. This stream of data can then be modified and analyzed with software.

2.2 GNU Radio

With the HackRF One offering the hardware support for this project, the software GNU Radio [3] is used to perform signal processing and analysis on the digital input received via USB port. GNU Radio has become an increasingly popular tool for research, due to its customizability and simple GUI interface [8]. This user-friendly GUI interface is known as GNU Radio Companion, often abbreviated GRC. GNU Radio offers the software equivalent of nearly every hardware tool used in signal processing, making it an extremely powerful tool for this project.

3 Reverse Engineering

The procedure of receiving a data packet requires filtering the signal, demodulating it with the correct modulation scheme, synchronizing clocks with the signal's data rate, transforming the demodulated signal into a binary data stream (data consisting of 0 and 1 values), then interpreting the binary data stream in order to discover packets sent by the clicker. In order to implement attacks such

Table 1. TurningPoint Clicker Specifications

Operating Freq.	Bandwidth	Modulation Scheme	Baud Rate
2.401GHz - 2.482GHz	1 MHz	GFSK	1 Mbps

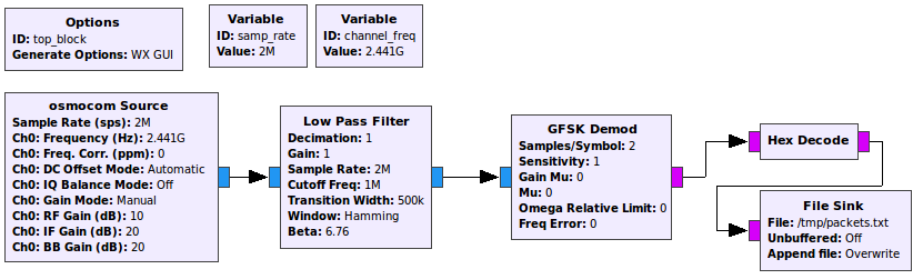

Fig. 1. GNU Radio implementation of the receiver.

as sniffing and spoofing, it is important to determine how clickers operate. To find the necessary information, we take advantage of specifications of the Nordic nRF24LE1 chip, data from the FCC website, and analysis of the clicker signal captured through the HackRF One. The information we found is summarized in Table. 1.

The final information required is the packet structure for the sake of sniffing and spoofing packets. The packet structure contains 8-bit preamble, 24-bit target address, 24-bit source address, 8-bit payload, and 16-bit CRC, where the preamble and target address are permanently 0x55 and 0x123456 and the CRC algorithm is CRC-CCITT (0xFFFF).

4 Sniffer Implementation

4.1 Flowgraph Blocks.

The GNU Radio flowgraph (see Fig. 1) consists of the following blocks:

1. The `Osmocom Source` generates a stream of complex numbers based on the signal that the HackRF One receives via its antenna.
2. This stream of numbers is passed through a `Low Pass Filter` in order to filter out all signals aside from the desired 1 MHz bandwidth clicker transmission channel.
3. That filtered stream of data is then passed through a `GFSK Demod` block which demodulates a GFSK modulated signal into bits.
4. Lastly, this stream of deciphered bits is pushed into the `File Sink` which saves the binary stream into a file.

4.2 GRC Implementation.

The only remaining step is to find a way to parse the binary stream in real time. To that end, we create a new block using GRC itself to decipher the packets. GNU Radio provides the option of writing custom blocks using C++ or Python, based on so-called Out-of-Tree (OOT) modules. Such modules are useful when one needs to implement a new function that GRC does not provide in its existing library. Toward this end, we create a simple Man-in-the-Middle block which directly parses the output from the `GFSK Demod` and logs the found packets to GRC's built-in console. We call the block `Hex Decode` (see Fig. 1), as it decodes the binary stream into hex.

5 Results

In order to assess the security of using clickers for high-stake graded assignments, we demonstrate jamming attacks, sniffing attacks, and spoofing attacks using the HackRF One device and gauge the efficiency of these attacks.

5.1 Sniffing

The goal of sniffing is to stealthily and passively acquire knowledge of others' answers and packet submissions. According to benchmarking results, sniffing should perform extraordinarily well within a lecture hall or classroom setting. An accuracy near or above 90% is achieved at almost all distances within 25 feet, with distance within 10 feet having near perfect results. Additionally, the sniffer receives on average twice as many packets as the receiver does, which means it is less prone to errors and could receive an answer earlier then the receiver. We note that in most scenarios, the user would be sitting near other clickers, generally within a vicinity of 25 feet radius. Thus, the clickers are extremely vulnerable to a sniffing attack, as such an attack is expected to receive nearly all answers that are submitted within the classroom.

5.2 Spoofing

Throughout our tests, we discovered several possible attacks using spoofed packets.

1. **Forging answers.** One attack involves changing the answers of other students. Once a clicker ID is known, the attacker can spoof a packet with the same ID with a different answer. The receiver, believing the packet is sent from the real clicker simply changes the answer stored for that ID, without notifying the student whose answer was altered. Since the HackRF One can quickly switch between transmitting and receiving, it is possible to collect IDs from an entire classroom of students and alter each answer in seconds.

2. **Tampering course statistics.** A second vulnerability lies in sending fake answers using fake IDs. Because all clicker IDs are a 6 digit hex number, it is possible to randomize an ID and an answer to provide false data. The TurningPoint software provides in-depth statistics to the teacher or professor for each question and poll. With skewed data, teachers and professors could apply inaccurate curves to quizzes and exams or focus on teaching material which most students already understand.
3. **Fuzzing.** Furthermore, while experimenting with the HackRF One spoofer, we found that the TurningPoint receiver has the ability to receive any two digit ASCII code in hex. While the TurningPoint clickers can only submit single digit, numerical answers (i.e., 0-9), the spoofer has the ability to send other two-digit ASCII hex code, including letters, mathematical symbols, punctuation, and control characters, such as the "Null" character. We discovered that the TurningPoint receiver does not outrightly reject or ignore such malformed inputs, which implies that the polling software could be open to brand new fuzzing attacks.

Acknowledgments

The authors thank Prof. Ari Trachtenberg for his suggestion to investigate fuzzing attacks. This work was supported in part by NSF under grants CNS-1409053, CNS-1563753 and CNS-1717858. The views expressed in this paper are those of the authors only, and do not necessarily reflect the views of NSF.

References

1. Students more accepting of using clickers for exams (Apr 2014), http://my.umbc.edu/groups/doit/posts/44012
2. HackRF One (2016), https://greatscottgadgets.com/hackrf/
3. The GNU Radio Foundation, Inc: GNU Radio (2017), http://gnuradio.org/
4. Han, J.H., Finkelstein, A.: Understanding the effects of professors' pedagogical development with clicker assessment and feedback technologies and the impact on students' engagement and learning in higher education. Computers and Education 65, 64–76 (2013), http://www.sciencedirect.com/science/article/pii/S0360131513000237
5. Kastner, M.: The use of an audience response system to monitor students' knowledge level in real-time, its impact on grades, and students' experiences. In: 2016 49th Hawaii International Conference on System Sciences (HICSS). pp. 104–113 (Jan 2016)
6. Kulatunga, U., Rameezdeen, R.: Use of clickers to improve student engagement in learning: Observations from the built environment discipline. International Journal of Construction Education and Research 10:1, pages 3-18. (2014)
7. Turning technologies: ResponseCard RF (2017), https://www.turningtechnologies.com/response-solutions/responsecard-rf
8. Valerio, D.: Open source software-defined radio: A survey on gnuradio and its applications. Tech. Rep. FTW-TR-2008-002 (August 2008), http://www.astro.square7.ch/Datenblaetter/SDRreport.pdf