Defending building automation systems using decoy networks

Caleb Mays, Mason Rice, Benjamin Ramsey, John Pecarina, Barry Mullins

Chapter 15

DEFENDING BUILDING AUTOMATION SYSTEMS USING DECOY NETWORKS

Caleb Mays, Mason Rice, Benjamin Ramsey, John Pecarina and Barry Mullins

Abstract The Internet of Things (IoT) and home and building automation systems are growing fields. Many automation networks use proprietary protocols and few publications have evaluated their security. INSTEON is a leading Internet of Things protocol for home and building automation and, like other proprietary protocols, little research is available relating to its vulnerabilities. This chapter presents techniques for analyzing INSTEON traffic and defending INSTEON networks using virtual decoys. By using a software-defined radio, the packet capture rate for INSTEON traffic is increased from approximately 40% to almost 75% compared with previous research efforts. Additionally, a virtual decoy network has been designed and tested for authenticity and targetability to better protect home and building automation systems.

Keywords: Internet of Things, home and building automation, honeypots

1. Introduction

The family was in the car ready for a great weekend getaway. They had planned a three-day camping trip – fishing, hiking, s'mores – the whole experience. The father was pulling away from the home and pushed the button on his phone to close the garage door. The mother used her smart phone to verify that the external doors were locked and that the thermostat was set. The father maneuvered the family car past parked vehicles on the street. The family started what they all hoped would be a great weekend.

The weekend was great! The weather was perfect; not too hot, not too cold and no rain. When the family returned home and their car approached the driveway, everything looked normal. The garage door was down, the doors were shut, there were no broken windows and no signs of a break-in. The inside, however, was a different story. The home had been burgled. The TV and entertainment systems were gone. Jewelry was missing. Family heirlooms

had been stolen. Somebody was able to open the garage door. The family thought that their garage door could only have been opened and closed via the application on the phone. How did this happen?

Little did the family know that, as they drove away, the intruder was in one of the cars parked on the street. The burglar noticed the car-top carrier, inferring the family would be gone all weekend. The intruder used an inexpensive radio and computer to capture the wireless commands used to open and close the garage door. With a few simple keystrokes, he replayed the command to open the garage door and walked into the home.

Could the family have done anything more to secure their home? Home automation devices are, no doubt, convenient, but automation networks lack proper security. Users are unknowingly making themselves more vulnerable to intruders by using home automation products.

INSTEON is a leading protocol for home and building automation and, like other proprietary protocols, little research has been published about its vulnerabilities. This chapter presents a technique for analyzing INSTEON traffic and defending INSTEON networks using virtual decoys.

2. Background

The Internet of Things (IoT) and home and building automation are growing fields. These technologies are used to connect to smart appliances in homes and buildings from anywhere in the world. Homeowners can control their lights, ensure their front doors are locked and view thermostat settings on their mobile devices. Building managers can control access and industrial HVAC settings with a few mouse clicks. While home and building automation is a relatively new field, a report by Transparency Market Research [19] projects that the global industry will grow to $21.6 billion by 2020. Gartner [6] predicts that Internet of Things devices – for home and business use – will rise from 6 billion to 20 billion by 2020.

2.1 Automation Technologies

The home and building automation market incorporates many technologies and protocols (e.g., Wi-Fi, Bluetooth, ZigBee, Z-Wave and INSTEON). Multiple studies have compared the technical specifications and capabilities of these automation technologies [1, 8, 21], but very few researchers have evaluated the security of proprietary Internet of Things protocols. The lack of security research leaves homes and businesses vulnerable and dangerously accessible to intruders.

Wi-Fi. Wi-Fi is a popular technology that is broadly incorporated in homes and businesses. Amazon's Echo and the Nest line of products use this technology. Wi-Fi attacks such as de-authentication and rogue access point attacks are well-known and tools are incorporated in the Kali version of Linux specifically for evaluating Wi-Fi security. Wi-Fi can be secured through strong encryption

techniques. Advances in Wi-Fi security will certainly benefit Wi-Fi-enabled Internet of Things devices.

Bluetooth. Rose and Ramsey [15] have demonstrated that many vendors do not implement optional Bluetooth security features (e.g., encryption by properly pairing devices). Improper Bluetooth pairing leads to confidentiality problems. Rose and Ramsey have passively sniffed passwords for several locks. They have documented several security flaws and have developed proof-of-concept tools that open several Bluetooth-enabled locks. They also note that, when presented with improper commands (e.g., unrecognized characters), many locks fail in a non-recoverable state, resulting in denial-of-service to users.

ZigBee and Z-Wave. Hall and Ramsey [9, 12] have researched the ZigBee and Z-Wave protocols and have developed tools for securing automation networks that use these protocols. The Philips Hue brand of lights is one product that uses ZigBee. Hall and Ramsey capitalized on Z-Wave broadcast commands to enumerate Z-Wave networks. Their research demonstrates the over-availability problem and general lack of confidentiality in the Z-Wave protocol. Additionally, they discovered problems similar to those encountered in Bluetooth systems, where manufacturers did not implement encryption and other security measures in a proper manner.

INSTEON. INSTEON has been producing home and building automation devices for more than ten years and recently announced plans to integrate with Revention's point-of-sale system for building automation control [13]. A wide range of devices, including LED lights, dimmable light switches, open/close sensors and security cameras, are available. INSTEON devices enjoy general acceptance within the Internet of Things community through interoperability with the Amazon Echo and Sonos home surround sound system. Applications on Apple iOS, Google Android and Microsoft Windows 8/10 provide interfaces for monitoring and modifying the devices. Figure 1 presents a schematic diagram of a basic INSTEON network.

2.2 Honeypots for Building Automation Defense

The use of honeypots for network defense is not new. The development of traditional honeypots prior to 2006 was constrained by costly hardware. Virtual technologies have since made traditional honeypots inexpensive, convenient and prevalent. Large research honeypots have been used to capture malicious software in information technology networks [2, 11].

Using honeypots to defend industrial control and supervisory control and data acquisition (SCADA) networks is a relatively new concept. The Honeynet Project [14] has made significant progress in developing an open source honeypot for industrial control networks. Winn et al. [20] have researched industrial control system honeypots and have extended the `honeyd` program to `honeyd+` to make it more authentic. They discuss two levels of interaction displayed by

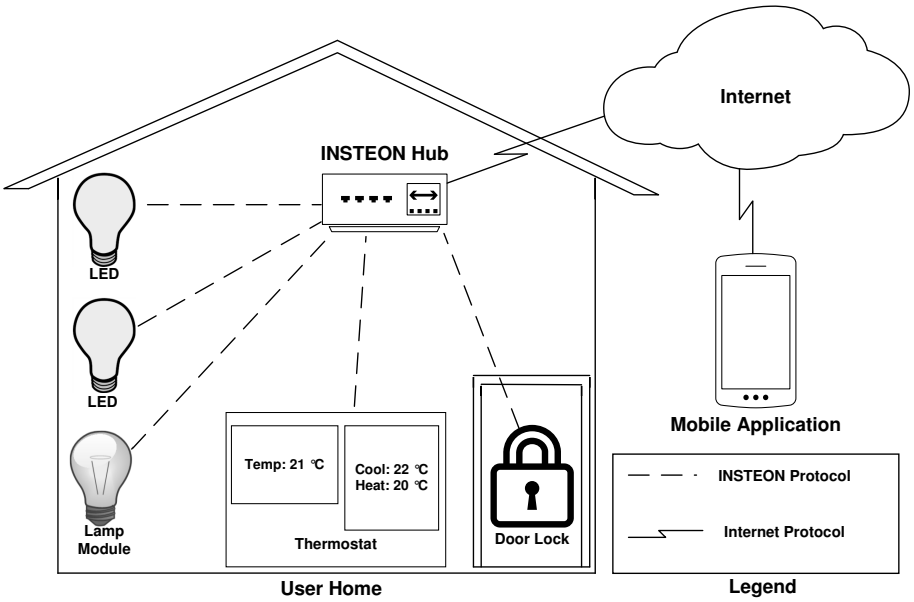

Figure 1. INSTEON network.

a honeypot: (i) high-level interaction; and (ii) low-level interaction. They also argue that the targetability of a honeypot needs to match the intended network and that the decoys must appear authentic enough for an attacker to target the fake devices instead of the real devices. Girtz et al. [7] have enhanced the performance of `honeyd+` by developing an application layer emulator.

Schneier [16] uses the terms Internet of Things devices and cyber-physical systems interchangeably. He correctly writes that the Internet of Things has given the Internet "hands and feet" with the ability to change physical systems through cyber means. Meanwhile, research in the area of Internet of Things security and industrial control system honeypots are converging. This chapter describes the design and implementation of a honeypot with authenticity and targetability characteristics that can protect an INSTEON home automation network.

3. Understanding INSTEON

There are three primary sources of information about the INSTEON protocol. The first two documents are published by INSTEON [4, 18]. The third source is a presentation and software produced by security researchers, Shipley and Gooler [17].

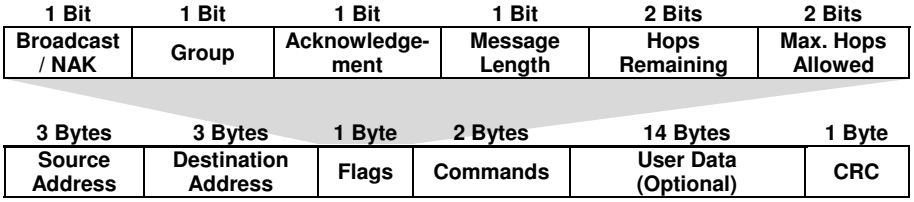

Figure 2. INSTEON packet structure specified in [18].

3.1 INSTEON Documentation

INSTEON created its own communications protocol and published many of the specifications. Two documents, *INSTEON: The Details* [18] and *INSTEON: Developer's Guide* [4], are intended to inform developers, security professionals and users about the protocol.

INSTEON: The Details. *INSTEON: The Details* [18] describes the specifications of INSTEON's wireless and power-line protocol settings, packet structure, hopping mechanism, timing scheme, network diagram and many other details. INSTEON claims to use 915 MHz as the center frequency and frequency shift keying (FSK) with a 64 KHz frequency shift as the radio frequency (RF) modulation method, and encode the signal using the Manchester scheme.

INSTEON defines its own packet structure, which is shown in Figure 2. An INSTEON packet contains five mandatory fields and one optional field. The structure starts with the source and destination device addresses. INSTEON device addresses, which are three bytes long, are set during manufacturing and remain static. The packet structure continues with one byte for flags, two bytes for commands, an optional user-data field and ends with a one-byte cyclic redundancy check (CRC).

INSTEON flags describe the type of message contained in a packet (e.g., broadcast or non-acknowledgment, group, acknowledgment, extended or standard length), the number of hops remaining and the maximum number of hops allowed for the packet. Standard-length packets are ten bytes long while extended-length packets include 14 bytes of user-defined data, making these packets 24 bytes long. INSTEON messages are not routed like typical Internet Protocol messages. Instead, packets traverse the network by hopping across devices. A packet may hop a maximum of three times as it traverses the network to its destination device.

INSTEON devices are statically networked during the setup (pairing) process. A user performs a series of button-pushes and uses the INSTEON mobile application to program the devices. One device is the controller (master) while the other device is the responder (slave). INSTEON devices can be networked to communicate in a mesh network to limit service disruptions due to malfunctioning devices. The INSTEON hub connects to the user's home router for Internet access. The specifications describe the ability to include encrypted

1 Byte	3 Bytes	3 Bytes	2 Bytes	14 Bytes	1 Byte
Flags	Destination Address	Source Address	Commands	User Data (Optional)	CRC

Figure 3. Packet structures specified in [4] and [17].

data using the user-data field of extended length messages. This capability is an add-on that may be implemented by device developers.

INSTEON: Developer's Guide. *INSTEON: Developer's Guide* [4] provides additional information about the INSTEON protocol. While most of the specifications in the document match those provided in the previous document [18], a few details about the radio frequency specifications differ. Specifically, the frequency shift keying deviation and symbol rate are listed as 200 KHz and 9.124 KBaud, respectively. Additionally, as shown in Figure 3, the INSTEON packet structure varies from the structure described in the previous document.

3.2 Previous Research

Shipley and Gooler [17] have reverse engineered the INSTEON protocol and developed a basic transmitter and receiver using the YARD Stick One software-defined radio. They also claim that INSTEON's documentation is incorrect. Specifically, they maintain that INSTEON:

- Does not use the 915 MHz center frequency, but instead 914.975 MHz (914.95 MHz is coded in the software-defined radio receiver).
- Does not use true Manchester encoding.
- Does not use "traditional" frequency shift keying, but instead inverted frequency shift keying.
- Does not use the frequency shift keying value specified in the documentation.
- Does not use the symbol rate specified in the documentation.
- Does not use the cyclic redundancy check algorithm as specified in the documentation.
- Does not use the packet structure specified in the documentation. Figure 3 presents Shipley and Gooler's claimed INSTEON packet structure.

While Shipley and Gooler's transmitter and receiver tool work, they are limited and could be improved; this was the impetus for the research described in this chapter. First, the receiver program outputs packets directly to the terminal. Second, the receiver seemingly drops packets; this chapter describes

the pilot studies and experiments used to test and improve the packet capture success rate. Third, Shipley and Gooler have not presented a method for enumerating or investigating INSTEON devices; this limitation provided the motivation to find commands that could be used to query and identify INSTEON devices beyond the device address.

3.3 Integrating Wireshark

Wireshark is the *de facto* network traffic analysis tool. Shipley and Gooler's research did not integrate with Wireshark. Instead, INSTEON network traffic was outputted to a terminal window. This made it difficult to view and analyze the packets or save traffic data.

The integration with Wireshark is a three-step problem: (i) outputting `pcap` data; (ii) developing a dissector to interpret the protocol; and (iii) configuring Wireshark to use the dissector. Over the course of this research, Shipley and Gooler's receiver was enhanced by integrating Wireshark [10].

3.4 Pilot Studies

The first pilot study described in this section captured packets between the INSTEON hub and an LED light. The results established the baseline for the packet capture success test. The second pilot study discovered and verified a command for querying a device for its characteristics.

Pilot Study 1: Determining Packet Count. This pilot study employed a similar experimental setup but a significantly more robust reception method than the one used by Shipley and Gooler. The study determined that, for every "on" command, the hub and LED exchanged eight packets. The same was true for every "off" command. Turning the light on and off 20 times produced a total of 320 packets. This established a baseline number for the packet capture success rate experiment. Additional details about this experiment are provided in Section 4.1.

Pilot Study 2: Enumerating Devices. Command tables from 2007 (see [3]) describe a command that can be used to request the identity of a device. Three messages are transmitted as illustrated in Figure 4. First, the controller device issues the command to request the identity of the responding device. The responding device transmits an acknowledgment of the command and then responds by transmitting a broadcast message. In the INSTEON protocol, broadcast messages are typically used in the device pairing process and have a unique structure. INSTEON broadcast messages contain the device category, subcategory and firmware version of the source device. Figure 5 shows the packet structure of a broadcast message.

The documentation [5] identifies device category `0x05` as corresponding to an "access control" device. Tests confirmed that the devices and their respective categories match the documentation. Thus, the device category can be used as

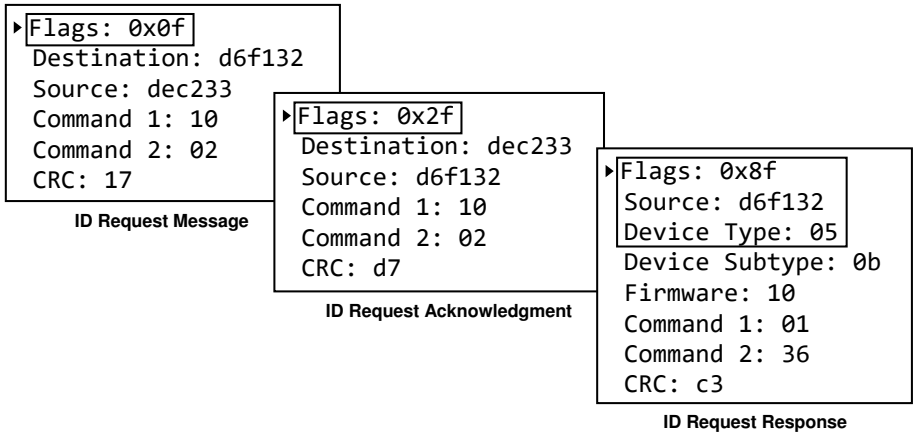

Figure 4. Device identification request, acknowledgment and response messages.

1 Byte	3 Bytes	1 Byte	1 Byte	1 Byte	2 Bytes	1 Byte
Flags	Source Address	Device Category	Device Subcategory	Firmware Version	Commands	CRC

Figure 5. Packet structure for broadcast messages.

the basis for device enumeration. Note that the device category also informs an attacker about the device that is being controlled (e.g., lighting control, climate control or access control).

3.5 INSTEON Protocol Summary

The INSTEON documentation and previous research are inconsistent about the wireless specifications and packet structure of the INSTEON protocol. As mentioned above, one document states that the center frequency is 915 MHz while Shipley and Gooler claim that the true center frequency is 914.975 MHz. Further examination of Shipley and Gooler's code reveal a third potential center frequency value of 914.95 MHz. Neither the INSTEON documentation nor previous research agree on the frequency shift keying value. Additionally, the documentation and previous research disagree on the frequency shift keying symbol rate – one document lists 76.8 KBaud, another lists 9.124 KBaud and Shipley and Gooler list 9.125 KBaud. Table 1 summarizes the discrepancies. These conflicts lead researchers to ask: Which values are correct?

4. Experiments

Two experiments were conducted. The first experiment investigated the packet capture success rate of Shipley and Gooler's receiver and attempted to improve the packet capture rate. The second experiment evaluated the authenticity and targetability of the honeypot developed during this research.

Table 1. INSTEON radio frequency specifications from all sources.

Specification	The Details	Developer's Guide	Shipley and Gooler
Center Frequency	915.000 MHz	915.000 MHz	914.975 MHz 914.950 MHz (in code)
Encoding Method	Manchester	Manchester	"Tokenized" Manchester
Modulation Method	FSK	FSK	Inverted FSK
FSK Shift	64 KHz	200 KHz	150 KHz
FSK Symbol Rate	76.800 KBaud	9.124 KBaud	9.125 KBaud

4.1 Packet Capture Experiment

The conflicting documentation summarized in Section 3.5 provided the control variables for this experiment. The experiment had two objectives. The first objective was to validate (or refute) Shipley and Gooler's claims regarding the INSTEON radio frequency protocol specifications, specifically the center frequency, frequency shift keying value and symbol rate. The second objective was to determine the best settings to obtain the maximum packet capture success rate with the YARD Stick One software-defined radio.

Two INSTEON devices were used in the experiment: (i) INSTEON hub with Internet connectivity; and (ii) INSTEON LED bulb. The INSTEON hub application on a Windows Surface 3 computer was connected via the Internet to the INSTEON hub to control the LED light. The YARD Stick One was connected to a separate laptop running Ubuntu Linux and the software-defined radio configurations. Figure 6 presents the experimental network environment.

The experiment used the pilot study results described in Section 3.4 as the baseline for the packet capture success rate. The light was turned on and off 20 times to generate 320 packet transmissions while the receiver listened for packets. Each received or dropped packet was viewed as a separate, binary trial (successful or failed reception).

The experiment produced data with a binary distribution. The 320 individual results were pooled and viewed as a single trial. The number of received packets was divided by the number of total packets transmitted to compute the mean packet capture success rate.

The experiment involved varying the frequency between the three values in Table 1 (914.95 MHz, 914.975 MHz and 915 MHz). The frequency deviation and symbol rate were held constant at the start according to Shipley and Gooler's specifications. The best frequency was used in the test for the next control variable, and so on. If there was no distinguishable difference between a given variable, then Shipley and Gooler's settings were used for the next trial. Note that the symbol rate of 76.8 KBaud was assumed to be incorrect in

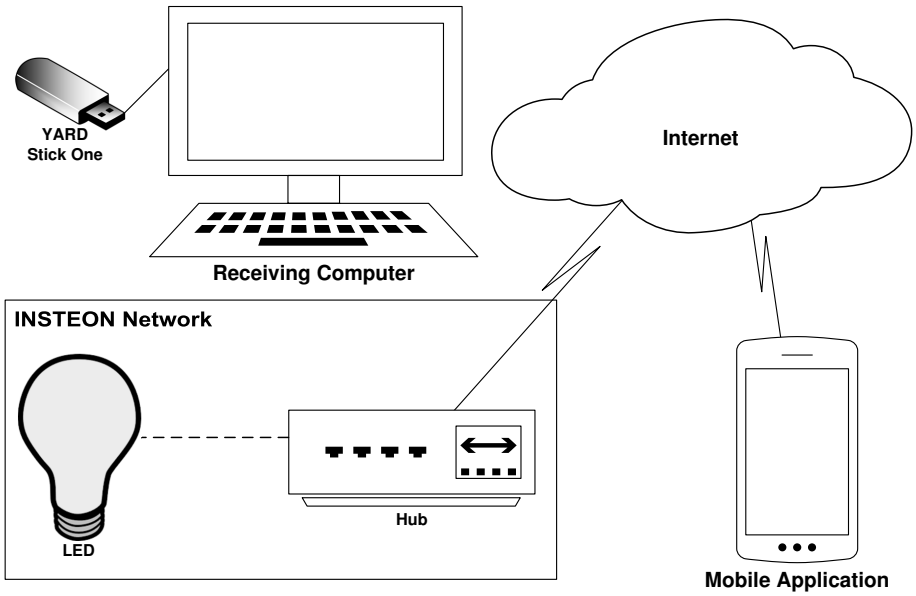

Figure 6. Experimental network environment for packet capture.

Table 2. Packet capture trial runs.

Trial	Frequency	Shift	Symbol Rate
1	914.950 MHz	150 KHz	9.125 KBaud
2	914.975 MHz	150 KHz	9.125 KBaud
3	915.000 MHz	150 KHz	9.125 KBaud
4	915.000 MHz	64 KHz	9.125 KBaud
5	915.000 MHz	200 KHz	9.125 KBaud
6	915.000 MHz	150 KHz	9.000 KBaud
7	915.000 MHz	150 KHz	9.250 KBaud

the experiment because it was very different from the values in the INSTEON documentation [4, 17]. The decision was made to vary the symbol rate between 9 KBaud, 9.125 KBaud and 9.250 KBaud. This resulted in seven radio configurations for the experiment. Table 2 shows the configuration for each experimental trial.

4.2 Functional Testing Experiment

The evaluation of the honeypot involved a functional test for authenticity and targetability. This experiment had two goals. The first goal was to ensure that a single honeypot network enumeration and map matched the genuine INSTEON device enumeration and map, enabling an attacker to believe that

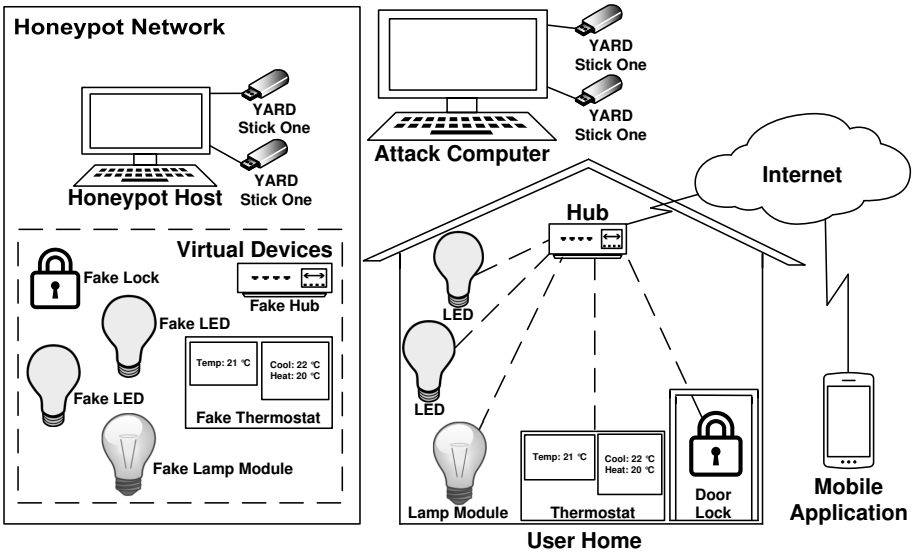

Figure 7. Experimental network environment for functional testing.

the honeypot devices were authentic. The second goal was to ensure that the honeypot network traffic mimics genuine network communications, making the honeypot targetable by an attacker. The simulated network traffic and devices were based on the results of the pilot study in Section 3.4 and other observations made over the course of the research.

The test environment consisted of an INSTEON hub, thermostat, two LED bulbs, a regular light bulb with a lamp dimmer module and a lock controller with door lock. Figure 7 illustrates the experimental environment. The honeypot was hosted on a high-performance computer platform (Dell Precision M4500 laptop running Ubuntu 14.04 LTS). A single honeypot host controlled the distinct honeypot networks.

The honeypot target had to be authentic enough so that an attacker could interact with the virtual devices. A successful implementation would involve the virtual devices responding to commands in a manner identical to genuine devices.

The authenticity portion of the experiment involved the attacker enumerating the devices using the Identity Request command described in Section 3.4. The attacker created a full network map by spoofing this command between all known devices. Figure 8 shows the results of the true INSTEON network scan. An interesting point is that the thermostat (ID: `D6 F1 32`) responded to every other INSTEON device. This is not typical of INSTEON devices and is due to a flaw in the thermostat logic. Based on this information, a decision was made to allow the honeypot thermostat to have the same "flaw."

A targetable honeypot should have multiple decoys to draw the attention of attackers. The decoys should appear to be positioned in the appropriate

```
------------------------------
D6 F1 32 is a: Climate Control
DE C2 33 is a: Network Bridge
33 D3 32 is a: Dimmable Lighting Control
56 E2 3E is a: Access Control
95 A3 2E is a: Dimmable Lighting Control
E7 5C 2F is a: Dimmable Lighting Control

Controllers are:
DE C2 33 controls: ['D6 F1 32', '33 D3 32', '56 E2 3E', '95 A3 2E', 'E7 5C 2F']
33 D3 32 controls: ['D6 F1 32']
56 E2 3E controls: ['DE C2 33', 'D6 F1 32']
95 A3 2E controls: ['DE C2 33, 'D6 F1 32']
E7 5C 2F controls: ['D6 F1 32']

Responders are:
D6 F1 32 responds to: ['DE C2 33', '33 D3 32', '95 A3 2E', '56 E2 3E', 'E7 C5 2F']
DE C2 33 responds to: ['95 A3 2E', '56 E2 3E']
33 D3 32 responds to: ['DE C2 33']
56 E2 3E responds to: ['DE C2 33']
95 A3 2E responds to: ['DE C2 33']
E7 5C 2F responds to: ['DE C2 33']
------------------------------
```

Figure 8. INSTEON network baseline enumeration.

environments. A single Internet of Things light bulb would be of little interest to an attacker when it is not connected to a larger home automation network. However, the presence of several connected devices could convince an attacker that the decoys are part of a legitimate home automation system. Indeed, an INSTEON hub, thermostat, multiple lights and lock controller would present a network that is reasonably similar to an INSTEON home network, thus posing as an attractive target to an attacker.

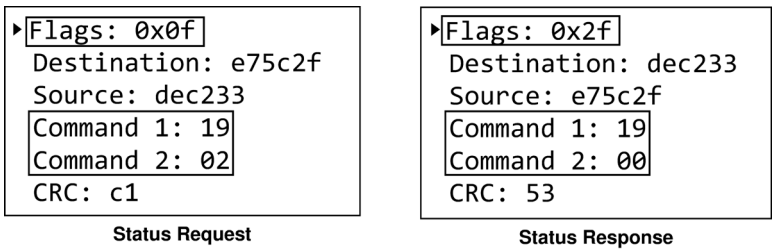

Figure 9. Wireshark output of device status request and response.

The targetability portion of the experiment involved checking for the presence of Android devices connected to the INSTEON application. When a user logs into the mobile application, the hub requests the status of all non-battery-powered devices in the INSTEON network. Figure 9 shows the Wireshark output of the traffic generated when the INSTEON hub requests the status of a device. The honeypot was intended to mimic similar network traffic, thus rendering the honeypot devices targetable by an attacker. The traffic was gen-

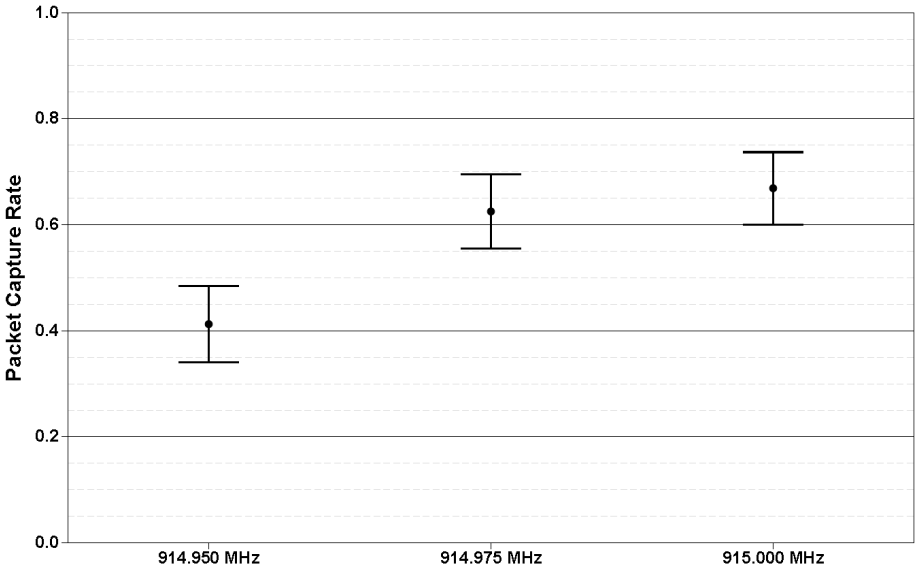

Figure 10. Packet capture results for various center frequencies.

erated using a simple Python program that made use of Shipley and Gooler's INSTEON transmission program.

The honeypot functional testing experiment involved the creation of a honeypot with one virtual network and five distinct virtual networks. If all the honeypot networks mimicked the genuine network, then the honeypot networks would be targetable and would help hide the genuine network from a would-be attacker.

5. Experimental Results

This section analyzes the results of the packet capture and honeypot functional testing experiments.

5.1 Packet Capture Experiment

A 99% confidence interval plot was generated for each pooled trial with comparison plots to present the results. Additionally, a linear model for the binary logistical distribution was produced to verify the results.

Figure 10 shows the impact of varying frequency while keeping the frequency shift keying value and symbol rate constant at 150 KHz and 9.125 KBaud, respectively. The 915 MHz frequency produced a packet capture rate of approximately 65%. The 914.95 MHz frequency, which is in the open-source code published by Shipley and Gooler, produced an inferior packet capture rate (approximately 40%) compared with the other frequencies.

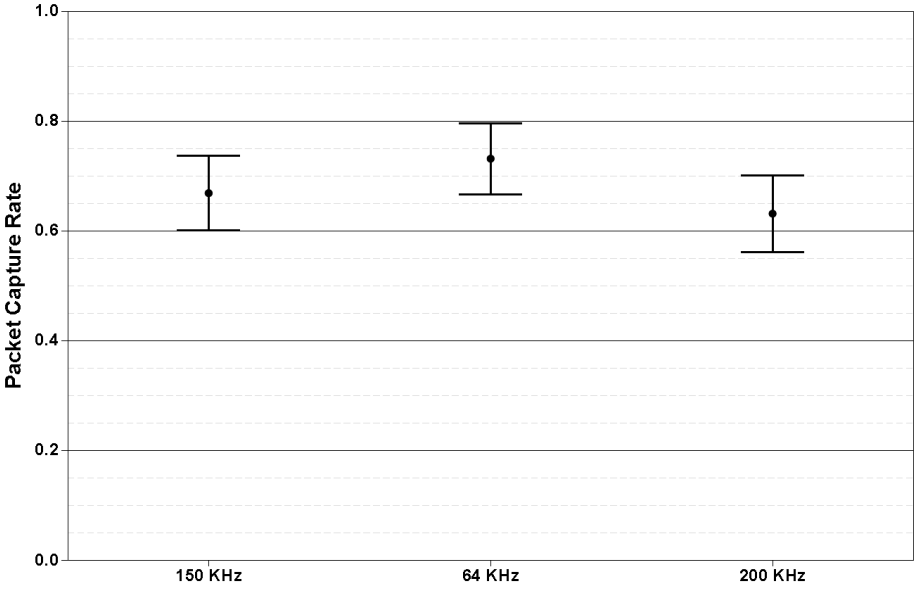

Figure 11. Packet capture results for various frequency shift keying values.

Figure 11 shows the confidence intervals when the frequency was held constant at 915 MHz and symbol rate was 9.125 KBaud, while the frequency shift keying value was varied between 200 KHz, 150 KHz and 64 KHz. No statistical difference in the packet capture rate was observed when varying the frequency shift keying value, although tuning the value to 64 KHz captured approximately 75% of the packets.

The final test varied the symbol rates between 9 KBaud, 9.125 KBaud and 9.25 KBaud while maintaining the center frequency at 915 MHz and the frequency shift keying value at 150 KHz. The confidence intervals revealed that the packet capture rates were not statistically different.

A graph containing the confidence intervals for each experimental trial was constructed and a linear model for the regression was created. Figure 12 shows the confidence interval plots for all the experimental trials. The best reception rate was obtained at approximately 75% when the center frequency was 915 MHz, frequency shift keying value was 64 KHz and symbol rate was 9.125 KBaud. The center frequency of 914.95 MHz encoded in Shipley and Gooler's software clearly yielded an inferior packet capture rate. Modifying the center frequency variable was determined to be statistically significant in improving the packet capture success rate.

Figure 13 shows the linear model. This analysis verifies that the frequency has a statistically significant impact on the packet capture rate and that no other variables are statistically significant.

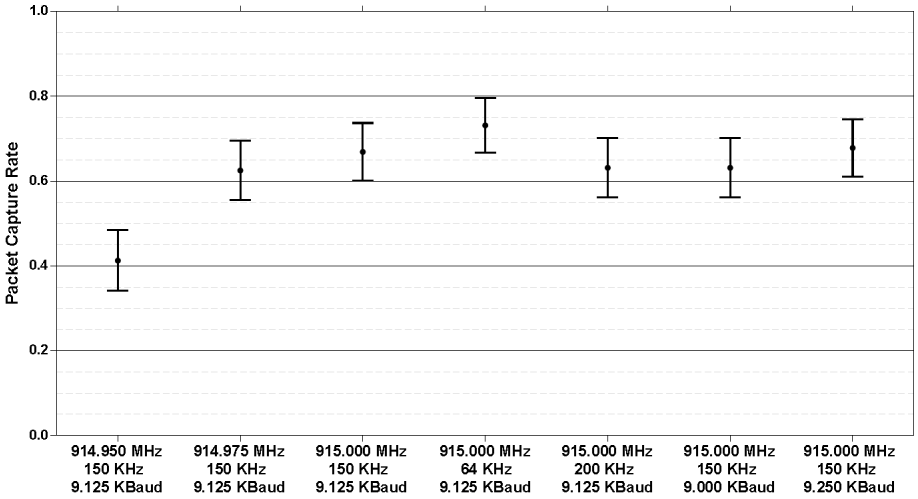

Figure 12. Results of all the experimental trials.

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-0.43267	0.31192	-1.387	0.165	
Frequency914.975 Mhz	0.86447	0.16195	5.338	9.41e-08	***
Frequency915 Mhz	1.05618	0.16432	6.427	1.30e-10	***
Shift150 Khz	-0.08592	0.16927	-1.508	0.612	
Shift200 Khz	-0.25087	0.16724	-1.500	0.134	
Shift64 Khz	0.21252	0.17449	1.218	0.223	
Symbol.Rate9125KBaud	0.16494	0.16593	0.994	0.320	
Symbol.Rate9250KBaud	0.20759	0.16656	1.246	0.213	

Figure 13. Linear model results.

5.2 Functional Testing Experiment

The honeypot functional testing experiment involved two parts. First, the authenticity of the honeypot was determined by mapping and investigating the honeypot network individually and comparing it with the baseline (genuine) INSTEON network. Next, the targetability of the honeypot was measured by determining if the honeypot devices presented themselves in a manner similar to the genuine INSTEON devices.

Figure 14 presents the network enumeration used to determine the authenticity of the honeypot. Table 3 summarizes the genuine INSTEON network enumeration compared with the honeypot enumeration. The honeypot devices match the true INSTEON devices with regard to device category information and when compared against the complete network map in Figure 8. Therefore, the honeypot network accurately mimics the genuine INSTEON network, helping convince an attacker that the virtual devices are authentic.

The targetability of the honeypot was determined by presenting distinct honeypot networks together with the genuine network. Figure 15 compares the

```
------------------------------
C1 C1 C1 is a: Climate Control
B1 B1 B1 is a: Network Bridge
E1 E1 E1 is a: Dimmable Lighting Control
F1 F1 F1 is a: Access Control
A1 A1 A1 is a: Dimmable Lighting Control
D1 D1 D1 is a: Dimmable Lighting Control

Controllers are:
B1 B1 B1 controls: ['C1 C1 C1', 'E1 E1 E1', 'F1 F1 F1', 'A1 A1 A1', 'D1 D1 D1']
E1 E1 E1 controls: ['C1 C1 C1']
F1 F1 F1 controls: ['B1 B1 B1', 'C1 C1 C1']
A1 A1 A1 controls: ['B1 B1 B1, 'C1 C1 C1']
D1 D1 D1 controls: ['C1 C1 C1']

Responders are:
C1 C1 C1 responds to: ['B1 B1 B1', 'E1 E1 E1', 'A1 A1 A1', 'F1 F1 F1', 'D1 D1 D1']
B1 B1 B1 responds to: ['A1 A1 A1', 'F1 F1 F1']
E1 E1 E1 responds to: ['B1 B1 B1']
F1 F1 F1 responds to: ['B1 B1 B1']
A1 A1 A1 responds to: ['B1 B1 B1']
D1 D1 D1 responds to: ['B1 B1 B1']
------------------------------
```

Figure 14. Honeypot baseline scan.

Table 3. Genuine INSTEON network compared with one honeypot network.

INSTEON ID	Honeypot ID	Device Category	Matched
DE C2 33	B1 B1 B1	Network bridge	✓
D6 F1 32	C1 C1 C1	Climate control	✓
E7 5C 2F	D1 D1 D1	Dimmable light control	✓
33 D3 32	E1 E1 E1	Dimmable light control	✓
95 A3 2E	A1 A1 A1	Dimmable light control	✓
56 E2 3E	F1 F1 F1	Access control	✓

two networks – one genuine INSTEON network and one honeypot network – running simultaneously. The honeypot generates network traffic that mimics a user checking the status of the lights, thermostat and other simulated devices. Therefore, the simulated traffic presents a targetable, distinct honeypot network to an attacker. The networks appear to be identical, thus deceiving the attacker that multiple distinct INSTEON networks are present.

Figure 16 shows six networks – one genuine INSTEON network and five honeypot networks – running simultaneously. In each case, the simulated network traffic presented by the honeypot devices is functionally identical to the traffic presented by the genuine INSTEON network devices. This helps "hide" the genuine network among the honeypot networks. Instead of a single genuine network, an attacker is presented with six distinct, targetable networks. This

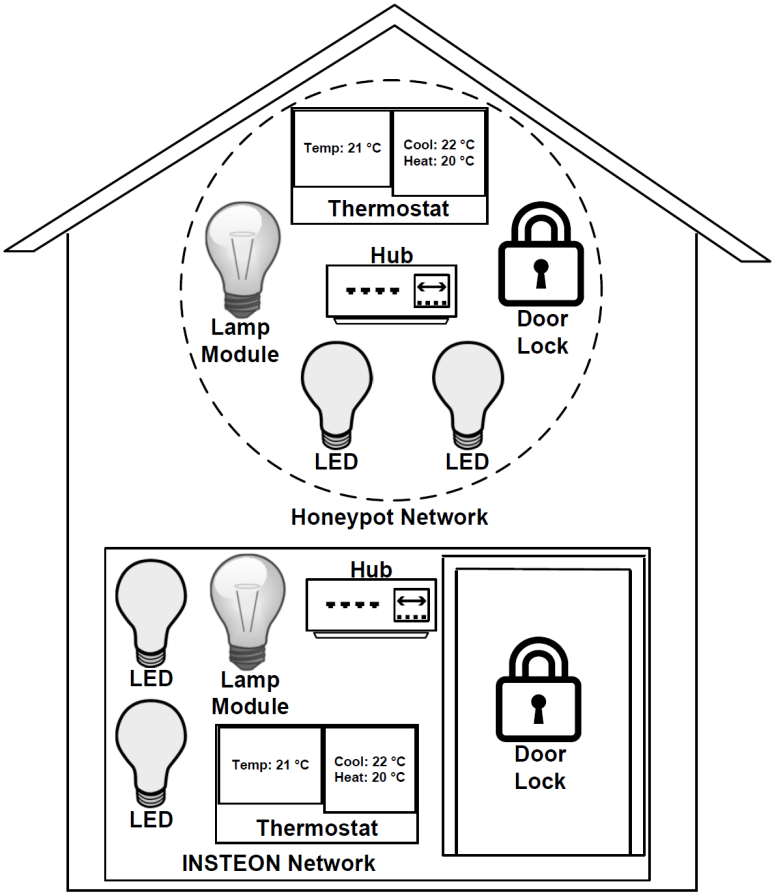

Figure 15. Genuine INSTEON network and a honeypot network.

protects the genuine network by reducing its attack probability to 1/6, unless the attacker performs deeper analysis before targeting the networks.

6. Limitations and Future Work

The honeypot developed in this research is limited in the types of devices it can replicate. Currently, the honeypot is programmed to mimic dimmable lights, access controllers, hubs and thermostats. A typical user would have these devices, but may have other types of devices as well. Other device types include switched lighting controls (e.g., in-wall light switches and dimmers) and security and safety sensors (e.g., door open/close, motion and leak sensors). These capabilities need to be investigated and incorporated in the honeypot.

Additionally, the performance of the honeypot was not measured in the experiments. An experiment that measures packet response times could be

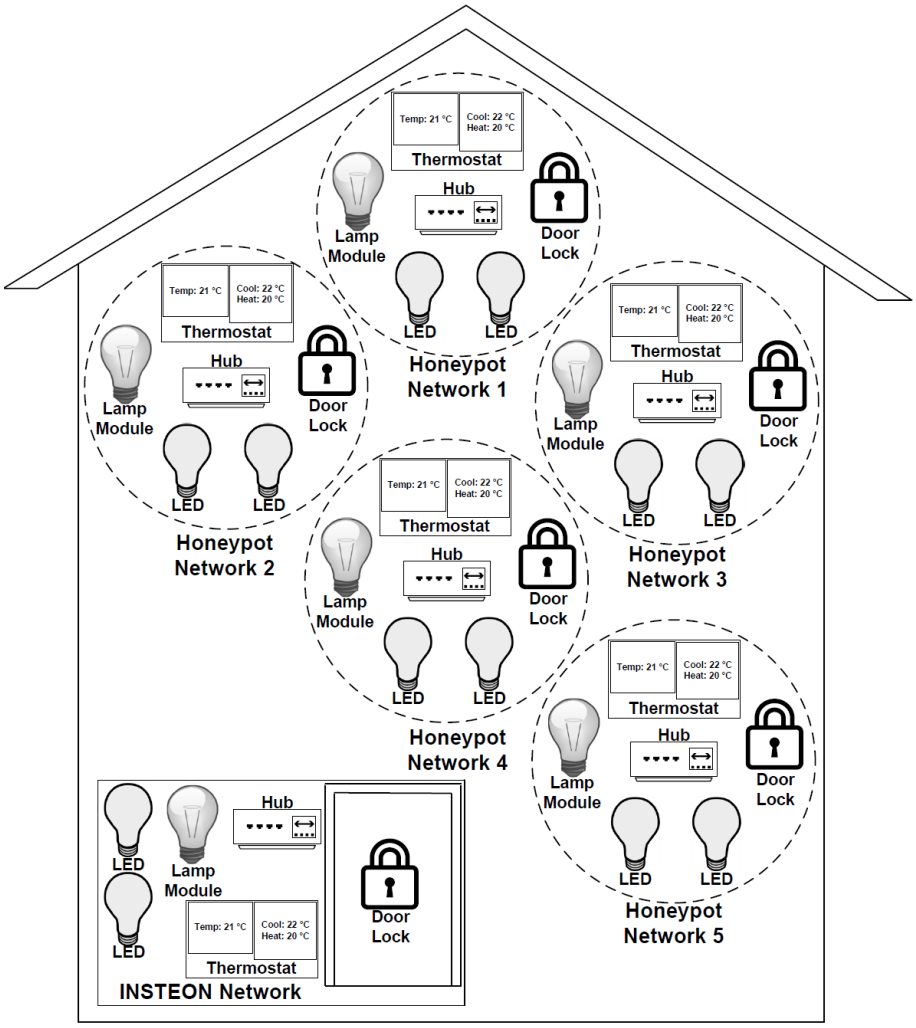

Figure 16. Genuine INSTEON network and five honeypot networks.

performed to identify the number of honeypot devices that could be instantiated before the honeypot experiences noticeable delays.

In its current state, the honeypot developed in this research is purely a decoy. Additional reporting and logging capabilities must be implemented before the honeypot can serve as a robust defensive tool. A relatively simple reporting capability could be implemented by installing an email server in the honeypot host that sends email messages to the user when events of interest occur.

7. Conclusions

Security and safety are becoming priorities as home and building automation technologies and Internet of Things devices proliferate. Device developers and manufacturers need to incorporate sound security engineering principles throughout the design and implementation phases of home and building automation systems. INSTEON is a leading Internet of Things protocol for home and building automation. The proposed technique for analyzing INSTEON traffic using a YARD Stick One software-defined radio improves the packet capture rate from approximately 40% to almost 75% compared with previous efforts. Additionally, the virtual INSTEON decoy networks developed in this research have excellent authenticity and targetability characteristics, which renders them attractive candidates for helping secure home and building automation systems as well as Internet of Things devices in general.

Note that the views expressed in this chapter are those of the authors and do not reflect the official policy or position of the U.S. Air Force, U.S. Army, U.S. Department of Defense or U.S. Government.

Acknowledgement

This research was partially supported by the U.S. Department of Homeland Security Industrial Control Systems Cyber Emergency Response Team (ICS-CERT).

References

[1] A. Baviskar, J. Baviskar, S. Wagh, A. Mulla and P. Dave, Comparative study of communication technologies for power-optimized automation systems: A review and implementation, *Proceedings of the Fifth International Conference on Communication Systems and Network Technologies*, pp. 375–380, 2015.

[2] M. Bowden, *Worm: The First Digital World War*, Atlantic Monthly Press, New York, 2011.

[3] P. Darbee, INSTEON Command Tables, Revision 20070927a, SmartLabs Technology, Irvine, California (`cache.insteon.com/pdf/INSTEON_Command_Tables_20070925a.pdf`), 2007.

[4] P. Darbee, INSTEON Developer's Guide (2nd Edition), SmartLabs Technology, Irvine, California (`cache.insteon.com/pdf/INSTEON_Developers_Guide_20070816a.pdf`), 2007.

[5] P. Darbee, INSTEON Device Categories and Product Keys, Revision 20081008, SmartLabs Technology, Irvine, California (`cache.insteon.com/pdf/INSTEON_DevCats_and_Product_Keys_20081008.pdf`), 2008.

[6] Gartner, Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015, Stamford, Connecticut (`gartner.com/newsroom/id/3165317`), November 10, 2015.

[7] K. Girtz, B. Mullins, M. Rice and J. Lopez, Practical application layer emulation in industrial control system honeypots, in *Critical Infrastructure Protection X*, M. Rice and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 83–98, 2016.

[8] C. Gomez and J. Paradells, Wireless home automation networks: A survey of architectures and technologies, *IEEE Communications*, vol. 48(6), pp. 92–101, 2010.

[9] J. Hall, B. Ramsey, M. Rice and T. Lacey, Z-wave network reconnaissance and transceiver fingerprinting using software-defined radios, *Proceedings of the Eleventh International Conference on Cyber Warfare and Security*, pp. 163–171, 2016.

[10] C. Mays and B. Ramsey, INSTEON, inste-off, inste-open? presented at *DEF CON 24*, 2016.

[11] N. Provos and T. Holz, *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*, Addison-Wesley, Boston, Massachusetts, 2007.

[12] B. Ramsey, Improved Wireless Security through Physical Layer Protocol Manipulation and Radio Frequency Fingerprinting, Ph.D. Dissertation, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2014.

[13] Revention, Revention integrates with Insteon to provide automated light fixtures, Houston, Texas (`revention.com/news/articles/Revention-Integrates-with-Insteon-to-Provide-Automated-Light-Fixtures`), November 29, 2016.

[14] L. Rist, J. Vestergaard, D. Haslinger, A. Pasquale and J. Smith, Conpot ICS/SCADA Honeypot, Honeynet Project (`conpot.org`), 2013.

[15] A. Rose and B. Ramsey, Picking Bluetooth Low Energy locks from a quarter mile away, presented at *DEF CON 24*, 2016.

[16] B. Schneier, Real-world security and the Internet of Things, *Schneier on Security* (`schneier.com/blog/archives/2016/07/real-world_secu.html`), July 28, 2016.

[17] P. Shipley and R. Gooler, Insteon: False security and deceptive documentation, presented at *DEF CON 23*, 2015.

[18] SmartLabs Technology, INSTEON Whitepaper: The Details, Version 2.0, Irvine, California (`cache.insteon.com/documentation/insteon_details.pdf`), 2013.

[19] Transparency Market Research, Global home automation market will be worth US$21.6 billion by 2020, Albany, New York (`www.transparencymarketresearch.com/pressrelease/home-automation-market.htm`), September 3, 2015.

[20] M. Winn, M. Rice, S. Dunlap, J. Lopez and B. Mullins, Constructing cost-effective and targetable industrial control system honeypots for production networks, *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 47–58, 2015.

[21] C. Withanage, R. Ashok, C. Yuen and K. Otto, A comparison of the popular home automation technologies, *Proceedings of the IEEE Conference on Innovative Smart Grid Technologies – Asia*, pp. 600–605, 2014.