LEARN HOW TO CONTROL EVERY ROOM AT A LUXURY HOTEL REMOTELY: THE DANGERS OF INSECURE HOME AUTOMATION DEPLOYMENT

Jesus Molina
@verifythentrust
security@nomeames.com

1 Abstract

The St. Regis ShenZhen, a gorgeous luxury hotel occupying the top 28 floors of a 100 story skyscraper, offers guests a unique feature: a room remote control in the form of an iPad2. The iPad2 controls the lighting, temperature, music, do not disturb light, TV, even the blinds and other miscellaneous room actions. However, the deployment of the home automation protocol contained several fatal flaws that allowed an arbitrary attacker to control virtually every appliance in the hotel remotely. I discovered these flaws and, as a result, was able to create the ultimate remote control. The attacker does not even need to be in the hotel - he could be in another country. This white paper discusses home automation and the anatomy of the attack: an explanation of reverse engineering of the KNX/IP home automation protocol; a description of the deployment flaws; blueprints on how to create an iPad Trojan to send commands outside the hotel; and, of course, solutions to avoid all these pitfalls in future deployments.

2 Introduction

In Hollywood movies, cyberattacks end with shocking results that stun the audiences and ruin the movie for security researchers. Traffic halts in LA as Seth Green switches all lights to red in the Italian Job. The bad guy in Jurassic Park shuts down the electrical fences and the dinosaurs roam free. These epic attacks sound whimsical, but nowadays appear closer than ever with the advent of what people call the Internet of Things (IoT), a vision where every single electronic device is networked.

Figure 1: The view

But if there was any doubt, imagine this scene in a James Bond movie. The bad guy and his minions guard the bomb detonator in a hotel suite. Time is running out, and Bond cannot access the room and fight the guards. All seems lost. Suddenly, his cell rings, and after grabbing it he hears Qs desperate voice (the techie guy). “Why are you so worried James?” says Q in an arrogant tone “In 20 seconds I will provide you with the distraction needed to do your job”. James hangs up in disbelief, cursing the kid (in the past Q looked like a revered old scientist, now he looks like a startup billionaire). Suddenly, every light and TV in the hotel starts flickering. The drapes go up and down erratically and deafening music roars from every room. All the guests start fleeing in panic and the guards are no exception. James aims the gun at them from a safe distance, muttering “Rats, I need to learn Python”.

No way, you will say. No one can take over a building in this fashion. Not even the super smart guy from the movie. Fear not, this paper presents a real scenario, and for added theatrical charm the location could not be any more perfect: a beautiful five-star hotel in China, absolutely worthy of a James Bond film.

But before going into the gory technical details, let’s first discuss why we are here: because the situation described in this work is by no means unique, and more hotels and residential buildings will follow suit.

This work is not intended as a critique of the security decisions of the property itself. The targeted hotel performed due diligence after being notified of the problem, and being a trailblazer in a new and upcoming technology always has drawbacks. This work intends to be a cautionary tale so that future implementations avoid the pitfalls that would allow the chance for this epic movie scene to come to life.

3 Home Automation

In commercial buildings, the automation of electronic components (HVAC, lighting) has been around for a long time. Commercial building automation systems need to know just a little about the owner. Sensors are sufficient for learning the usage patterns of the building and the environment circumstances. Home automation (or residential building automation), while technically similar, had an erratic component that until today was difficult to track, understand and communicate with: the person living at the home. Home automation is user centric, and hence it also includes other items not usually present in commercial buildings, in particular, entertaining appliances such as music, TV and little pet robots (in my personal case). The usage patterns for entertainment depend heavily on the user and hence are hard to predict.

The advent of new technologies such as predictive learning, and most importantly, the red hot IoT provides a path to better automate your home by having all electronics networked to you and your desires. You may compare home automation to your TV remote, just on steroids, and with better knowledge than you about which channel to watch. To exemplify, by deploying home automation technology you may be able to light up the house when you enter, change TV channels through your phone, get toasts when your stomach grumbles and even have a drone deliver fresh bread every morning at say, 8 AM (on Sundays at 10 AM).

The astute reader (who also may yell "you kids get of my lawn!" most often than not) will protest and state that the beloved home toaster is sacred and will never be networked. But few doubts exist that home automation is already present and here to stay, with its deployment and utilization rising dramatically. According to Reuters [1] home automation is a $1.5 billion business and expected to rise to remarkable estimated $2.5 billions by year 2015. Leaving behind our craving for economic statistics to demonstrate every point, home automation is overdue: we possess the technology to make our lives way comfier, and home automation will significantly reduce the energy footprint in everyone's daily life. Machines love the earth way more than we do - we program them when we are in our best spirits. That is rarely the case when we finally reach home after a couple of beers and make it to bed leaving the lights and TV on, with the remote falling out of our hand and mother earth being just a loving figure in our dreams.

3.1 Security in Home Automation

Home automation will make our lives more comfortable, help the environment, and in the long run provide significant savings reflecting the decrease of energy consumption. However, home automation presents challenges as it requires a set of technological elements to be orchestrated together. For the end provider, it is a little bit like making a burger: everyone understands what the end result should be, but the ingredients vary significantly from one home deployment to the other. And that lack of standardization is always a field day for attackers out there.

Hence, to the home automation party an uninvited guest, security, arrives - there is always one. And is not an easy one to handle. The mighty struggle between usability versus protection seldom appears more acute than in home automation. The user requires simplicity (a PIN to operate my toaster? What is this, Fort Knox?), while the evil guy next door stands ready to create havoc by performing an orchestrated toaster uprising.

Second, it requires all the complicated components to be (at least partially) networked wirelessly, and possibly accessed from outside the home. In the past, building automation was performed by wired buses. These buses, initially proprietary, were slowly standardized in several protocols. As always, each continent chose its own darling (e.g., KNX in Europe), but they provide similar features. These protocols aided enormously in creating an ecosystem to buy and sell new appliances to network electronics without having to be bound to a single vendor. However, they did not add security (meaning authentication and encryption) as a part of the standard. After all, the scenario was totally different when they were created, with the building itself protecting the access to wired networked elements. A new standard was developed for wireless commercial building automation with strong security: Zigbee. However, WiFi for home networking is as extended as it gets, so legacy standards are still able to thrive under that umbrella.

If you already deployed home automation at your place, it probably works like this: you utilize a panel or your phone to connect to a router, and the router connects to the device. The connection, if it is your home, is done wirelessly from the panel, and the other devices are wired to a digital actuator. You can also connect directly to the device if it is a new appliance such a smartTV or an internet camera, utilizing an API that it presents to the world. For example, if you have some sort of streaming dongle such as the ROKU$^{\text{TM}}$ the application in your phone sends HTTP requests to the included RESTful API presented by the device. This API is available to any developer, so anyone can scan the network for a ROKU and contact it using any application out there or the one home-brewed. There is no security in this process, as that will force the user to add some sort of authentication token and that is a hassle for most scenarios. To control devices that are a little more backwards, say, a bulb, blinds, old TVs or temperature, you need to first connect them to a home automation hub, and that hub will connect to the panel.

So where is the security? The cybersecurity of home automation in a vast majority of cases is in your wireless key, usually WPA2. Obviously, if someone gets a hold of your WPA2 key, they can go ahead and take control of your home, but it is a fair assumption that if the key is sufficiently strong it will take some time for that to happen.

3.2 Hotel Room Automation

Hotels are unique in that they combine requirements from both commercial and residential spaces. Adding home automation to guest rooms as an added amenity provides hefty incentives for hotels. Everything that applies to a single user, applies to every room in the hotel: cost savings, guest satisfaction and increased utilization of amenities. Bundling the room control with other offerings, such as in-room dining or movie selection, seems to increase overall spending by the guest in the room, at least according to Intelity [2], a hotel automation vendor.

Hotels time-share the rooms and everything in them (lights, TV, HVACs, entertainment units), so they fall into the tragedy of the commons when it comes to energy usage. Most guests care little about leaving the light on at a hotel. While attempts have been made to correct the problem, such as forcing you to "switch on" the room by placing your card in a receptacle (ever come back to your room to pick up that phone you left charging, just to realize it is still dead?), home automation is a far better answer.

Customer comfort increases dramatically by utilizing centralized room controls too. When you arrive to a hotel room, you don't know what appliances you can switch on. Is there music? TV channels? Where is that light switch? All that guess work is over if you can automatically discover every switchable element in the room just by looking an in-room IPAD, or better still, an application in your own iPhone or iPad.

4 KNX

According to their webpage, KNX is "the world's only open Standard for the control in both commercial and residential buildings". It goes on by saying "KNX is therefore future proof" [3]. The reality is that KNX is as open as the VIP section in a Las Vegas club: it is open as long as you pay (a lot) to get in. For the second claim, in the advent of wireless communications and transparency, it may at least be aging. All the information in this section is publicly available or inferred from open source code, as public access to the specifications does not exist.

KNX is a widely deployed bus communication standard. It is the successor of the EIB/Instantbus European Standard, developed in the early 1990s. The St. Regis chose KNX to perform home automation, possibly because it is a Chinese Standard (GB/T 20965). The core protocol provides details to connect actuators to appliances. To get on with the new funky times we live in, KNX may be encapsulated inside an IP packet and sent over mediums such as wireless or the internet, and this version is called KNX/IP [4]. A KNX/IP frame is a connectionless datagram (UDP) envelope for a payload named cEMI in the KNX standard. The payload carried the protocol commands for connection, and, what is more important for an attacker, the end message to the KNX backbone (see Fig. 2).

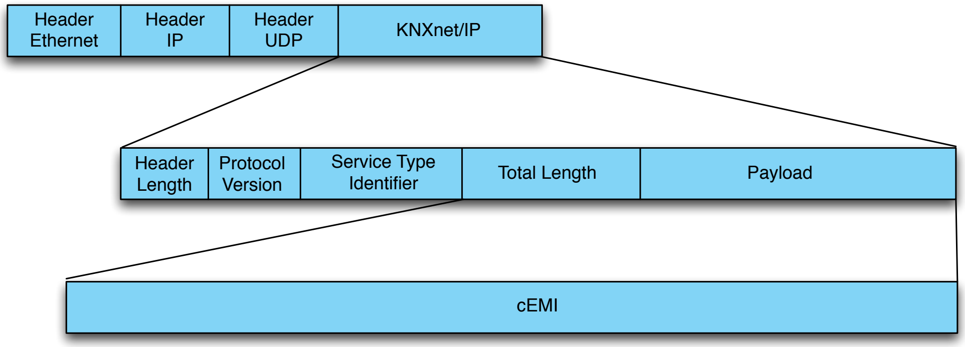

Figure 2: KNX/IP packet

The protocol provides several commands and modes of operation, for both communication with the end devices and for configuration. The KNX/IP mode utilized at the St. Regis was “Tunnel” mode, where a KNX/IP router tunnels KNX message requests from the IP backbone to the KNX network. The notation for KNX addresses is in the form A/B/C. It works in a similar fashion as IP addresses. The first and second digits define subnets, and the last digit defines the actual address of the element in the subnet. For example 1/2/3 represents a device with an address of 3, on 1/2 subnet.

The cEMI frame, among other fields, contains the address of the sender, the address of the receiver, the command type and command payload, which may or may not exist depending on the command type. For the interested, one of the few (free) available KNX open documents describes the data types of the payload [5].

To send a message, the protocol performs a simple sequential handshake (every message is followed by an ACK or fail message from the router). The handshake first starts the communication by sending two datagrams in order. The first is a CONNECTION_REQUEST.

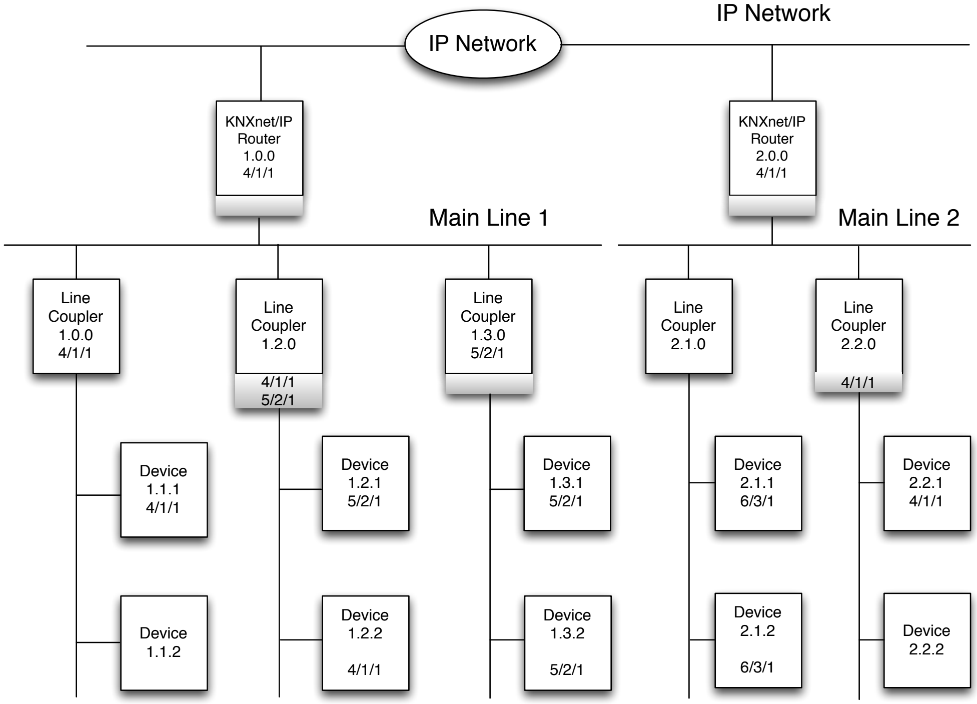

Figure 3: Example KNX/IP network

The second is a CONNECTIONSTATE_REQUEST. After this you can start sending messages to the KNX backbone by using a TUNNELLING_REQUEST message. Once you are finished, you terminate the communication by issuing a DISCONNECT_REQUEST. The Tunneling request command stands out in importance for practical purposes, as it carries the end action for the networked electrical appliance.

Listing 1 shows an example of a tunnel request (treq) for switching a light bulb on a KNX address 1/0/2 [6].

Listing 1: Tunnel Request example

```
treq = [06 10 04 20 00 15 04 49 00 00 11 00 be e0 00 00 08 02 01 00 81]
/*  TUNNELLING_REQUEST */
/*  Header (6 Bytes) */
treq[0] = 0x06; /* 06 - Header Length */
treq[1] = 0x10; /* 10 - KNXnet version (1.0) */
treq[2] = 0x04; /* 04 - hi-byte Service type descriptor (TUNNELLING_REQUEST)
*/
treq[3] = 0x20; /* 20 - lo-byte Service type descriptor (TUNNELLING_REQUEST)
*/
treq[4] = 0x00; /* 00 - hi-byte total length */
treq[5] = 0x15; /* 15 - lo-byte total length 21 bytes */
/* Connection Header (4 Bytes) */
```

```
treq[6] = 0x04; /* 04 - Structure length */
treq[7] = iChannelID & 0xff; /* given channel id */
treq[8] = 0x00; /* sequence counter, zero if you send one tunnelling request
    only at this session, otherwise count ++ */
treq[9] = 0x00; /* 00 - Reserved */
/*  cEMI-Frame (11 Bytes) */
treq[10] = 0x11; /* message code, 11: Data Service transmitting */
treq[11] = 0x00; /* add. info length (bytes) */
treq[12] = 0xbc; /* control byte */
treq[13] = 0xe0; /* DRL byte */
treq[14] = 0x00; /* hi-byte source individual address */
treq[15] = 0x00; /* lo-byte source (replace throw IP-Gateway) */
treq[16] = (destaddr    8) & 0xff; /* hi-byte destination address (20: group
address) 4/0/0: (4*2048) + (0*256) + (0*1) = 8192 = 20 00 */
treq[17] = destaddr & 0xff; /* lo-Byte destination */
treq[18] - 0x01; /* 01 data byte following */
treq[19] - 0x00; /* tpdu */
treq[20] = 0x81; /* 81: switch on, 80: off */
```

As an apparent sign of good will, universities have been able to work with the standard and as a result, there are open source projects for hobbyist and researchers. One of the tools created from this collaboration is the excellent eibd [7]. The eibd software package is the only tool an attacker needs to send messages to the KNX bus, as it implements the KNX/IP UDP wrapper and the tunneling handshake. Another method is to code the handshake yourself, which is very simple.

5 Case Study: The St. Regis ShenZhen

5.1 A beautiful hotel

For visitor to the ShenZhen area (the “Silicon Valley” of China), I recommend a stay or at least a visit to the St. Regis. The property is nothing short of stunning: placed at the top of the K100 building, this hotel provides an unrivaled experience for guests in the area: a personal butler, dwarfing area views, freestanding bathtubs next to floor-to-ceiling windows, a pool claiming half of one floor, and so far up that even the white pollution smog so pervasive in the Shanghai-Shenzhen area appears mystical and benign. Every room features an IPAD2^{TM} (we will call it the iPad from now on) loaded with an application controlling every electronic device in the room. Four major components provided the room automation for this property: The iPad, a wireless communication channel, the automation protocol (KNX) and the KNX backbone.

5.2 The Wireless Communication Channel

Wireless internet is now widely available in most hotels. Smaller hotels and motels provide the guest with the network WiFi key, but this solution is not scalable for bigger properties. Besides, hotel chain guests suffer the internet fee malady: the requirement to pay for the internet, even after spending a large amount of money for just being in the room (or in the case of Vegas, a “resort fee”). This annoying and extended practice also faces an interesting security flipside: to charge guests, the hotel deploys a captive portal to keep track of internet access. The captive portal first redirects the browser to a page asking for the guest name and room number. After providing the right information, the captive portal places your device’s MAC address into a white-list, and the guest is then able to connect to the internet. But to access the captive portal, this solutions grants initial access to the network to guests and strangers alike (you just cannot access the internet, but can surely “see” other guests). As a result, anyone can “listen” or even intercept your communications while you are connected to a hotel network. For the paranoid or not so paranoid, a hotel network should be treated as an adversarial network: bad stuff may happen to you, so methods such as using a VPN are required for security and privacy. As this is an “open” network, and is already deployed, the guest network could be used for other purposes, such as, you guessed it, interconnecting devices in a room.

5.3 the iPad

The iPad in the room does not provide any physical security. A guest can unplug it from its power attachment and transport it around freely. The guest can also attach it to his or her own computer and sync the applications, verify and modify the configuration settings and reboot it. At launch, the iPad shows (in full screen) the only application installed: a room control amenity, created by the company AYcontrol. The application presents several tabs to the user. Each tab allows guests to communicate with the room’s networked devices: TV (including channel selection), temperature, outside lights (Do Not Disturb), inside lights, blinds and music. The iPad also can allow selecting scenarios, bundling different room actions to create a certain mood (romantic, night, day and so forth). The iPad IP address was pre-configured in the iPad.

When the application is launched, the iPad sends two types of packets. The first one is a connectionless datagram (UDP) packet directed to a multicast address. This message is sent periodically by the iPad and contains basic information: the iPad IP address and room location. This message’s role is uncertain, but is most likely used for maintenance to keep track of the iPad location. This message was not required to communicate with the end connected devices. The second message type is also an UDP packet, but with a different payload. In this case, the UDP payload forms a KNX/IP protocol communication to a fixed IP address, and they are triggered while pressing a button in the iPad application requesting an action, such as switching on the lights. The KNX/IP IP destination was unique for each room and correlative for adjacent rooms. An attacker could easily infer the pattern and create a map between room and IP address of the KNX/IP router even without collecting any traffic from that room. The messages seem to follow the KNX/IP protocol requirements, except for a field that was modified from the standard for no apparent reason. However, an attacker could easily change these fields to mimic the slightly different packet.

5.4 The KNX network

The CEMI frame contains several fields, must unnecessary to perform the attack. An attacker needs only to understand what the "moving parts" of the protocol are. In this case, an attacker only requires the IP address inside the CEMI frame, the KNX destination address, the action code for that address and the payload (if any) for that destination in order to modify the IP address for every room (the destination). The source IP address is not necessary - there is no mechanism for checking the source IP address by the router.

The IP address for each room provided (at least) access to two different KNX subnets. The first subnet contained every element controlled by the automation control in the room. The address space assignments for each room where correlative - after collecting four or five room addresses, an attacker can easily guess all the room KNX numbers, similarly as performed with the IP address.

The second subnet, that can be accessed by every KNX/IP router in a floor, contained element outside the room. That IPAD action for that "floor" KNX address space was to switch on/off the "Do Not Disturb Lights" and the "Make Room" light.

The KNX network also responded to administrative and configuration commands. However, its modification was not necessary to allow arbitrary KNX commands to go through.

5.5 The Attack

With the knowledge of the KNX/IP router and KNX address of the room, the KNX address of the appliance and a dictionary of actions, the adversary can send any arbitrary action to any room, as long as it followed the KNX protocol sequence to transmit. The dictionary of actions is simple to create - just press every button on the iPad application and record the action and payload sent by utilizing a network sniffing tool such as Wireshark. To create a complete map of each room an attacker can either listen for iPads communicating (as discussed, the iPad "sings" its room number and IP address periodically). Or, a simpler method is to find an excuse to change rooms.

Once armed with this knowledge the attack is trivial: send the action to the target device, using the IP/KNX address. KNX does not provide any free software of application, but there are many open source software solutions that implement the KNX protocol. One such application is `eibd`. You can launch the `eibd` daemon in tunnel mode with the listen local option, and set it to the target IP of the room. As an example, let's suppose the address for a light is 3. The action to switch on is 80, and to switch off is 81. Finally the pair KNX room subnet and router IP address are 2/0 (KNX) - 172.31.20.160 (IP). After this we can launch `eibd` with the target IP address as a daemon. This action will perform the first two handshakes of the connection sequence and keep the connection alive. After that, the `eibd` will listen to request that can be sent using an application (also provided by `eibd`) writing to the local file. A sample sequence to switch on/off a light is presented in Listing 2.

Listing 2: Sample commands

```
#eibd -- T -listen-local ipt:172.31.20.160
#groupswite local:/tmp/eib 2/0/3 80 (Switch on light)
#groupswite local:/tmp/eib 2/0/3 81 (Switch off light)
```

5.6 Attack Scenarios

From here, it is trivial for an attacker to create any hack scenario: raise all the blinds at the same time (note: `eibd` does not provide a method to send parallel messages to different IP destinations, so in this case an attacker needs to code the handshake itself, which is simple), trojanize the iPad to control every room instead of only your own (the ONE iPad), or just prank the neighbor. As the only requirement to perform the attack is access to the local open network of the hotel, you can just point an antenna to the building, bridge the hotel network to the internet and start sending commands. A fancier option is to install an iPad application that connects to an external network regularly waiting for commands. There is nothing to stop the attacker from rooting the iPad, or replacing it entirely with another one.

The attacker could go deeper. First, the router configuration seemed modifiable and several configuration attacks may have been possible at the infrastructure level. Second, there seemed to be devices in the KNX network other than the ones controlled by the iPad application. Clues pointed to it: for example, in the "outside" corridor address space, many "ghost" addresses not used by the iPad (blank address with no apparent reason) existed.

5.7 Solutions

The hotel room control featured multiple elements which made it vulnerable to a hack. In this case, an iPad with a beautiful room control application; a wireless communication channel; a KNX/IP router; and a KNX wired network to connect to the appliances. It is common not to follow the tired adage advising to provide security not as an afterthought but from the start. But in this case the adverse results were quite telling.

The three elements do not provide any possible security mitigation. iPads are gadgets intended from personal use, and do not provide a multiuser environment to perform access control. The network for the hotel requires openness for guests to access and pay for the internet.

Finally, and this is quite embarrassing, there is no option to provide security utilizing the KNX protocol. The security problems in KNX/IP are known in the research community for years [8]. Earlier works suggest the use of ZigBee for wireless communications [9]. In [10] researchers discuss the lack of security on KNX/IP and that in the standard only some "rudimentary" countermeasures are presented. Researchers proposed modification to the protocol called EIBsec, but only prototypes have been implemented. It seems that in the last specification version (2.1, published October 2013) the KNX consortium added some security, but as specification are not available to the general public, these claims cannot be reviewed. So at this point we will assume KNX provides no security whatsoever in the protocol.

Given these facts, the only viable option (without changing the architecture significantly) is to provide a secure tunnel between the iPad and the KNX/IP router. This could easily be achieved by adding a certificate and tunnel code in the iPad, and a network device preceding the KNX/IP router that enables the secure tunnel (by deploying a commercial solution or using open source tools and hardware). The tunnel must provide mutual authentication, which could be achieved with different secure connection protocols, such as SSL. However, in this environment an attacker may steal the certificates from the iPad. To prevent this, before the next guest check-in, the iPad certificate could be reinstalled and the software integrity of the application installed verified. The old certificate could be automatically revoked from the tunneling server at time of check-out and the new certificate added at the time of check-in. This whole process could be easily automated by utilizing a configuration server (triggered after user check-out). Another method to perform this process off-line is to provide guest with the iPads at check-in. This option adds the benefit of allowing guests to inquire about the iPad functionality, or even decline its use if they are not comfortable with a multi-purpose computing device in their room.

6 Conclusion

This particular attack has important implications for large scale home automation applications, as several hotels around the world are beginning to offer this room amenity. The severity of these types of security flaws cannot be understated - from creating a chaotic atmosphere to raising room temperatures at night with fatal consequences - hoteliers need to understand the risks and liabilities they are exposed to by faulty security deployments. Security researchers, leaders in the automation market and members of the hotel industry need to start conversations to provide guest with reasonable protection standards while enjoying this new and promising technology.

References

[1] N. Zeminsky. Analysis: U.S. industrials, telecoms to face off in home automation. http://www.reuters.com/article/2012/08/23/us-usa-manufacturing-homeautomation-idUSBRE87M0U320120823.

[2] http://www.intelitycorp.com/main/press/facts.php.

[3] Why use KNX? http://www.knxuk.org.

[4] Siemens AG Dipl.-Ing. Hans-Joachim Langels. KNX IP –using IP networks as KNX medium.

[5] Interworking. KNX association.

[6] KNXnet/IP / KNX TP tunneling howto. http://www.eb-systeme.de/?page_id=479.

[7] eibd. http://www.auto.tuwien.ac.at/~mkoegler/index.php/eibd.

[8] W. Granzer, G. Neugschwandtner, and W. Kastner. EIBsec: a security extension to KNX/EIB. In *Konnex Scientific Conference*, November 2006.

[9] Christian Reinisch. Wireless communication in knx/eib. In *KNX Scientific Conference*, 2006.

[10] Daniel Lechner, Wolfgang Granzer, and Wolfgang Kastner. Security for knxnet/IP. In *Konnex Scientific Conf.*, 2008.