Home Insecurity: No Alarms, False Alarms, and SIGINT

Logan Lamb
lamblm@ornl.gov

ABSTRACT

The market share of home security systems has substantially increased as vendors incorporate more desirable features: intrusion detection, automation, wireless, and LCD touch panel controls. Wireless connectivity allows vendors to manufacture cheaper, more featureful products that require little to no home modification to install. Consumer win, since adding devices is easier. The result: an ostensibly more secure, convenient, and connected home for a larger number of citizens. Sadly, this hypothesis is flawed; the idea of covering a home with more security sensors does not translate into a more secure home. Additionally, the number of homes using these vulnerable systems is large, and the growth rate is increasing producing a even larger problem. In this paper, we will demonstrate a generalized approach for compromising three systems: ADT, the largest home security dealer in North America; Honeywell, one of the largest manufacturers of security devices; and Vivint, a top 5 security dealer. We will suppress alarms, create false alarms, and collect artifacts that facilitate tracking the movements of individuals in their homes.

1. INTRODUCTION

Home security systems have advanced tremendously in the past 25 years. They have evolved from simple systems composed of wired sensors, keypads, and control panels to a central hub for all home security and automation needs. Newer home security systems have incorporated most every advancement in consumer electronics to make more featureful systems including touchscreens, two-way communication, wireless sensors, and wireless home automation. Some can even be controlled from a smartphone.. This rapid incorporation of new technology to create innovative features not only increases the attack surface of the system, but also reduces the resources expended on the upkeep of legacy features. Because of this trade-off, allocating more resources for expansion of features instead of maintenance, we arrive at the current situation where cutting edge security systems are still using wireless protocols created 20 years ago. In this paper, we will demonstrate how this is a major security risk that has no clear remediation path. We will explore the motivations of the adversary. We will develop a model for the adversary and the home security systems. Using the developed models, a methodology will be developed for evaluating the efficacy of the adversary's attacks. Then, we will cover the attack primitives that are available to the adversary and their use cases. We will then move on to the application of the attack primitives: we will suppress alarms, create false alarms, and collect artifacts that facilitate tracking the movements of individuals in their homes. We then apply these attack primitives to three different security systems. We conclude our analysis by observing and explaining the efficacy of these attack primitives.

2. MOTIVATION

Consumers purchase home security systems to be safe in their residences. These systems ostensibly protect both the valuables of the occupants and the occupants themselves. Adversaries have repeatedly demonstrated the ease of cutting the phone lines which alert the monitoring companies. This is a well known attack, demonstrated in approximately 25% of invasions [Chianis 2014]. Because of this and the ease of installation, many consumers are advised to purchase wireless security systems. It has been demonstrated the cellular link back to the monitoring company can be compromised [Porter and Smith 2013], and that some wireless home automation systems can be compromised as well [Fouladi and Ghanoun 2013]. Subverting magnetic and PIR sensors so they never communicate an alarm has also been demonstrated [Porter and Smith 2013].

All of these attacks attempt to accomplish similar goals. As the adversary, we would like to subvert these systems so that they provide a false sense of security, and ideally become a liability to the occupants. To completely subvert the security systems, the adversary needs the ability to covertly infiltrate and exfiltrate the premises. To make the systems a liability, the adversary wants to monitor the behavior of the occupants and use the system to induce behavior in both the occupants and monitoring companies. Our adversary also wants a cheap, easy, and generalizable attack. The adversary believes he can accomplish these goals by attacking the intra-home wireless communications.

3. MODELS

3.1 Adversary Model

Intra-home wireless communications for home security systems have been in use for over 20 years. The adversary expects these communications to be vulnerable and fairly easy to compromise across multiple manufacturers. In addition, technology is trending towards wireless communications, so the adversary expects the attacks to be high yield. Given the attacks are a success, the attacks should be easy to commodify since software defined radios are becoming cheaper and more ubiquitous. Now that the adversary has decided on a wireless approach, what is required to accomplish the goals?

The adversary requires three attack primitives. The first is jamming of transmissions, which will suppress alarms and allow covert infiltration and exfiltration. The second is SIGINT, which will be used to intercept transmissions and monitor occupants. The third is replay, which will trigger false alarms and be used to induce behavior.

The adversary will have some stringent constraints placed on him in hopes of providing the cheapest, easiest, and most generalizable solution. The first constrains knowledge acquisition techniques. There will be no dumping of ROMs or firmware, there will only be black-box testing. The second constrains possible attacks. There will be no fuzz testing or crafting of malicious transmissions. The adversary will be restricted to the three available attack primitives.

3.2 Security System Model

We model intra-home security system communication as a directed graph with two edge labels (communication types) and four vertex labels (device types). The two communication types are:

1. vulnerable
2. secure

The 4 device types are:

1. sensors (e.g. door sensor, glass break, motion detector)
2. alerting devices (e.g. keypad for occupants, control panel for monitoring companies)
3. bridges
4. other

Sensors are devices that trigger an event when some criteria is met. They generally support one-way communication and simply broadcast their event using their supported communication type. Some more advanced sensors contain some state, and will broadcast a periodic heartbeat and alert when their battery is low. Alerting devices report the system state, the aggregate of all events received by the device, to an authority, i.e., the occupants of the protected area or the monitoring company. The third device type, bridge, is any device that simply passes transmissions along. Bridges act to extend range and translate transmissions from one communication type to another. Our fourth device type, other, is to cover all other devices that do not fit the other types. Given the adversary model, we treat wired communication and non-legacy wireless communication as secure.

We model home security systems as directed graphs (digraphs) since the transmissions from sensors happen regardless of whether or not the alarm devices are in an armed or disarmed state. Also, alarm devices generally only signal an event if they are armed and receive a transmission from one of the sensors. So, communications are modeled as originating at the sources (sensors) flowing through the graph (through bridges and alarm devices) to sinks (alarm devices).

As can be seen in Figure 1, the digraph for the exemplar Honeywell system is composed of 5 sensors (blue nodes), 2 alarm devices (red nodes), 5 vulnerable communications channels (solid edges) and 2 secure communication channels (dashed edges). The black box encompassing the blue and red nodes signifies the barrier of the protected area. So, all communications occur within the protected area except for the single communication channel connected to the monitoring company (cowboy badge).

Since all events pass through the keypad, the center red node, all the adversary needs to do is compromise the sensor communication channels and the keypad will never receive any events to alert the occupants or the monitoring company.

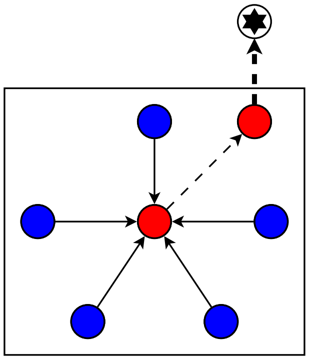


Figure 1: Honeywell System Digraph

3.3 Methodology

Given our objectives, system model, and primitives, the following is our general approach to analyzing new security systems:

1. Identify all devices and their supported communication types.
2. Generate a directed graph from sources (sensors) to sinks (alerting devices).
3. If any wireless communication channels exist, attempt our SIGINT primitive.
4. If a path exists from a source to a sink that involves a vulnerable communication channel, attempt jamming and replay primitives.

5. Evaluate the attained level of control and situational awareness of the system.

Before applying our methodology to two additional systems, we will show implementations of the three attack primitives when applied to the Honeywell system.

4. ATTACK PRIMITIVE IMPLEMENTATION

In this section we will detail the hardware and software required for implementation, the implementation of these primitives, and some of the capabilities that they provide.

The implementations will be targeting a Honeywell system. The Honeywell system is comprised of two 5815MN door sensors, 3 5800 PIR-RES motion detectors, a 6160RF keypad, and a Vista 20P control panel. This system was purchased approximately two years ago.

4.1 Required Hardware and Software

There are four prerequisites:

1. A software defined radio that is capable of transmitting and receiving on the frequencies used by the home security devices. We use a USRP N210 software defined radio with a WBX daughterboard.

2. A tuned antenna. We use several cut-to-length wire antennas.

3. Software to program the software defined radio. We use GNU Radio. GNU Radio is open-source, free, and supports the vast majority of SDRs on the market. It comes with a graphical tool, GNU Radio Companion (GRC), which is invaluable for general use and rapid prototyping. GRC is similar to Simulink and LabView with its flow-based programming. The output of GRC is a Python program. So, it is standard workflow to prototype in GRC and let it create the Python boilerplate.

4. A test system. We use the previously mentioned Honeywell system.

4.2 Tuning In

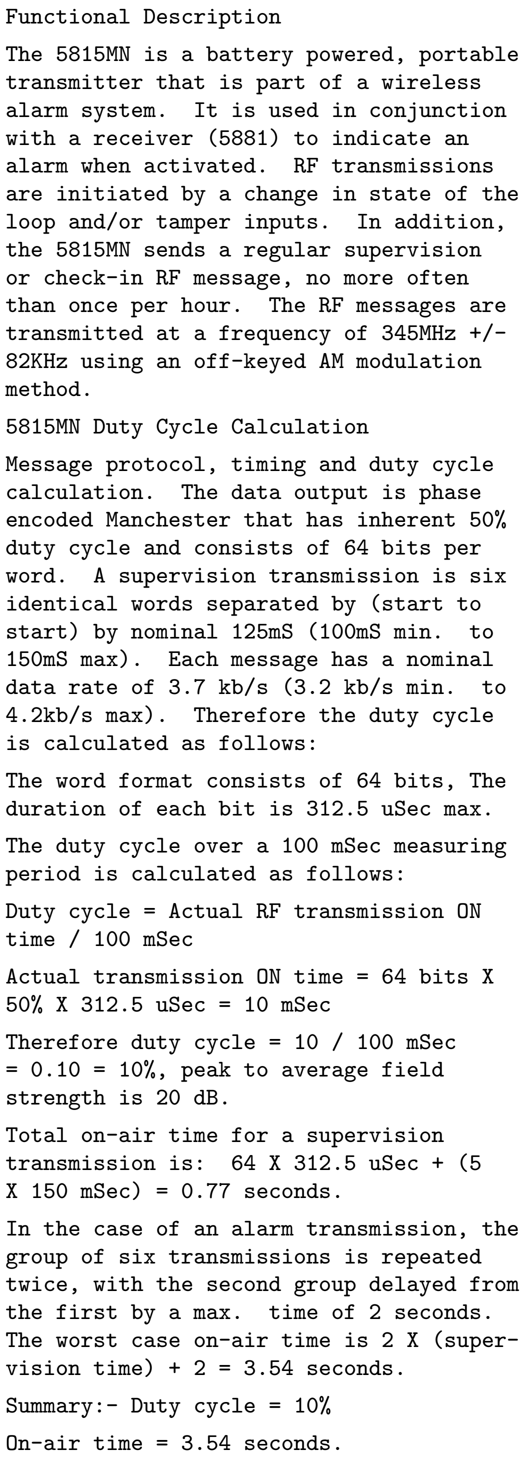
The first step is to figure out where in the frequency spectrum communications are taking place. This can be done using a dedicated spectrum analyzer, an SDR as a spectrum analyzer, or by simply consulting the FCC [FCC 2014]. We searched the FCCID of the 5815MN door sensor (FCCID: CFS8DL5815) and found the information in Figure 2.

We will be referencing figure 2 throughout the paper. For tuning in, the Functional Description provides us with the needed center frequency: 345MHz.

4.3 Jamming

4.3.1 Spot Jamming Implementation with GRC

This flow chart is simple. Our source is a random number generator with an output type of integer. Our sink is the USRP N210 with center frequency set to 345MHz and gain set arbitrarily high. We cannot wire these two blocks together because they are of different types. So we add our



Functional Description

The 5815MN is a battery powered, portable transmitter that is part of a wireless alarm system. It is used in conjunction with a receiver (5881) to indicate an alarm when activated. RF transmissions are initiated by a change in state of the loop and/or tamper inputs. In addition, the 5815MN sends a regular supervision or check-in RF message, no more often than once per hour. The RF messages are transmitted at a frequency of 345MHz +/- 82KHz using an off-keyed AM modulation method.

5815MN Duty Cycle Calculation

Message protocol, timing and duty cycle calculation. The data output is phase encoded Manchester that has inherent 50% duty cycle and consists of 64 bits per word. A supervision transmission is six identical words separated by (start to start) by nominal 125mS (100mS min. to 150mS max). Each message has a nominal data rate of 3.7 kb/s (3.2 kb/s min. to 4.2kb/s max). Therefore the duty cycle is calculated as follows:

The word format consists of 64 bits, The duration of each bit is 312.5 uSec max.

The duty cycle over a 100 mSec measuring period is calculated as follows:

Duty cycle = Actual RF transmission ON time / 100 mSec

Actual transmission ON time = 64 bits X 50% X 312.5 uSec = 10 mSec

Therefore duty cycle = 10 / 100 mSec = 0.10 = 10%, peak to average field strength is 20 dB.

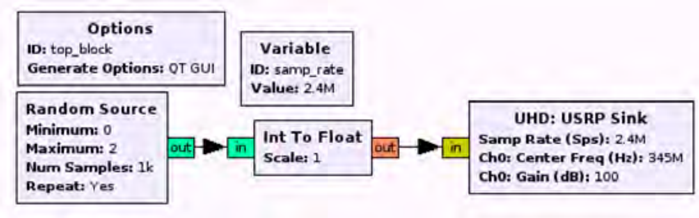
Total on-air time for a supervision transmission is: 64 X 312.5 uSec + (5 X 150 mSec) = 0.77 seconds.

In the case of an alarm transmission, the group of six transmissions is repeated twice, with the second group delayed from the first by a max. time of 2 seconds. The worst case on-air time is 2 X (supervision time) + 2 = 3.54 seconds.

Summary:- Duty cycle = 10%

On-air time = 3.54 seconds.

Figure 2: Excerpt from FCC filings



Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 2.4M

Random Source
Minimum: 0
Maximum: 2
Num Samples: 1k
Repeat: Yes

Int To Float
Scale: 1

UHD: USRP Sink
Samp Rate (Sps): 2.4M
Ch0: Center Freq (Hz): 345M
Ch0: Gain (dB): 100

Figure 3: Honeywell Jamming Flow-Chart

third block, a type conversion from int to float, and create a valid flow chart that can generate noise on 345MHz.

We found the spot jammer flow chart to be surprisingly effective on the Honeywell system. With this capability, an adversary can covertly infiltrate and exfiltrate from a protected area without the system alerting the occupants or the monitoring company. But there is a caveat. Manufacturers of home security equipment are aware of this attack, and have incorporated 'RF Jam' detection into most of their alarm devices.

4.3.2 Jamming with RF Jam Enabled

After enabling RF Jam on the Honeywell system the previous flow chart no longer worked. If left running for too long, the system would notify the occupants and monitoring company of the RF Jam event. Interestingly, the system did not notify the occupants until the flow chart had been running for about a minute, so we devised some tests to see how the RF Jam detection is implemented. Our first hypothesis is that it simply checked if the noise floor was elevated for a particular period of time. The second hypothesis is that after the system received a number of malformed packets it would trigger the RF Jam event.

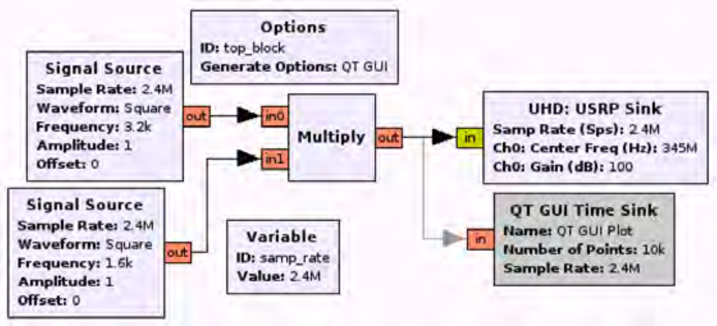
We tested the elevated noise floor hypothesis by running the spot jammer flow chart for 20 seconds, turning the jammer off for a second, and turning it back on. The code to do this was a simple modification to the generated Python program of the spot jammer. We found we could lower the off time to a quarter of a second and still avoid RF Jam detection.

We tested the arbitrary number of malformed packets hypothesis by creating a flow chart which broadcasts a simple square wave at the baud rate of transmission with duty cycle under 25%. The pulse jamming flow chart can be seen in Figure 4. The flow chart's purpose is not to jam the transmissions from sensors, but to mangle them. After testing, this approach was effective at jamming.

We found two approaches to jam transmissions without triggering RF Jam events. Given a choice, systems with RF Jam detection enabled are actually more desirable targets than without. An adversary can both suppress alarms for covert infiltration and exfiltration with active jamming and trigger alarms with the spot jamming.

4.4 SIGINT

There are multiple tiers of SIGINT. The first, and simplest, is the capture of RF transmissions. If the adversary can discern through observation what event the RF transmission is triggering, then they can replay the transmission and trigger the witnessed event. The second, less simple approach is to capture the RF transmissions and demodulate them to recover transmission packets. The adversary may not know exactly what the packets are communicating, but if the transmitted packets have no dynamic components, i.e., sensors always send the same couple of packets, then the adversary may be able to discern the meaning without fully reverse engineering the protocol. The last approach is full reverse engineering of the captured packets, which requires the most effort and has the highest payoff. We will now cover these three approaches to SIGINT.



Options
ID: top_block
Generate Options: QT GUI

Signal Source
Sample Rate: 2.4M
Waveform: Square
Frequency: 3.2k
Amplitude: 1
Offset: 0

Signal Source
Sample Rate: 2.4M
Waveform: Square
Frequency: 1.6k
Amplitude: 1
Offset: 0

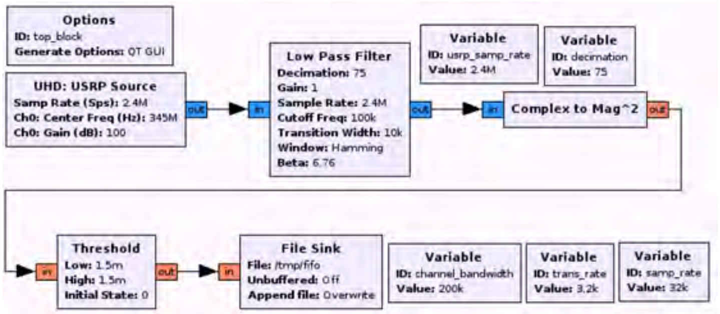
Multiply

Variable
ID: samp_rate
Value: 2.4M

UHD: USRP Sink
Samp Rate (Sps): 2.4M
Ch0: Center Freq (Hz): 345M
Ch0: Gain (dB): 100

QT GUI Time Sink
Name: QT GUI Plot
Number of Points: 10k
Sample Rate: 2.4M

Figure 4: Honeywell Pulse Jamming Flow-Chart



Options
ID: top_block
Generate Options: QT GUI

UHD: USRP Source
Samp Rate (Sps): 2.4M
Ch0: Center Freq (Hz): 345M
Ch0: Gain (dB): 100

Low Pass Filter
Decimation: 75
Gain: 1
Sample Rate: 2.4M
Cutoff Freq: 100k
Transition Width: 10k
Window: Hamming
Beta: 6.76

Variable
ID: usrp_samp_rate
Value: 2.4M

Variable
ID: decimation
Value: 75

Complex to Mag^2

Threshold
Low: 1.5m
High: 1.5m
Initial State: 0

File Sink
File: /tmp/fifo
Unbuffered: Off
Append file: Overwrite

Variable
ID: channel_bandwidth
Value: 200k

Variable
ID: trans_rate
Value: 3.2k

Variable
ID: samp_rate
Value: 32k

Figure 5: Converting RF to bitstream

4.4.1 RF Transmission Capture with GRC

The simplest of flow charts, we connect our USRP source with center frequency set to 345MHz to a file sink. In doing so, we store whatever is captured by the USRP while the flow chart is running. We will use the generated file at a later time for replay. The flow chart can be seen in Figure 4.

4.4.2 Bitstream Capture with GRC

From Figure 2, the following pertinent information is available to us:

- Center Frequency: 345MHz
- Modulation: off-keyed AM modulation (OOK)
- Baud Rate: 3200
- Line Coding: Manchester

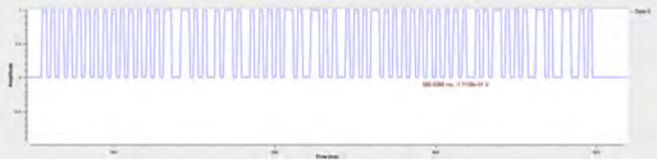


Figure 6: A Honeywell transmission converted to a binary signal

The purpose of the flowchart in figure 5 is to convert an OOK modulated signal to a bitstream. It is composed of a low pass filter, complex to magnitude squared, and thresholding blocks. The primary purpose of the low pass filter is to decimate the signal from the sampling rate of the SDR to something more manageable. We selected a decimation rate of 75 so that the output sampling rate of this flow graph will be ten times the baud rate of 3.2K. This relatively high sampling rate will allow us to easily detect long and short pulses. The complex to magnitude squared is used to convert the complex signal into something closer to a square wave. Finally, the thresholding block is used to convert the signal into a binary signal. The file sink dumps the binary signal.

The generated bitstream file is not just a step towards reverse engineering the protocol; it increases the effectiveness of replay attacks. By creating a bitstream, we have removed all the noise from the RF signal. So, when we replay we can increase the gain and not worry about distorting the binary signal. A sample output packet can be seen in Figure 6.

4.4.3 *Bitstream to Packets with GNU Radio and Python*

Now that we have a reliable bitstream, we can use additional pertinent information from Figure 2:

- Word Length: 64 bits

From Figure 6, it appears the signal has a preamble for synchronization. So, we'll leverage that to figure out the average number of samples per bit. Once we have an average number for samples per bit, we read from the stream until we have 64 and then proceed to manchester decode them. This is a pure programming exercise. Of interest though, it is very easy to fill the buffer of the SDR resulting in dropped samples. The implementation has a concurrent solution to consume samples quickly. One thread is constantly doing block reads from the SDR output file and removing dead air (all zeros). If there is an instance of live transmission it adds those samples to a deque which the program reads from.

We now have a reliable packet stream. For each door sensor in our the Honeywell system, we trigger door open, door closed, and door tamper events. We also set off the motion detectors. The packets captured from each device were static for each event type. So, a door open event will be the same every time it triggers for a particular sensor. The captured packets are in Figure 7.

```
# door sensor, serial:  A 031-6418
0xfffe84d40280512c
0xfffe84d402a0d1ef
0xfffe84d402e0506c
# door sensor, serial:  A 102-6691
0xfffe8faa83804d3d
0xfffe8faa83a0cdfe
0xfffe8faa83e04c7d
# motion sensor, serial:  unknown
0xfffe8cf96c00944e
0xfffe8cf96c021441
0xfffe8cf96c80174d
# motion sensor, serial:  A 070-4201
0xfffe8abec9003728
0xfffe8abec902b727
0xfffe8abec980b42b
# motion sensor, serial:  A 085-0206
0xfffe8cf91e00384b
0xfffe8cf91e80bb48
```

Figure 7: Honeywell Sensor Packets

4.4.4 *Reverse Engineering the Protocol*

Now that we have packetized the bitstream we can focus on reverse engineering of the protocol. Again, from the Duty Cycle Calculation documentation in Figure 2, it appears for each type of broadcasted message there is only one word which is repeated multiple times.

We now induce the behaviors detailed above (door open, door closed, tamper) in the two door sensors and get the results in Figure 7.

Now we focus our efforts on identifying static and dynamic parts of the packets. Within each door's packets the first 5 bytes are static, and for all devices the first 5 nibbles are static. Immediately we recognize what looks like a preamble and sync bit, `0xfffe`. For each door sensor, there is a static part of the message that is unique to the door sensor, `0x84d402` and `0x8faa83`. Ignoring the leading `0x8`, `0x4d402` and `0xfaa83` are the serial numbers of the door sensors in hex. All wireless Honeywell sensors start with an `A` so that part of the serial is implied. By identifying the serials in the packets, we now have the capability of uniquely identifying sensors.

The last 3 bytes of these packets are the only ones which are dynamic. The first byte of these three appears to be the packet type (`0x80`, `0xa0`, `0xe0`). That leaves only the last two bytes to be reversed. The last two bytes are most likely for integrity checks, so we run RevEng over our packets to see if the last two bytes are the product of a known CRC. Sure enough, they are CRC `BUYPASS`.

The completely reverse engineered protocol for these door

```
0x0fffe Preamble and sync bit
0x8 Unknown
0xXXXXX Device serial number
0x{80,a0,e0} Packet type
0xXXXX CRC16-BUYPASS
```

Figure 8: Honeywell Packet Format

sensors is in Figure 8. The correctness of the protocol is confirmed by applying it to the motion detectors.

4.5 Situational Awareness using SIGINT

We now have a solution to convert the RF transmissions from sensors into meaningful messages. The sensors transmit events regardless of the system's armed state. This is what allows us to accumulate information on occupants. The utility of the captured transmissions is directly proportional to the number collected. So, a single captured transmission in isolation does not provide much intelligence. However, a single transmission in the context of all captured transmissions can provide quite a bit of insight, allowing us to draw conclusions on habitual and anomalous behavior.

4.5.1 Differentiating sensors types

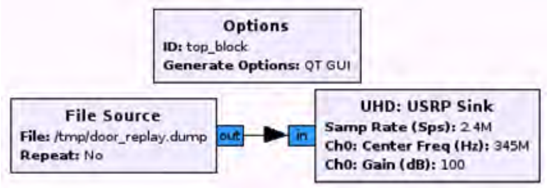
Doors and motion detectors share a common packet type, namely 0x80. Thankfully we have some other features that help us differentiate the sensor type. Doors sensors transmit on both opening and closing of doors, so if we see pairs of transmission type 0xa0 and 0x80 then we have a door open and close, respectively. If we encounter a 0x80 followed by 0x00 then it is a motion detector. In addition, some motion detectors will only transmit an event once every three minutes to conserve power whereas doors transmit every event.

4.5.2 Home Layout

The way in which sensors are placed in a home, fortunately, is sensible. Most homes will have less than a dozen sensors, and we can be assured their placement will be prioritized by the most high value and highly trafficked areas of the home. So, while we may not know where a motion detector is located in a home, if it is the only one in a home it will lie in a path that must be traversed to access the bedroom. The bedroom is the most high value room in a home. If we find multiple motion detectors belonging to a home, then we can look at the times the sensors are set off and figure out which one is most likely the motion detector protecting the bedroom (most likely the last one to transmit prior to a sleep cycle), and which one is the living room/dining room sensor (will transmit throughout the day most likely). The same reasoning applies to the door sensors. Occupants tend to add door sensors to all doors in a home that allow access to the interior (including the garage). So, If we find a door sensor that typically transmits around the time a resident goes to work that'll be the door sensor closest to the garage.

4.5.3 Multiple Residents

We can draw meaningful conclusions from the aggregate information of a system, e.g., when is the residence occupied and unoccupied. We can also tell when aggregate behavior deviates, e.g., emergency situation, birthday party with many attendees. We require additional intelligence on the occupants to differentiate between them. This area requires future work.



Options
ID: top_block
Generate Options: QT GUI

File Source
File: /tmp/door_replay.dump
Repeat: No

UHD: USRP Sink
Samp Rate (Sps): 2.4M
Ch0: Center Freq (Hz): 345M
Ch0: Gain (dB): 100

Figure 9: Honeywell Replay Flow-Chart

4.6 Replay

A replay attack involves conducting some level of SIGINT to acquire a transmission. Once a transmission is acquired, the adversary plays back the transmission to accomplish the original transmission's intent.

4.6.1 Replay Implementation with GRC

The source for our replay flow chart is one of the output files from the SIGINT step. The contents of the file could be the raw transmissions, bitstream, packet, or completely reverse engineered capture. The important part is whatever our source file is, we must do the inverse of the capture function to output on the USRP sink. We will focus on replay of a raw transmission capture. The flow chart is the inverse of the SIGINT raw RF transmission capture, and can be seen in Figure 9. So, this flow chart uses the captured file as the source and the USRP as the sink.

Replay is an effective attack on the Honeywell system. With this capability, an adversary can create false alarms for the monitoring company whenever the system is armed. Due to the ease of this attack, an adversary can cause false alarms at multiple protected areas to cause the monitoring company to misallocated resources. When targeting the occupants, the adversary does not have to rely on an armed system. The adversary can induce behavior by triggering particular sensors, e.g., basement door, hallway to bedroom, hallway to child's bedroom, to either attract or repel occupants to that area. This level of behavior influence requires a great deal of information on the occupants and the protected area.

5. APPLYING THE METHODOLOGY

Now that we have covered the models, attack primitive implementations, and methodology, we apply our methodology to two additional systems. The first system is detailed in section 4. In summary, the adversary has complete control of the Honeywell system and full monitoring capability.

5.1 ADT System

The ADT system is comprised of 4 door sensors, 3 glass break sensors, 1 motion detector, and a keypad control panel, all of which are manufactured by DSC. The primary differences between this system and the Honeywell system is the more advanced panel. The panel in this installation acts as both the keypad and control panel, reducing required hardware. It also alerts the monitoring company over GSM, so this system is completely wireless. It was installed less than a year ago by ADT.

Like the Honeywell system, all sensor communications can potentially be jammed and intercepted. See Figure 10.

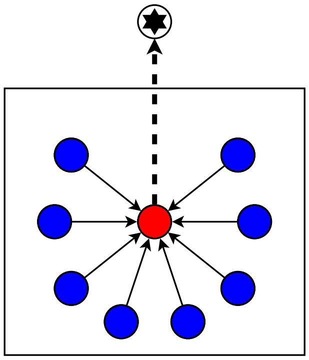


Figure 10: ADT System

The only change made to the jamming implementation in section 4 was to change the center frequency to 433.92MHz. We found the spot jammer to be very effective, giving the adversary covert infiltration and exfiltration capabilities. We attempted to enable RF Jam on the panel, but were unable to acquire the required installer code per ADT's policy. The adversary has covert infiltration and exfiltration capabilities with this system.

The changes to the SIGINT flow charts include changing the center frequency to 433.92MHz and removing the Manchester decoding. All SIGINT primitives were implemented, however the final reverse engineering effort was not taken. This was due to time constraints, and because the entire protocol format could be found in the FCC documentation for FCCID F5300NB912 [FCC 2014]. An excerpt can be seen in Figure 11. With minimal effort the adversary would have full monitoring capability.

The only change made to the replay implementation in section 4 was to change the center frequency to 433.92MHz. Replay attacks are effective, giving the adversary the capability to cause false alarms and induce behavior.

5.2 2GIG System

This is by far the most interesting system. It is composed of 4 wired door sensors, 1 wired motion detector, 1 12V control panel, 1 wireless 2GIG door sensor, 1 Go!Control Panel, and 1 2GIG takeover module. The system appears to be a new wireless system retrofitted onto an older-style wired system. Both the wired and wireless components were installed in a new house which was completed in 2014. 2GIG equipment, including the Go!Control panel in this system, is used by Vivint.

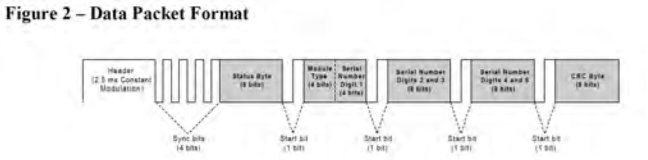


Figure 2 – Data Packet Format
Header (2.5 ms Constant Modulation)
Status Byte (8 bits)
Module Type (4 bits)
Serial Number Digit 1 (4 bits)
Serial Number Digits 2 and 3 (8 bits)
Serial Number Digits 4 and 5 (8 bits)
CRC Byte (8 bits)
Sync bits (4 bits)
Start bit (1 bit)
Start bit (1 bit)
Start bit (1 bit)
Start bit (1 bit)

Figure 11: ADT Packet Format

As can be seen in Figure 12, this system's topology is quite different from the two previous systems. This is the only system covered that has sensors wired directly to an alarm device that is capable of alerting the monitoring company. Unfortunately, the alarm device is not acting in that capacity. In fact, its sole purpose is to aggregate all of the wires and present them to the takeover module, which converts the wired transmissions into vulnerable, wireless transmissions. Because of this translation, the system is equivalent to the other two systems, but with fewer points of failure since the wireless transmissions of five sensors are radiating from one takeover module.

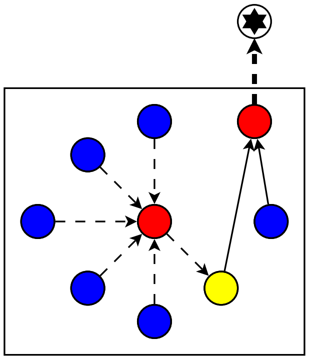


Figure 12: 2GIG System

No changes were made to the jamming implementations in section 4. We found the spot jammer to be very effective, giving the adversary covert infiltration and exfiltration capabilities. We enabled RF Jam on the panel and found the periodic jamming worked effectively with an on time of 50 seconds and off time of 0.20 seconds. So, the adversary has the capability to covertly infiltrate and exfiltrate.

No changes were made to the SIGINT implementations. While the modulation, line coding, and packet sizes were all the same as the Honeywell equipment, the contents of the packets diverged slightly. Despite the slight changes, we are still able to uniquely identify each device and the event types, giving the adversary full monitoring capabilities.

No changes were made to the replay implementation in section 4, giving the adversary the capability to cause false alarms and induce behavior.

6. OBSERVATIONS

The attack primitives are effective against all three systems despite different graph topology, hardware, and communication protocols. We present several likely explanations.

6.1 Simple Protocols

In each of these systems, the alarm devices implicitly trust the sensor communications and have no way of querying the device from which the transmission originated. The protocols seen in these security systems are very similar to legacy

15.231(a) Continuous transmissions such as voice, video or data transmissions are not permitted.
15.231(a)(1) A manually operated transmitter shall employ a switch that will automatically deactivate the transmitter within not more than 5 seconds after being released.
15.231(a)(2) A transmitter activated automatically shall cease transmission within 5 seconds of activation.
15.231(a)(3) Periodic transmissions at regular pre-determined intervals are not permitted. However polling or supervisory transmissions to determine system integrity of transmitters used in security or safety applications are allowed if the periodic rate of transmission does not exceed one transmission of not more than one second duration per hour for each transmitter.
15.231(a)(4) Intentional radiators which are employed for radio control purposes during emergencies involving fire, security, and safety of life, when activated to signal an alarm, may operate during the pendency of the alarm.

Figure 13: CFR 47 Part 15 Requirements

protocols, like Modbus, which lack authorization for commands and security against interception.

It is also interesting that each system had the same packet sequences to signify supervision messages (a sequence of repeated packets), and alert messages (two supervision messages separated by dead air). This was most likely done to reduce the time and monetary cost of implementing the protocols.

6.2 Legacy Technology

From the FCC documentation, Honeywell has been using the same Manchester encoded OOK scheme since at least 1998. Digital Security Controls, the manufacturer for the ADT system, has been using the same protocol since at least 2000. 2GIG, the manufacturer of the Go!Control panel which is used by Vivint, curiously adopted a communication stack very similar to Honeywell's. Like Modbus use in industrial control systems and ATMs running Windows XP, once a component is used long enough in a process it is very difficult to remove.

6.3 FCC Regulations

All of the sensors covered in this paper communicate using unlicensed transmissions which fall under the purview of FCC CFR 47 Part 15 [ECFR 2014]. FCC Part 15 compliance is required for all electronics sold in the USA, and assures electronics do not cause electromagnetic interference. The wireless sensors are Part 15 compliant, but also meet more stringent requirements since they are intentional radiators, i.e., they communicate wirelessly. A sampling of these requirements can be seen in Figure 13.

Due to the FCC restrictions, the manufacturers of these devices are limited in their radiated power, transmission time, and frequency of periodic heartbeats. These regulations restrict the possible features of devices, including security.

7. CONCLUSION

In this paper, we identified the primary motivations of adversaries and likely ways in which their goals of undermining home security systems can be met. We modeled the adversary, home security systems, and created a general methodology for evaluating the susceptibility of systems to the adversary's attack primitives.

Based on these models, we implemented the adversary's attack primitives and applied them to three different security systems. For each of these systems, the adversary has the capability to covertly infiltrate and exfiltrate, induce behavior in the occupants and monitoring companies, and monitor the activities of the occupants.

8. REFERENCES

[Chianis 2014] Alexia Chianis. 2014. 8 Surprising Home Burglary Statistics. (May 2014). "http://www.safewise.com/blog/8-surprising-home-burglary-statistics/"

[ECFR 2014] ECFR. 2014. FCC CFR 47 Part 15. (June 2014). http://www.ecfr.gov/cgi-bin/text-idx?tpl=/ecfrbrowse/Title47/47cfr15_main_02.tpl

[FCC 2014] FCC. 2014. FCC ID Search Form. (June 2014). http://transition.fcc.gov/oet/ea/fccid/

[Fouladi and Ghanoun 2013] Behrang Fouladi and Sahand Ghanoun. 2013. HONEY, I'M HOME!! - HACKING Z-WAVE HOME AUTOMATION SYSTEMS. In *Black Hat*.

[Porter and Smith 2013] Drew Porter and Stephen Smith. 2013. LET'S GET PHYSICAL: BREAKING HOME SECURITY SYSTEMS AND BYPASSING BUILDINGS CONTROLS. In *Black Hat*.