Detecting Malicious Activities in ZigBee Networks using Cognitive Radio

Tulin Mangir, Lelass Sarakbi, Harvy Younan

Electrical Engineering Department California State University- Long Beach, CA. USA., CA 90840

temangir@csulb.edu; lsarakbi@student.csulb.edu; Harvy.Younan@student.csulb.edu

Abstract

ZigBee is a low power, low data wireless protocol that is used for low rate short range (PAN type) networks due to simplicity and ease of use. Different attempts have been proposed to secure the ZigBee networks by proposing a stack modification. Our approach, however, does not require any changes to the protocol stack or to the ZigBee device, and therefore has potential for further application. This approach, as summarized in this paper, is by detecting malicious activities in ZigBee network and denying access.

To test the approach we have built a testbed and have experimented with the most likely scenarios of attack for ZigBee sensor networks. It was discovered that the Network Discovery process is the key in all attack scenarios. The idea of adding an intrusion detection system eliminates the threats that are presented by the stack pitfalls and requires no particular change to the existing stack or network. It was observed that the system has successfully secured this process without interrupting or adding traffic to the ZigBee network.

Keywords

ZigBee, Security, Cognitive Radio, Intrusion Detection.

1. Introduction

ZigBee is a low power, low data rate wireless protocol that is used for low rate sensor network applications. ZigBee has achieved a widespread market in home automation industry and it is used in home control systems (such as lighting controls and security system), smart grid technologies (advanced metering infrastructure, and automation system industries (healthcare, industries, and transportation) [1], [2], [3].

Unlike Wi-Fi networks, Zigbee networks are control based and can control a physical device and can instantly change the state of the device. Using Zigbee, for example, we can control the door locks, change the temperature of a thermostat, turn fire alarms on and off, and control the energy consumption of each device.
The security issues that ZigBee suffer from are not something newly introduced, but inherent in the protocol stack. However, its widespread use in smart grid networks and home area networks (HANs) exert a serious threat. Attackers can simply redirect the traffic, dump the network or inject some data in the packets in order to alter the network. This can result in serious disaster, if an attacker locked all the doors in case of a fire, or alters the data being sent back and forth to the smart grid.

DOI : 10.5121/ijdps.2011.2605

2. Related Work

There are several security issues that are represented in the ZigBee Stack. The first attempt to identify these security issues was presented in paper [4]. The author in [4] developed an attack tool kit that addresses the security vulnerabilities and shows how a ZigBee network could be hacked. The author has showed that an Attacker can take advantage of these threats and redirect the traffic, dump the network, or inject some data in the packets in order to alter the state of the network and can even extract the network key from the captured files. For example, an attacker can lock the doors in case of a fire, get unauthorized access or alter the data being sent to the smart meter.
The security structure of ZigBee has been discussed in many other papers. In [5], the authors reviewed the security model of the ZigBee smart energy stack, and showed that the application of this model is limited due to the insufficient key and certification management processes.
In [6] different security problems have been studied such as encryption durability, conflicts in address assignment, and channel interference .The authors suggest some improvement in ZigBee stack to overcome the security problems for example they suggest modifying the routing algorithm and to add security suite (ECMQV).

In [7], the author provides an analytical study of the security structure of the MAC, network, and application layers. Security issues discussed earlier are addressed at the application layer of the ZigBee stack, which makes the PHY and MAC layer vulnerable. Effectively, if you use sensor-based networks, and an adversary is able to steal a device, they can extract key information from the hardware which can be used to exploit the rest of the network.

Our approach aims to particularly focus on securing the PHY and MAC layer. In addition, unlike most of the previous papers, which suggest a stack modification, our approach, suggests no particular change in the stack. In fact, the ZigBee network is secured by an intrusion detection system that operates using cognitive radio that could be used to secure existing and new ZigBee network [8], [9], [10], [11], [12] that monitors the activity on the network, and detects if there is any malicious activity that needs to be prevented.
The rest of the paper is organized as follows: In the Section II of this paper, our approach of using the cognitive radio and its benefits are discussed in detail. In Section III, the methodology and attack scenarios we have used are presented. Preliminary analysis of the experimental results is also included. Section IV represents the discussion about the obtained results. Section V explains the intrusion detection implementation and techniques. Finally, the paper is ended with a conclusion of the work presented.

3. Possible Solutions

3.1 Solution for Network Discovery and Location Tracking Attack

As mentioned earlier, the hacking tool that is used to discover the ZigBee network uses the same mechanism that is used by ZigBee devices [4]. There is no solution for this attack since the network discovery process can't be disabled by any means as it is part of the ZigBee mechanism. However, it is helpful to understand the impact of this attack and evaluate the ZigBee network accordingly.

3.2 Solution for Packet and Key Sniffing

Packet sniffing or eavesdropping in general is one of the well known attacks. The only mechanism that is used to avoid such attack is the CCM* integrity algorithm that provides encryption for the data being transmitted [4]. So, in order to avoid this attack, the network administrator should ensure that a strong key is selected to avoid data leakage.

However, to avoid key sniffing, the key could be pre-installed on the ZigBee device to avoid key transportation over the air. This method has its own disadvantages as mentioned earlier, which is the key is hard to change and the devices are required to keep track and process a list of authenticated list which required more NVRAM and RAM [4].

3.3 Solution for Replay Attack

In order to avoid the replay attack, the ZigBee stack should be able to identify the frames by a sequence number and make sure that the received number is greater than that previously received frame [4]. However, ZigBee stack has only 8 bits of network sequence numbers in which an attacker can take advantage of and retransmit the frame after waiting 255 frames [4].

3.4 Benefits of Using Cognitive Radio

The advantage of using cognitive radio is its ability to use software in processing the signal, which results in higher cycle rates when compiling. In addition, it is easier and more efficient to program the modulation and the demodulation of the signal while during experiments. There is no need for additional hardware circuitry to be developed. So using CR as a spectrum analyzer gives us the ability to scan the spectrum and analyze received ZigBee packets passively (no load is placed on the actual ZigBee network).

It was proven in paper [14] that the USRP had the least drop of packets when scanning two channels simultaneously, which is due to the interface speed used in cognitive radio. Thus, the intrusion detection system has the capability to scan and secure two channels simultaneously. The attacker can't detect the presence of the intrusion detection system in the network. In addition, the programming capability of the CR makes it easy to add new rules or modify existing ones to have a more secure intrusion detection system.

4. Methodology

Our approach is based on building a test bed and using Cognitive Radio (CR) that is set to operate on and secure the PHY and MAC layers for ZigBee networks. Several critical attacks chosen from [4] are performed and analyzed using cognitive radio. The attacks are selected based on the overall risk rating that takes into consideration popularity, simplicity, and impact of the attack. Intrusion detection system algorithm and rules are set based on our analysis of the attacks and results.

The Test-bed that is used to perform these experiments includes a computer that is running the ZigBee Packet Capture [13], [14], [15] connected to USRP, a laptop that is running KillerBee, and the CC2430 ZigBee evaluation kit that is used to build the ZigBee network.

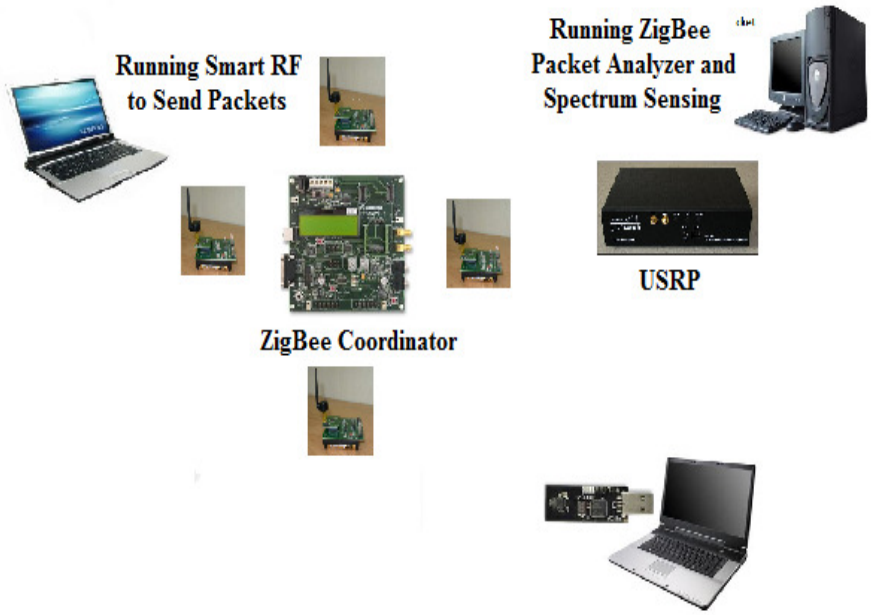

Figure 1. Experiment Testbed

We have performed different attack scenarios and monitored the results with CR for further analysis. The attacks are performed by using readily available and accessible tools to mimic the popular intrusion attempts. The first attack is performed using the "Zbstumbler" tool, which is a ZigBee discovery tool used to detect IEEE 802.15.4 networks [4]. The tool displays the Channel, PAN ID, extended PAN ID, and the source address. The first attack is used as a base experiment for further attacks. The second attack scenario is performed using the Zbfind tool that is used to track the location of IEEE 802.15.4 transmitter by measuring the RSSI [4]. The display shows how far the ZigBee node is and displays all other information that the Zbstumbler does. The third attack is the "Zbreplay" in which a dump file, that was previously captured, is replayed in order to force a specific device to perform a specific task or flood the network. The fourth attack is the "Zbsniff" which is used to analyze a packet capture file that was previously captured, and examines the configuration of APS frames for the Key-Transport command. Once the network key is found, it is then decrypted using Wireshark or any other software package.

4.1 Attack Scenario I

This attack is considered as a base experiment in which further attacks are based on it. The attack works on mimicking the network discovery process that is used by ZigBee. After establishing the ZigBee network, we have set the cognitive radio to listen to the ZigBee network. The results presented in the figure below show that the cognitive radio has identified three ZigBee nodes operating on channel 16 (2.43 GHz).

```
Cognitive@cognitive-desktop:-$ ./cc2420_rxtest.py -c2430000000
Cerdic_freq = 2.43G
Date_rate = 2M
samples_per_symbol = 2
usrp_decin = 16
fs = 4M
Using RX d'board a: Flex 2400 Rx MIMO 8
>>>gr_fir fff: using SSE
882_15_4_pkt: waiting for packet
recieved packet
checksun: 15887, recieved 15887
ok = True pktno = 760  len(payload) = 10 1/1
   payload: ['0x3', '0x8', '0x5', '0xff', '0xff', '0xff', '0xff', '0x7', '0x3d']
.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 54623, recieved 54623
ok = True pktno = 128  len(payload) = 24 2/2
   payload: ['0x0', '0x80', '0xc6', '0x0', '0x12','0x7b', '0x28', '0xff', '0x8f',
 '0x8', '0x0', '0x0', '0x21', '0x8c', '0x10', '0x6', '0x8', '0x0', '0x0',
'0x4b', '0x12', '0x8', 0x5f', '0xd5']

.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 29857, recieved 29857
ok = True pktno = 128  len(payload) = 24 3/3
    payload: ['0x0', '0x80', '0x5b', '0x0', '0x12','0x1', '0x0', '0xff', '0x8f',
 '0x8', '0x0', '0x0', '0x21', '0x8c', '0x10', '0x6', '0x8', '0x0', '0x0',
'8x4b', '8x12', '0x8', 0xal', '8x74']

.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 35823, recieved 35823
ok = True pktno = 128  len(payload) = 24 4/4
  payload: ['0x0', '8x80', '0x58', '0x0', '0x12','0xb8', '0x3c', '0xff', '0x8f',
 '0x8', '0x0', '0x0', '0x21', '0x8c', '0x10', '0x6', '0x8', '0x0', '0x0',
'0x4b', '8x12', '0x8', 0xef', '0x8b']
```

Figure 2. Monitoring Zbstumbler by Cognitive Radio

As shown in Figure1, four packets are captured by cognitive radio. As expected the results that are presented by KillerBee program matches the results of the cognitive radio captured. The first received packet is the beacon request which was transmitted by the KillerBee program, the three remaining packets are the beacon frame responses which were sent by the ZigBee coordinator and the two ZigBee routers of the tested network. The cognitive radio has successfully captured these packets that were sent over the ZigBee network and they were saved in a file for further analysis.

4.2 Attack Scenario II

Based on the information that was gathered in Zbstumbler, a Zbfind attack was performed to locate ZigBee devices. The Zbfind operates in active mode by sending Beacon Request frames and displaying the responses from ZigBee routers and coordinators. Cognitive radio was used to capture the packets. The results shown in the figure below show that the cognitive radio has identified one ZigBee nods that replied to the beacon request frame sent multiple times in order to calculate the RSSI.

```
Cognitive@cognitive-desktop:~$ ./cc2420_rxtest.py -c2430000000
Cerdic_freq = 2.43G
Date_rate = 2M
samples_per_symbol = 2
usrp_decin = 16
fs = 4M
Using RX d'board a: Flex 2480 Rx MIMO B
>>>gr_fir fff: using SSE
802_15_4_pkt: waiting for packet
recieved packet
checksun: 10552, recieved 10552
ok = True pktno = 776  len(payload) = 10 1/1
   payload: ['0x3', '0x8', '0x0', '0xff', '0xff', '8xff', '0xff', '0x7', '0x38', '0x29']
.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 11490, recieved 11498
ok = True pktno = 128  len(payload) = 24 2/2
   payload: ['0x0', '0x80', '0xee', '0x0', '0x12','0x7b', '0x28', '0xff', '0x8f', '0x0',
'0x0', '0x0', '0x21', '0x8c', '0x10', '0x6', '0x8', '0x0', '0x0',
'0x4b', '0x12', '0x0', 0xe2', '0x2c']

.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 10552, recieved 10552
ok = True pktno = 776  len(payload) = 10 3/3
   payload: ['0x3', '0x8', '8x0', '0xff', '0xff', '0xff', '0xff', '0x7', '0x38', '0x29']

.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 10552, recieved 10552
ok = True pktno = 776  len(payload) = 10 3/3
    payload: ['0x0', '0x80', '0xef', '0x0', '0x12','0x7b', '0x28', '0xff', '0x8f', '0x0',
'0x0', '0x0', '0x21', '0x8c', '0x10', '0x6', '0x8', '0x0', '0x0',
'0x4b', '0x12', '0x0', 0xb2', '0xf3']
```

Figure 3. Monitoring Zbfind by Cognitive Radio

According to the mechanism that the Zbfind uses, and as expected, the beacon request messages are sent multiple times to the target router or coordinator waiting for a response. The distance is measured according to the RSSI value of the received frame from the target device.

4.3 Attack Scenario III

The Zbreplay attack is an attack that reads packets from a packet captured file and retransmits the frames in the network. Cognitive radio is used to capture those packets that are transmitted and retransmitted. Figure 4 shows that cognitive radio showed no difference could be detected between the original and the retransmitting packets.

```
Cognitive@cognitive-desktop:~$ ./cc2420_rxtest.py -c2430000000
Cerdic_freq = 2.43G
Date_rate = 2M
samples_per_symbol = 2
usrp_decin = 16
fs = 4M
Using RX d'board a: Flex 2480 Rx MIMO B
>>>gr_fir fff: using SSE
802_15_4_pkt: waiting for packet
recieved packet
checksun: 0, recieved 0
ok = True pktno = 18398 len(payload) = 36 1/1
   payload: ['0x47', '0xde', '0xb3', '0x12', '0x4d', '0xc8, '0x43', '0xbb', '0x8b',
'0xa6', '0x1f', '0x3', '0x5a', '0x7d', '0x9', '0x38', '0x25', '0x1f',
'0x5d', '0xd4', '0xcb', '0xfc', '0x96', '0xf5', '0x45', '0x3b', '0x13', '0xd', '0x89',
'0xa', '0x0', '0x0', '0xae', '0xba', '0x0', '0x0']
.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 0, recieved 0
ok = True pktno = 18398 len(payload) = 36 2/2
   payload: ['0x47', '0xde', '0xb3', '0x12', '0x4d', '0xc8, '0x43', '0xbb', '0x8b',
'0xa6', '0x1f', '0x3', '0x5a', '0x7d', '0x9', '0x38', '0x25', '0x1f',
'0x5d', '0xd4', '0xcb', '0xfc', '0x96', '0xf5', '0x45', '0x3b', '0x13', '0xd', '0x89',
'0xa', '0x0', '0x1', '0x63', '0xab', '0x0', '0x0']
.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 0, recieved 0
ok = True pktno = 18398 len(payload) = 36 3/3
   payload: ['0x47', '0xde', '0xb3', '0x12', '0x4d', '0xc8, '0x43', '0xbb', '0x8b',
'0xa6', '0x1f', '0x3', '0x5a', '0x7d', '0x9', '0x38', '0x25', '0x1f',
'0x5d', '0xd4', '0xcb', '0xfc', '0x96', '0xf5', '0x45', '0x3b', '0x13', '0xd', '0x89',
'0xa', '0x0', '0x2', '0xf8', '0x99', '0x0', '0x0']
.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 0, recieved 0
ok = True pktno = 18398 len(payload) = 36 4/4
   payload: ['0x47', '0xde', '0xb3', '0x12', '0x4d', '0xc8, '0x43', '0xbb', '0x8b',
'0xa6', '0x1f', '0x3', '0x5a', '0x7d', '0x9', '0x38', '0x1f',
'0x5d', '0xd4', '0xcb', '0xfc', '0x96', '0xf5', '0x45', '0x3b', '0x13', '0xd', '0x89',
'0xa', '0x0', '0x3', '0x71', '0x88', '0x0', '0x0']
```

Figure 4 (a). Monitoring Zbreplay by Cognitive Radio (Data Transmitted)

```
Cognitive@cognitive-desktop:~$ ./cc2420_rxtest.py -c2430000000
Cerdic_freq = 2.43G
Date_rate = 2M
samples_per_symbol = 2
usrp_decin = 16
fs = 4M
Using RX d'board a: Flex 2480 Rx MIMO B
>>>gr_fir fff: using SSE
802_15_4_pkt: waiting for packet
recieved packet
checksun: 0, recieved 0
ok = True pktno = 18398 len(payload) = 36 1/1
   payload: ['0x47', '0xde', '0xb3', '0x12', '0x4d', '0xc8, '0x43', '0xbb', '0x8b',
'0xa6', '0x1f', '0x3', '0x5a', '0x7d', '0x9', '0x38', '0x25', '0x1f',
'0x5d', '0xd4', '0xcb', '0xfc', '0x96', '0xf5', '0x45', '0x3b', '0x13', '0xd', '0x89',
'0xa', '0x0', '0x0', '0xae', '0xba', '0x0', '0x0']
.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 0, recieved 0
ok = True pktno = 18398 len(payload) = 36 2/2
   payload: ['0x47', '0xde', '0xb3', '0x12', '0x4d', '0xc8, '0x43', '0xbb', '0x8b',
'0xa6', '0x1f', '0x3', '0x5a', '0x7d', '0x9', '0x38', '0x25', '0x1f',
'0x5d', '0xd4', '0xcb', '0xfc', '0x96', '0xf5', '0x45', '0x3b', '0x13', '0xd', '0x89',
'0xa', '0x0', '0x1', '0x63', '0xab', '0x0', '0x0']
.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 0, recieved 0
ok = True pktno = 18398 len(payload) = 36 3/3
   payload: ['0x47', '0xde', '0xb3', '0x12', '0x4d', '0xc8, '0x43', '0xbb', '0x8b',
'0xa6', '0x1f', '0x3', '0x5a', '0x7d', '0x9', '0x38', '0x25', '0x1f',
'0x5d', '0xd4', '0xcb', '0xfc', '0x96', '0xf5', '0x45', '0x3b', '0x13', '0xd', '0x89',
'0xa', '0x0', '0x2', '0xf8', '0x99', '0x0', '0x0']
.......................

802_15_4_pkt: waiting for packet
recieved packet
checksun: 0, recieved 0
ok = True pktno = 18398 len(payload) = 36 4/4
   payload: ['0x47', '0xde', '0xb3', '0x12', '0x4d', '0xc8, '0x43', '0xbb', '0x8b',
'0xa6', '0x1f', '0x3', '0x5a', '0x7d', '0x9', '0x38', '0x1f',
'0x5d', '0xd4', '0xcb', '0xfc', '0x96', '0xf5', '0x45', '0x3b', '0x13', '0xd', '0x89',
'0xa', '0x0', '0x3', '0x71', '0x88', '0x0', '0x0']
```

Figure 4 (b). Monitoring Zbreplay by Cognitive Radio (Data Retransmitted)

4.4 Attack Scenario IV

The "Zbsniff" attack is targeted to discover the network address by analyzing a previously captured file. Once the network address is detected, the network confidentiality, authentication, and integrity are threatened. The network key is decrypted using Wireshark and it is available for the attacker to be part of the network. However, this attack depends on the network discovery

process that was performed in the first attack. If the attacker succeeded in sniffing the network information using Zbstumbler then leveraging to the network and being part of it becomes an easy task using Zbsniff .

5. Discussion

After performing these experiments we have observed that the common point of vulnerability in all of the attacks is the network discovery process. The beacon frame request of the network discovery process is identified by the following ten fields (0x03 0x08 0x-- 0xff 0xff 0xff 0xff 0x07 0x-- 0x--). These fields are inserted into Wireshark for better analysis as shown in Figure 5. The three fields represented by 0x-- are the sequence number and the following two are the frame check sum fields respectively. These fields could be any numbers. As shown in Figure 1 and Figure 3, the Zbstumbler and Zbfind and Zbsniff attacks started with this frame format to request the network information about the existing networks on a specific channel. However, only ZigBee routers and coordinators can respond to these requests with a different beacon frame that has 24 fields, as described in Figure 6. In Zbfind, even though the channel number should be identified first by Zbstumbler, the attacker can still send the beacon requests several times to determine how far the target device is, based on the RSSI feature of the Zbfind.

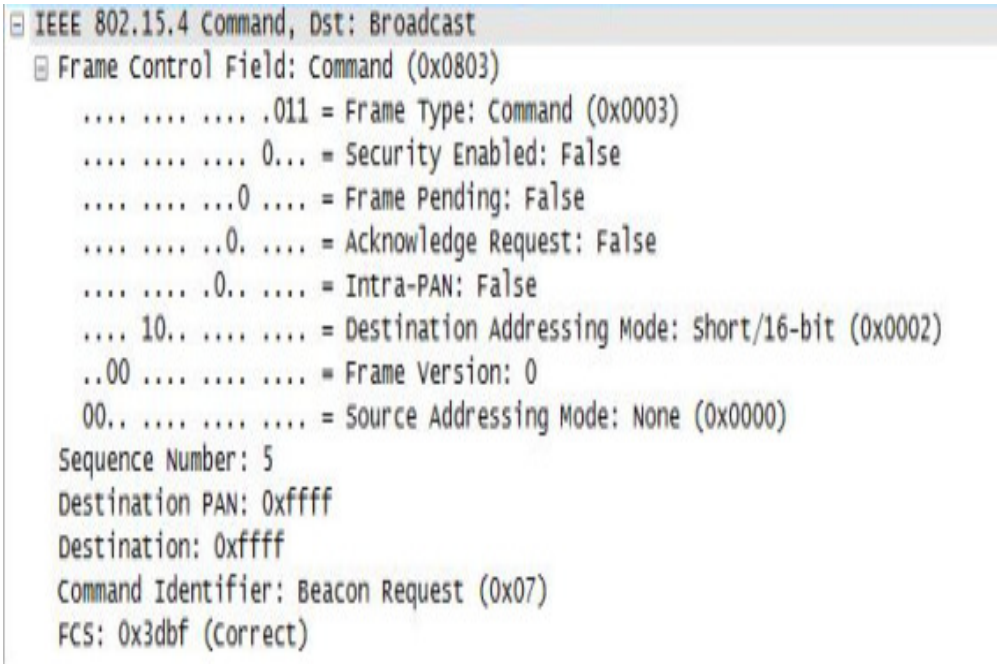

Figure 5. Beacon Request Frame Description

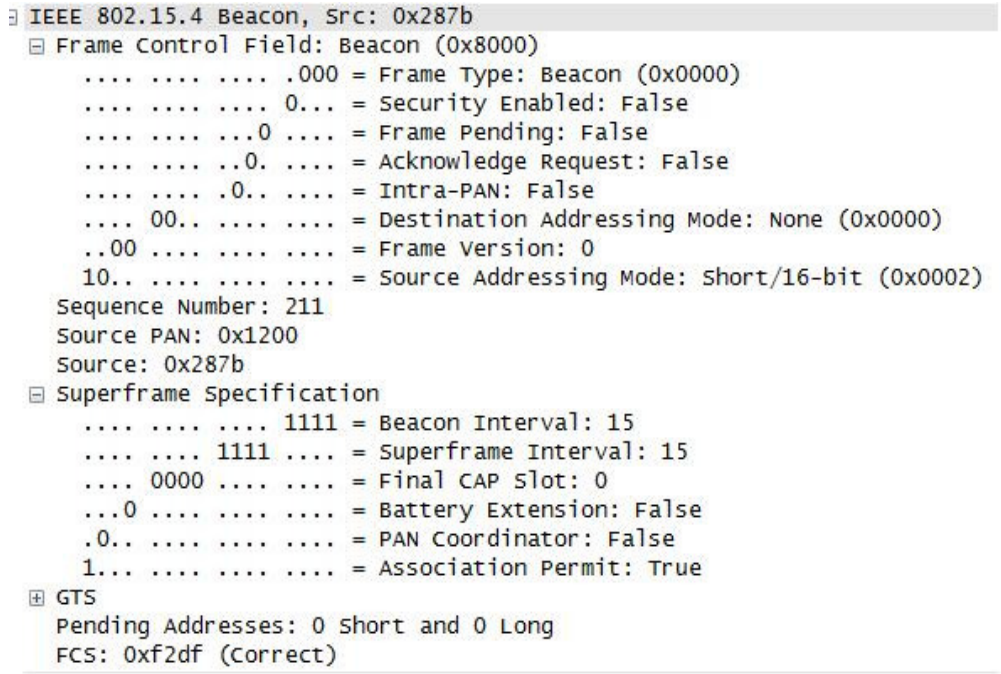

Figure 6. Beacon Response Frame Description

Stopping the attack by identifying the network discovery process at its early stages is essential. As long as the attacker gets to perform the Zbstumbler attack, or any other attempt to discover the channel, PAN ID, and the source address other attacks becomes easier. If network discovery is not stopped, it becomes easier to discover more information about the network in later attacks (replay, key provisioning attack, etc.).

Even though ZigBee uses AES, sending the key in a plaintext over the wireless network defeats AES security. So having the ability to eavesdrop and capture the key allows the attacker to decrypt targeted traffic or impersonate the authentication process and be part of the network.

To better understand the functionality of the intrusion detection system we have setup an algorithm that shows how the detection process is identified. Details will be discussed in the following section.

6. IDS Implementation and Techniques

The cognitive radio (CR) is configured to monitor the activity of the ZigBee network. It acts as a packet analyzer that analyzes the nodes' communication with each other, coordinator or router in real time. The packets captured are compared to the knowledge base of attack signatures that were previously programmed based on previously gathered attack signatures. Mainly, the intrusion detection is set to detect any discovery frame packet that has the following format (0x03 0x08 0x-- 0xff 0xff 0xff 0xff 0x07 0x-- 0x--) .

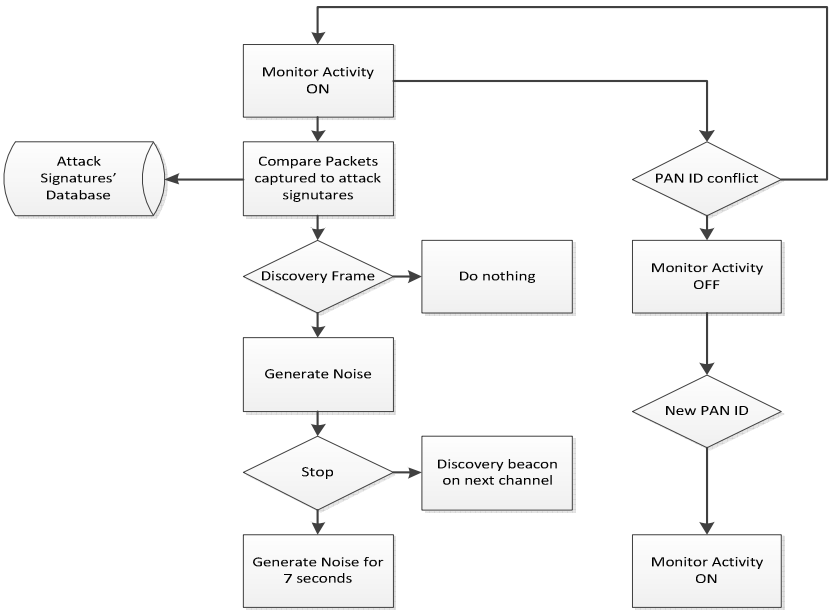

Figure 7. Intrusion Detection Algorithm

The network discovery process shouldn't be allowed except in the case where the network is being established for that reason. The intrusion detection should be turned on (start monitoring) immediately after forming the network and connecting all required devices (routers, coordinators, and nodes). In case, we are adding a device to the network (join), the CR should be disabled to allow the normal joining function. In the case of rejoining the network, there is no need to turn off the intrusion detection system since the beacon request will have a different format and the node already has the network information such as PAN ID, extended PAN ID, and short address.

In order to stop the network discovery that is considered a malicious activity, the CR will generate a noise signal lasting for 7 seconds (the maximum scanning time on each channel is 5 seconds). On the other hand, the system may stop the noise earlier in case the CR has detected a discovery beacon request on the next consecutive channel.

Finally, if the network detects a PAN ID conflict, CR will disable itself upon receiving the PAN ID conflict notification until the coordinator performs an active scan to select a new appropriate PAN ID. Also, if the CR receives the channel change indication message from the channel master, it will change its operating channel to the new one.

7. Assessing the Functionality of the IDS

The current IDS implemented is focused on protecting the ZigBee network from the set of attach that are presented earlier (network discovery, packet and key sniffer, packet replay, location tracking attack). Any other attacks that the system is not aware of are still considered as a threat to the network. In other words, the IDS system is not adaptive; however, it can be further developed in order to detect the attacks according to a learning scheme.

Even though, the IDS implementation assumes that the attacker is not aware of the IDS system present in the network. In case the attacker was acknowledge of the CR presence, he can deviate the attack toward the cognitive radio itself. Once the attacker was able to shut down the

cognitive radio, the network is open again and unsecure. Possible attacks on the cognitive radio include denial of service attack [16].
The IDS needs to be turned off while adding a trusted device into the network. If the attacker stated to attack during that short period, the attacker can be part of the network. However, this possibility is less likely to happen, the attacker needs time to scan all ZigBee channels to figure out the operating channel and the PAN ID (1 to 2 minutes). The time needed to turn on/off the cognitive radio is around 5-10 seconds.

The IDS system is implemented using USRP I which means that it is capable for scanning two consecutive channels. This may leave some of the channels opens or not considered for operation. The spectrum is not completely secured; there are some channels that remain uncovered. More details about resolving the issue is discussed in future work.

8. Conclusion

In this paper, a new approach using CR as an intrusion detection system that is designed for ZigBee networks is presented as a step forward to implement a more secure sensor network. A testbed was developed and several attack scenarios for ZigBee Networks have been tested on the testbed. Adding an intrusion detection system using CR eliminates the threats that are presented by the stack pitfalls, and requires no particular change to the existing stack or network. It was observed that the network discovery process is the key in all attack scenarios. Since the network discovery can not be prohibited, malicious attempts that mimic the network discovery process are identified using the intrusion detection system designed. Moreover, the approach presented has successfully secured this process without interrupting or adding traffic to the ZigBee network.

Future work include implementing the IDS using USRP II instead of USRP I. USRP II has the ability to scan four consecutive channels at a time. In this way, we make sure we are securing a 25% of the spectrum. However, according to [1] for a single channel, the number of dropped packets was around 1% for the longest possible packet. Four channel capture performance had a PER of 15% for the largest packet. Compared to the USRP I the PER of URSP I is negligible. The USRPII does have a 14-bit ADC compared to the 12-bit ADC in the USRP, which could contribute to a higher accuracy.
In order to capture all 16 channels and if trying to sample a very large window, PCI express link should be used. The PCI express link is capable of 1GB/s per lane.

Other prevention techniques could be used. One of the prevention techniques is sending a frame in which the node that is being targeted is being shut down .Other than that, the intrusion detection could be programmed to be adaptive, so it has the ability to automatically identify the attacks according to a specified learning mechanism of the implemented ZigBee network daily activities. Last but not least, the intrusion detection could be programmed in order to send fake beacon request to the attacker in order to lead the attacker to a channel in which the ZigBee network is not operating on. In this way, the attacker is confused between the actual traffic and the noise send over another channel.

References

[1] D. Gislason and I. Books24x7, Zigbee Wireless Networking. Burlington, Mass.: Newnes/Elsevier, 2008.

[2] S. Farahani and I. Books24x7, ZigBee Wireless Networks and Transceivers. Burlington, Mass.: Newnes/Elsevier, 2008.

[3] A. Elahi and A. Gschwender, ZigBee Wireless Sensor and Control Network. Upper Saddle River, N.J.: Prentice Hall, 2010.

[4] J. Cache, J. Wright, V. Liu, E. Scott, B. Antoniewiecz, C. Wang and I. Books24x7, Hacking Exposed Wireless. New York: McGraw-Hill Companies, 2010.

[5] G. Dini and M. Tiloca, "Considerations on security in ZigBee networks," in Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on, 2010, pp. 58-65.

[6] Bin Yang, "Study on security of wireless sensor network based on ZigBee standard," in Computational Intelligence and Security, 2009. CIS '09. International Conference on, 2009, pp. 426-430.

[7] Meng Qianqian and Bao Kejin, "Security analysis for wireless networks based on ZigBee," in Information Technology and Applications, 2009. IFITA '09. International Forum on, 2009, pp. 158-160.

[8] Tulin Mangir, Lelass Sarakbi, Harvy Younan "Analyzing the Impact of Wi-Fi Interference on ZigBee Networks Based on Real Time Experiments," International Journal of Distributed and Parallel Systems (IJDPS) Vol.2, No.4, July 2011.

[9] Harvy Younan. "Experimental analysis of the interference impact of Wi-Fi on ZigBee using Cognitive Radio," M.S.thesis,California State University of Long Beach, Long Beach, CA, USA, 2011.

[10] Tulin Mangir, Mukul Khairatkar, "Network Access Security Policies for Cognitive radio," SDR '08, Washington, D.C., USA., Oct. 26-30, 2008.

[11] Tulin Mangir , Mukul Khairatkar , "Security for Smart Grid Using Cognitive Radio," Southern California Smart Grid Research Symposium, October 2009, USC, LA., CA., USA .

[12] Mukul Anil KhairatKar. "Building Cognitive Radio for physical Layer Intrusion Detection," M.S.thesis,California State University of Long Beach, Long Beach, CA, USA, 2009.

[13] Leslie Choong, " Multi-channel IEEE 802.15.4 packet capture using software defined radio," UCLA Networked & Embedded Sensing Lab, 03 2009.

[14] Thomas Schmid, " GNU radio 802.15.4 en and decoding," UCLA '06 Los Angeles, California USA, 2006.

[15] Blossom, E. 2009. Gnu radio. http://www.gnu.org/software/gnuradio/.

[16] Li Zhu; Huaibei Zhou; , "Two Types of Attacks against Cognitive Radio Network MAC Protocols," Computer Science and Software Engineering, 2008 International Conference on , vol.4, no., pp.1110-1113, 12-14 Dec. 2008.