HID RW300/RW400 Hardware Implementation Vulnerabilities and Mitigation Strategies

Taylor Nelson

Cal Poly San Luis Obispo

06/03/13
# Table of Contents

Abstract ........................................................................................................................................... 3  
Introduction, Overview and Motivations ....................................................................................... 4  
Implementation Details .................................................................................................................. 7  
Results ........................................................................................................................................... 10  
Results Continued: Mitigations ..................................................................................................... 12  
Miscellaneous Door Security Issues .............................................................................................. 13  
Conclusion and Future Work ........................................................................................................ 14  
References .................................................................................................................................... 16  

# List of Figures

Figure 1: IClass Authentication System........................................................................................... 5  
Figure 2: A diagram of Electronic Codebook mode encryption...................................................... 6  
Figure 3: Read memory bug. ........................................................................................................... 8  
Figure 4: Wiha Security bit set and reader with back cover off. .................................................... 9  
Figure 5: Memory dump in progress. ........................................................................................... 11
Abstract

The HID RW300/RW400 readers are in use and deployed in buildings around the world. Designed for industrial/commercial facilities security requirements, you can find them outside biotech companies (AMGEN), educational institutions (Cal Poly San Luis Obispo), and doctors offices. In a 2010 paper entitled “Heart of Darkness - exploring the uncharted backwaters of HID iCLASSTM security” by Milosch Meriac, numerous vulnerabilities were outlined. Despite this research HID continues to sell these readers to consumers.

These HID devices are no longer adequate to properly protect the safety of individuals and assets. I have verified that these attacks are feasible by a moderately skilled adversary (myself) with limited resources, less than 60$ and a laptop. These attacks would enable an individual to quickly subvert the security afforded by these devices and immediately gain access to protected areas. In this paper, I will present my findings as well as mitigations that institutions currently utilizing these devices may adopt to better improve their facilities security.
Introduction, Overview and Motivations

RFID stands for radio frequency identification. This protocol is used by the HID readers to interrogate the authentication medium which are plastic cards embedded with RFID chips. There are two major kinds of RFID reader/tag pairings, passive and active. Active RFID chips are typically powered by a battery and may transmit several hundred meters via UHF. This is the kind of RFID system used in the Toyota Prius. Passive RFID chips, on the other hand, are powered by electrical inductance from the reader, which drives them to produce a response. This response is often a tag id code or “EPC” code. If the reader is providing a security key or authentication data, it may respond to a cryptographic challenge from the RFID reader with a signature or permutation of that challenge.

The HID RW300 and RW400 are affordable RFID readers in the 13.56 MHz range, which is for short range passive RFID. These readers are designed to be integrated into a perimeter security system, which allows an administrator using a computer workstation to assign new RFID keys to individuals, track and audit access and manage identities. One example deployment of this system is found in the PolyGAIT RFID lab in building 40 on the Cal Poly San Luis Obispo campus. Figure 1 shows a diagram of one of the rooms.
The HID RW300 and RW400 readers have two major security modes, Standard and High Security. The Standard mode is the most common and is used by the PolyGAIT RFID lab. The data on iClass cards is 3DES encrypted and the 3DES encryption keys are stored on the RW300/RW400 devices. These iClass cards contain a facility ID and tag ID, which are mapped on the authentication server to an individual who is assigned that card.

RFID authentication is typically subject to three major types of compromise: card cloning, reader bypass, and key compromise. In card cloning, one user’s card may be duplicated and used to authenticate as that user. The cloning data is often obtained either
through active interrogation of the card or through passive sniffing of the authentication between the facility reader and the user’s card. In reader bypass the actual reader is circumvented this may be possible in numerous ways such as manually powering the electric door strike with a battery. Lastly, in key compromise, the manufacturer’s key is exposed, which most often allows users to create new cards/keys. This situation should not be possible. New card creation is still unlikely as the EPC or tag id is read-only at the hardware level.

In “Heart of Darkness - exploring the uncharted backwaters of HID iCLASSSTM Security”, Meriac describes numerous cloning vulnerabilities. The first issue he describes is that the Omnikey reader enables one to read and WRITE the iClass cards, which should not be possible. The second is that the iClass cards utilize the ECB cipher mode, which means different blocks can be swapped on the card and cards can be cloned by directly. The third is that the ECB invocation is not tied to the card ID which means blocks can be swapped between cards.

![Diagram of Electronic Codebook (ECB) mode encryption](image)

**Figure 2: A diagram of Electronic Codebook mode encryption.**

In an article posted to the opensecurityresearch blog[4], Brad Antoniewicz describes an improved method to dump the 3DES keys on an iClass reader using a FTDI TTL-232R-5V-WE.
This allows you to quickly build a serial interface to dump the memory from a vulnerable iClass reader onto a USB interface. This Future Technology Devices International serial to USB interface is controlled by a laptop. This methodology makes exploiting these readers practical and inexpensive for less than $60 you can build an interface to the RW300/400.

When I was first considering what to do a senior project on I knew I wanted to do something to do with physical security. Being practical I considered what was close to Cal Poly, I knew that there was RFID authentication on campus and I suspected it was vulnerable. This suspicion was due in part to the fact that Cody Brocious had recently published[1] several vulnerabilities in the Onity INTEGRA door locks which can be found in Poly Canyon Village. To investigate, I visually inspected the authentication devices around campus such as in Building 4 and found them to be iClass readers of various designation. After researching the iClass system, I discovered that these readers and the system possessed several glaring vulnerabilities.

The final stroke in deciding on this project came one day when I was attending a yoga class. My friend Sarah Clemn said to me something along the lines of “Taylor you know with all those school shootings happening you think you could think of some kind of improved door locking system to protect people”. I knew then that my project would have to be dedicated to bringing awareness of the vulnerability of these readers and constructing solutions to improve buildings security on the Cal Poly Campus.

Implementation Details

To implement these attacks, I chose to try to replicate the memory dump of an HID RW300. I chose the RW300 because it is more affordable to purchase online than the RW400.
I chose to do a memory dump because once you have the keys from the dump cloning, other attacks are already well documented and trivial. I additionally chose to attack the serial interface because I consider the presence of an unsecured serial interface a vulnerability in and of itself.

The serial interface has six pins. These pins are mapped left to right as VSS, VDD, VPP/MCLR, PGD, PGC, PGM. In the paper by Meriac [5], he discloses a copy protection bug which enables reading memory from the RW300/RW400 devices. He also published this code on the openpcd website [6]. This copy protection bug can be activated by the sequence in the listed in Figure 3. Using this bug and code from [6] as well as libFTDI or libD2XX, it is trivial to read the device memory.

<table>
<thead>
<tr>
<th>ICSP 4-bit Command</th>
<th>ICSP Data/ PIC Instr.</th>
<th>18F452 PIC Assembly Code</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0x0200</td>
<td>MOVWL, 0</td>
<td>Set Upper byte of Index Addr - 0 Define Start Addr</td>
</tr>
<tr>
<td>0000</td>
<td>0x626A</td>
<td>MOVWF FSRWUH</td>
<td>Set Lower byte of Index Addr - 0 Define Start Addr</td>
</tr>
<tr>
<td>0000</td>
<td>0x0B00</td>
<td>MOVWL, 0</td>
<td>Read File Register &amp; Incr Index Loop here for all</td>
</tr>
<tr>
<td>0000</td>
<td>0x5002</td>
<td>MOVWF, MOCRINOD</td>
<td>Move Reg data to ICSP Register</td>
</tr>
<tr>
<td>0000</td>
<td>0x625F</td>
<td>MOVWF, TABLEAT</td>
<td>Send data byte read to ICSP I/F</td>
</tr>
<tr>
<td>0010</td>
<td>Reg Data</td>
<td>N/A</td>
<td>1536 Reg's</td>
</tr>
</tbody>
</table>

Figure 3: Read memory bug.

To get access to a device, I purchased a used RW300 from Ebay; I also ordered a USB TTL-232R-5V-WE from Sparkfun. The USB-TTL-232R-5V-WE is a serial to USB interface, which supports bit-bang mode that allows you to individually control the pins. Individual control of the pins allows you to emulate a pic programmer. Emulating a pic-programmer allows you to easily read device memory to a computer without needing an additional pic-computer interface.
After you connect the FTDI cable you also need to power the RW300 device, so I purchased a power supply at Radioshack. These parts were all relatively easy to find. However it took me some time to find the 6-pin socket connector to interface with a serial device. These connectors follow the pin socket’s dimensions 1x6. The pin sockets allow you to easily feed wires in from the FTDI to the serial interface in the proper order to facilitate pic emulation. The FTDI-serial mapping is given in [4] as Black ->1(VSS), Red ->2(VDD), N/A -> 3(VPP), Green -> 4(PGD), Orange -> 5(PGC), Brown -> 6(PGM).

To construct the interface, I removed the back panel of the RFID reader, which involves removing several screws. These security screws provide no protection against a simple WIHA security screwdriver bit set, which retails for about $20.

![Figure 4: Wiha Security bit set and reader with back cover off.](image)

After removing the back cover, I twisted wires together as listed above in the serial->FTDI mapping and placed them into the proper slots in the 1x6 pinsocket.
After I had constructed the USB-serial adapter, I then connected it to the RFID reader RW300. I finished by connecting in the 9V power source and FTDI power source. The fully constructed interface can be seen in Figure 5.

Results

I dumped the memory off the device using a C implementation based off [4]. This was done by powering the FTDI controller and then powering the device and programming the PIC controller, with a laptop connected. The serial dump was successful, verifying that reading the memory is possible. The device I purchased turns out to have had its memory cleared. Because its memory was cleared, I could not read or verify the location of the master keys. I was disappointed that the memory on the RW300 I purchased was wiped but I had no access to other such devices to perform the testing I wanted. I thought about purchasing more but with any probability, if they had been removed from a site facility they would have their memory wiped. Figure 5 shows the memory dump is shown in progress.
Despite the fact that the memory on the RW300 device I purchased was zeroed I don’t consider this a failure of my project. My main interest in this project was to verify that these attacks were possible and to think of mitigations.

I verified that the dump was indeed successful by observing the reader LED strobing as described and beeping when powered on. I also verified that all the commands being sent successfully over the FTDI as done by the dumping program.
The ease in which this memory data was accessed made me realize that in order to protect against this attack I would need to consider the pathway to the memory dump. In considering this pathway, I thought of ways that it could be slowed. Slowing access to the serial pins would primarily mean defeating access to them and several ideas are listed in the following mitigations section.

Results Continued: Mitigations

The RW300/400 devices and variations thereof can be found throughout the Cal Poly campus. I have personally observed these readers in buildings 4, 40, 13, 14, 007, and 197. These devices are vulnerable to numerous attacks and present a serious risk to the resources protected by these devices. Considering the low cost and readily available technical information on these attacks, it would seem prudent to remove or mitigate this risk.

I consider the simplest mitigation a removal of access to the serial interface. It is clear from the above section that security screws are not effective. I think that this could be easily accomplished by soldering a small strip of metal over the interface pins. Later, these could be de-soldered. This fix requires several minutes per device and a person skilled in soldering. At $30/hour and a third of an hour required to patch a device, estimating the number of readers on campus as 100-200, I would expect this to be reasonably priced for a good fix to be approximately $1000-2000$. This would deter many people from attempting to compromise these systems as it would take significantly longer to compromise a device.

There are other ways to remove access to the serial interface. Mostly removing access to the serial device involves extended the length of time required to attack the device. This causes the adversary to have a greater risk of exposure. However barring the serial device
does not remove the risk of card cloning. Card cloning is actually a much more common attack vector than direct reader compromise and is generic across many kinds of cards.

Card cloning can be prevented by users of the system. The users may adopt methodologies which may include using an insulated wallet (common now in passport wallets), entering rooms alone (to prevent cloning when the card is exposed), and keeping their card safe from theft.

User prevention for RFID cloning is unlikely and impractical. Another option is to enable High Security Mode on the iClass RW RFID readers, which would require reprogramming the device and distributing new card material. Enabling High Security mode is impractical.

The last option to prevent card cloning is to use a two-factor authentication of RFID card and pin combination, which is supported by many of the readers around campus. Although this option is expensive because it requires replacing the RW300/400 models, these units are already vulnerable and will need to be replaced in the near future anyway.

Miscellaneous Door Security Issues

In the course of this work, I also observed some general door security issues I would like to comment on.

As mentioned early the INTEGRA system deployed in PCV is vulnerable to an attack documented by Cody Brocious[3]. Onity has issued a fix[1].
It is well known that handle doors, or doors in general that are not flush to the floor can be bypassed with a device to extend under the door and accurate the handle. I believe the persons in charge of resource security at Cal Poly are aware of this, as the door to the keying office was flush with the floor.

It should be noted the keying office on Cal Poly campus also utilizes RFID readers, which I consider a serious security concern. The keying office should not rely on electronic proprietary authentication to such a critical facility. Electronic locking technology is still in its infancy in comparison to conventional locking mechanisms.

Conclusion and Future Work

Every cryptographic system except those proven mathematically unbreakable are eventually defeated in time. Whenever a new electronic locking device, much like a physical locking device it will eventually need to be replaced. However, there are ways to prolong the life of these devices. Using 2-factor devices, which can be found around campus, is one good way to reduce dependence on the keying medium. Installing security cameras or otherwise monitoring critical resources is a good way to reduce dependence on the authentication medium, which are card readers in this case.

Managing the budget and security of any campus is a daunting task and I do not doubt the iClass readers were purchased with these ideas in mind. Unfortunately, the iClass readers are relatively new. These new electronic locking systems are relatively untested by time in comparison to conventional locking mechanisms.
Despite the fact that electronic locking systems are new, that does not excuse leaving vulnerable systems in place. The persons in charge of campus security should keep appraised of the technologies used to protect physical mediums. The persons in charge of budget should remember the ethical implications of leaving insecure doors in place.

I expect electronic locks will continue to become more popular and so I consider it imperative that Cal Poly address the vulnerable electronic locks on campus if we are to continue to use them.

To continue this work, I would investigate the difficulty of producing and maintaining secure electronic locking systems. I am particularly interested in making verifiably secure devices. As physical locks have evaluation criteria and expected times of compromise, I would like to investigate implementing the same for electronic devices.
References


