Wireless Intrusion Detection of Covert Channel Attacks in ITU-T G.9959-Based Networks

Jonathan Fuller, Benjamin Ramsey, John Pecarina and Mason Rice
Air Force Institute of Technology, Wright-Patterson AFB, USA
jonathan.fuller@afit.edu
benjamin.ramsey@afit.edu
john.pecarina@afit.edu
mason.rice@afit.edu

Abstract: We introduce herein an information hiding technique for injecting manipulated packets into wireless sensor networks (WSNs). We exhibit how an attacker can apply information hiding as a type of covert channel attack over radio frequency transmissions into the WSN. The feasibility of our injection method is demonstrated through an attack on the most common implementation of the ITU-T G.9959 recommendation, commercially known as Z-Wave. More specifically, we illustrate that after accessing a Z-Wave gateway controller through compromising the WLAN backbone, the attacker has the ability to install malware. The malware scans incoming Z-Wave packets for information hidden in Media Access Control (MAC) frames received by the Z-Wave controller. Upon identification of hidden information, a Reverse Secure Shell is initiated through the WLAN back to the attacker. The outcomes of this attack include control of the Z-Wave network and access to the networked devices on the target WLAN from any Internet connected device. Given this new application of information hiding techniques to Z-Wave networks, we recognize the need for countermeasures. We therefore offer an effective Misuse-based Intrusion Detection System (MBIDS) capable of distinguishing between manipulated and correctly formed packets. A Universal Software Radio Peripheral (USRP) Software-Defined Radio (SDR) is used in conjunction with a packet monitoring tool capturing incoming transmissions and inspecting them for any violations of the ITU-T G.9959 MAC specification. We then analytically and experimentally estimate the efficacy of the USRP as a packet capture device in a realistic test setup, and then evaluate the total efficiency of our MBIDS solution. By employing the MBIDS in the Z-Wave network, we show the MBIDS is capable of detecting packet manipulation attacks with 92% mean accuracy.

Keywords: covert channel, wireless sensor networks, wireless threats, Z-Wave, intrusion detection

1. Introduction

Wireless Sensor Networks (WSNs) consist of multiple sensor nodes that collect information, respond to commands, and report updates to other nodes in the network. The use of WSNs is emerging because they extend communications ranges at low-cost, low-power, and low-complexity (Patel et al. 2014). WSN are applied in military surveillance, health care, environmental science, and home automation (Ramsey and Mullins 2013). Reaves and Morris (2012) discuss the security implications of various protocols used in critical infrastructure analyzing attack classes including reconnaissance, packet injection, denial-of-service, and man-in-the-middle. They further analyze IEEE 802.11-based protocols, IEEE 802.15.4-based protocols, and proprietary protocols. However, since proprietary protocols are closed-source in nature, there are few security research publications regarding their use. Therefore, researchers have struggled to conduct a thorough analysis.

Of the numerous wireless protocols, those based on the International Telecommunications Union-Telecommunication Standardization Sector (ITU-T) G.9959 recommendation specifying short range narrow-band sub-GHz communications (ITU-T G.9959 2015) have significant growth potential in WSNs. The most common implementation of the ITU-T G.9959 recommendation is known as Z-Wave. However, security evaluations of Z-Wave are difficult because developers are required to sign nondisclosure agreements (NDAs) limiting the availability of tools to perform open source research. New and emerging network protocols are often initially believed secure, but are not vetted until tools exist for security research (Goodspeed et al. 2012).

Previous works discover Z-Wave vulnerabilities and indentify points of compromise at the physical (PHY) layer or vulnerabilities in the Z-Wave network when controlled by a globally accessible controller. Conversely, to the best of our knowledge, this work presents the first discussion and evaluation of information hiding in the Medium Access Control (MAC) layer, which is leveraged to exploit the Z-Wave network.

For this work, we evaluate the Raspberry Pi 2 Model B with RaZberry Pi General-Purpose Input/Output Daughter Card (hereinafter, RaZberry Pi), a Z-Wave gateway controller. We find that the RaZberry Pi is vulnerable to

exploitation after gaining access by (i) compromising the Wireless Local Area Network (WLAN) that the RaZberry Pi resides on, (ii) physical access, or (iii) social engineering attacks to install malware on a device.

Using the RaZberry Pi, we offer three contributions. First, we show the feasibility of hiding information in MAC frames as a type of covert channel. Second, we present a new covert channel-initiated attack that allows an attacker full control of the Z-Wave network and access to other WLAN devices. Given the possibility of such attacks, we thirdly develop a Misuse-based Intrusion Detection System (MBIDS) capable of identifying manipulated packets and analytically and experimentally evaluate its effectiveness with 92% mean accuracy. This is the first information hiding attack in Z-Wave networks but also the first proposal and demonstration of a MBIDS capable of monitoring Z-Wave networks.

2. Background and related work

In this section, we present relevant technical details of the Z-Wave protocol. We then discuss related works in the security evaluation of the Z-Wave protocol.

2.1 The Z-Wave protocol

All Z-Wave networks adhere to the ITU-T G.9959 PHY and MAC layer specification, ensuring interoperability between vendor devices, but differ at the application layer (Figure 1). ITU-T G.9959-based networks operate at 908.4 MHz in North America and additional frequencies in other regions. A Z-Wave network consists of control nodes that send commands and slave nodes that respond to commands. As a meshed topology network, slave nodes also forward commands to other nodes outside the radio frequency (RF) range. Message forwarding has a four-node hop limit and a maximum of 232 nodes are allowed. For network overlap, each Z-Wave network has a unique home identification (ID).

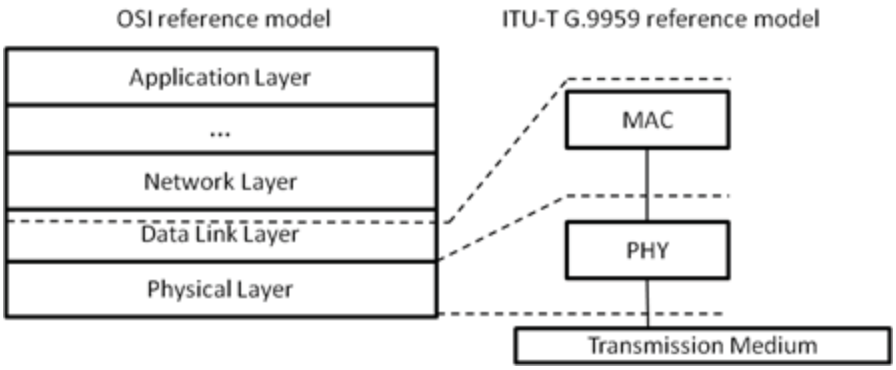

Figure 1: Protocol reference model of a Z-Wave transceiver

2.1.1 PHY layer

The PHY layer uses carrier sense multiple access with collision avoidance to moderate access to the wireless medium. Z-Wave supports three data rates: 9.6 kbps (R1), 40 kbps (R2), and 100 kbps (R3). Figure 2 illustrates the PHY Protocol Data Unit consisting of the PHY Header, PHY Service Data Unit (PSDU) and the End of Frame delimiter (EoF). The maximum PSDU size is 170B while operating at R3. However, if the transceiver operates at R1 or R2, the maximum PSDU size is 64B.

2.1.2 MAC layer

The MAC layer (Figure 2) is responsible for transferring data between nodes and frame acknowledgment, data validation, and retransmission. The three types of MAC frames are singlecast, multicast, and acknowledgment. When a node is transmitting a singlecast frame, there is one destination address. Upon receipt of a singlecast frame, a device responds with an acknowledgment frame. Acknowledgment frames and singlecast frames are structured similarly with the exception of a zero-byte MAC Service Data Unit (MSDU) in the acknowledgment frame. Conversely, multicast frames are sent to multiple destination nodes without acknowledgments.

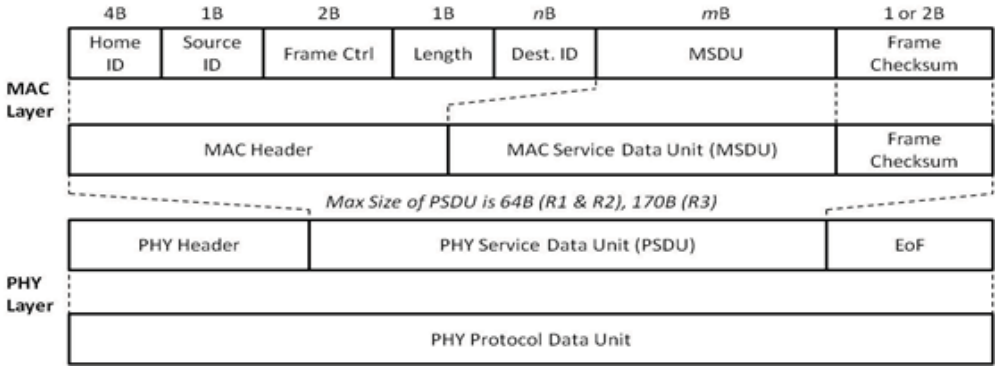

Figure 2: Z-Wave PHY and MAC layer frames

2.2 Vulnerability analysis and exploitation in Z-Wave networks

Previous works exhibit vulnerability exploitation in Z-Wave networks. These works are categorized as a bottom-up or top-down approach exploiting PHY layer transmissions and device vulnerabilities respectively. The bottom-up approach includes packet capture and injection attacks whereas the top-down approach includes exploitation of Z-Wave gateways.

2.2.1 Packet injection in Z-Wave networks

Packet injection attacks enable an attacker to masquerade as a legitimate user. Due to the broadcast nature of wireless networks, an attacker armed with tools capable of capturing wireless information can inject forged packets into the network.

Since Z-Wave developers are required to sign NDAs, researchers use publicly available tools to conduct packet capture and injection attacks. Fouladi and Ghanoun (2013) develop Z-*Force* (not open source), a packet capture and injection tool. Using Z-Force, they exploit an encryption vulnerability resulting from improper vendor implementation and open a secure door lock. The authors of (Picod, Lebrun, and Demay 2014) also create a packet capture and injection tool, *Scapy-Radio*. This tool is used to sense a legitimate alarm-on command and automatically inject a subsequent alarm-off command, effectively nullifying alarm notification. Extending Scapy-Radio, Badenhop et al. (2015) develop the *AFIT Sniffer* using a USRP N210 and a Z-Wave Wireshark dissector. Their tool is used to passively discriminate between Z-Wave devices by functionality and vendor. We use the AFIT Sniffer to conduct packet capture for our experiments.

2.2.2 Gateway attacks in Z-Wave networks

There are two types of control devices in Z-Wave networks. Portable controllers are hand held devices that allow a user to control other devices within RF range. Gateway controllers provide the user the ability to manage their Z-Wave network locally or globally. Local accessibility lessens the control span whereas global accessibility provides access to the network from any location (e.g., using a mobile device) but also introduces new vulnerabilities. In (Crowley, Bryan, and Savage 2013) several vulnerabilities are discovered in Z-Wave gateways caused by lack of user authentication, lack of encryption, and open ports. Once an attacker gains access to the gateway controller, HTTP packet capture and injection are used to control connected devices. Barcena and Wueest (2015) produce similar results by performing firmware modification giving an attacker full control of the gateway.

Insecurities in numerous Z-Wave gateways are exploited demonstrating the ability to insert a rogue controller creating persistent access to Z-Wave devices (Fuller and Ramsey 2015). Although a persistent attack channel is created, if the target realizes a rogue controller was inserted or their gateway has been compromised, following the mitigation strategies proposed by Fuller and Ramsey (2015), an attacker is unlikely to gain further access to the Z-Wave network.

These early studies illustrate possible attacks resulting from packet capture and injection or compromise of the Z-Wave gateway. After further evaluation of the Z-Wave protocol, we find that new vulnerabilities exist. To illustrate the type of vulnerabilities, Martins and Guyennet (2010) use information hiding techniques in the MAC

layer of the IEEE 802.15.4 protocol. Information hiding is embedding a secret message into another media to provide a covert channel. Martins and Guyennet (2010) use this technique to secure the data transfer in WSNs preventing an attacker from eavesdropping and identifying sensitive information. The ability to hide information in the MAC layer of the IEEE 802.15.4 protocol motivates the evaluation of the feasibility of those techniques in an ITU-T G.9959-based protocol.

3. Information hiding in the Z-Wave MAC frame

In this section, we show the possibility of hiding data in the Z-Wave MAC frame. As discussed in Section 2.1.2, the MAC layer supports three frames. The maximum size of the payload in each frame differs depending on the data rates end devices use. Although newly released Gen5 Z-Wave devices can operate at rate R3, we will focus on identifying positions to hide information in the MAC frame at rates R1 and R2 for singlecast (acknowledgment messages are a subset of singlecast) and multicast messages (Figure 3). The maximum packet size at rates R1 and R2 is 64B. This evaluation represents the most challenging case for information hiding since the payload size available to hide information at R1 and R2 are less than half of R3.

Singlecast MAC frames includes the home ID, source ID, frame control, length, and destination ID totalling 9B (Figure 2 and 3). A one byte non-correcting frame checksum is used to validate the MAC frame for data rates R1 and R2 (2B are used for R3). Given our channel configuration, we know that the frame checksum of messages supported in our experimental Z-Wave network is one byte. Thus, there are 54B remaining for the MSDU (frame payload). A similar calculation can be done to discover the maximum payload length of the multicast frame (Figure 3). There are 39B used in the multicast frame excluding the MSDU. Therefore, the MSDU is 25B.

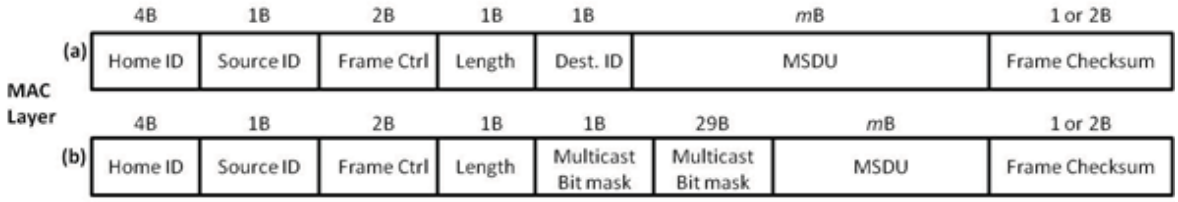

Figure 3: MAC layer frame structure: (a) singlecast frame and (b) multicast frame

3.1 Space available to hide information

The MSDU field has a variable length and contains the frame payload information. The length depends on the command class being sent to a device in the Z-Wave network. The command class determines the function that needs to be performed by a device. One example is the *Basic* command class which is supported by all Z-Wave devices. The Basic command class uses three commands: SET to turn a device on or off, GET to request a device status, and REPORT to respond to a request.

Since we know that all devices, including the controller, will accept messages containing the Basic command class (RaZberry Project n.d.), we can determine the length required in the MSDU for the Basic command class portion of the payload. This will allow us to determine available bytes remaining.

The representation of the Basic command class in the MAC frame is *0x20*. The byte field following is dependent upon the command SET (*0x01*), GET (*0x02*), or REPORT (*0x03*). In either case, the payload length needed to support the Basic command class is at least 3B. Since all Z-Wave devices support the Basic command class, an attacker can hide up to 51B of data in a singlecast MAC frame and up to 22B in a multicast MAC frame. When the Z-Wave transceiver receives a packet, the header and EoF are stripped away as the packet moves up the protocol stack. Once at the application layer, the command class function is executed and the injected bytes are ignored.

4. Z-Wave covert channel-initiated reverse secure shell attack

This section presents the first information hiding attack in a Z-Wave network. The experiment setup includes the target RaZberry Pi on a WLAN backbone and a HP EliteBook 8570w 64-bit with i7-3720QM CPU, Ubuntu 14.04, LiClipse 2.2.0.2, USRP and VERT900 3dBi antenna, and the AFIT Sniffer software framework.

Fuller and Ramsey (2015) illustrate the possibility of gaining access to the target Z-Wave gateway. Additionally, after gaining access, the attacker can upload a Python script that is executed but sleeps for a predetermined

time (our experiment - ten minutes). When re-access to the Z-Wave network is needed, the attacker executes a Secure Shell (SSH) server on their machine. Using Scapy-Radio, the attacker crafts a Z-Wave packet containing hidden information and transmits the packet to the Z-Wave gateway using a Software-Defined Radio (SDR). To ensure the injected packet is accepted by the Z-Wave network, the attacker first captures a Z-Wave packet and extracts the home ID using it to construct the packet similar to Figure 4. The attacker follows the ITU-T G.9959 8-bit checksum algorithm and calculates the checksum before injection.

```
[DA 67 9E 36] [02] [41 03] [14] [01] [20 01 FF] [FE FE 01 A2 56 C0 05] [51]
     (a)       (b)    (c)    (d)   (e)     (f)                (g)          (h)
```

Figure 4: MAC frame consisting of (a) home ID, (b) source ID, (c) frame control, (e) length, (e) destination ID, (f) basic command class, command, and payload, (g) hidden information in MSDU, and (h) checksum

The key to this attack is locating Z-Wave packets on the gateway controller. The RaZberry Pi keeps a detailed log that contains all actions on the Z-Wave network. Once awake, the Python script scans the log file for any injected packet containing the hidden information. The injected packet contains a marker [*FE FE*] in the MSDU (Figure 4.g). This marker is used because it is unlikely that a standard Z-Wave packet will contain consecutive bytes [*FE*]. Upon identification, the Python script dissects the packet retrieving needed information including the attack type [*01*] and IP address [*A2-56-C0-05*] (Figure 5). Any attack type can be added depending on the needs of the attacker. Examples include Ping of Death to deny service or Reverse File Transfer to retrieve the */etc/passwd* file or other important files.

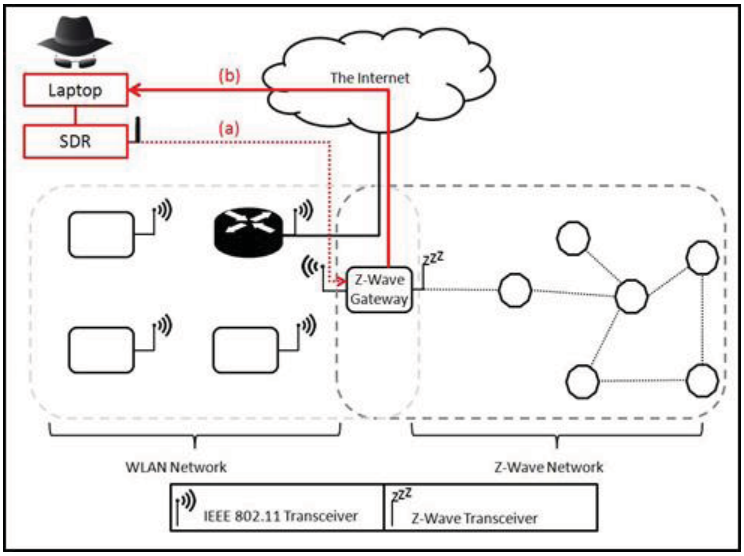

Figure 5: After the attacker executes a listening server, (a) a malicious Z-Wave packet is injected to the gateway using a SDR. When the malware on the infected gateway detects the injected frame, it retrieves the IP address and (b) opens a R-SSH. The attacker now has full control of the Z-Wave gateway and access to WLAN devices

After retrieving the hidden information, the Python script clears the log file to cover tracks and initiates a R-SSH client to the attack SSH server at IP address 162.86.192.5 (Figure 6 (a) (3)). The attacker can now perform command line instructions as *root* user on the target RaZberry Pi allowing access to Z-Wave devices and WLAN devices. Once complete, the attacker simply closes the connection which instructs the Python script to sleep for the predetermined time.

Although this attack was conducted within 20 meters of the target location, more than enough distance for an attacker to remain inconspicuous, previous works demonstrates the exploitation of WSN devices from 40 miles away (Apa and Hollman 2013). Given improved tools, an attacker with this capability need not be in close proximity to the target location to inject packets in to the network.

Given this new application of information hiding techniques to Z-Wave networks, we recognize the need for countermeasures to protect against packet manipulation attacks including the aforementioned. We therefore

develop a MBIDS capable of distinguishing between manipulated packets and correctly formed packets allowing for the first monitored Z-Wave network.

```
[D] [zway] Job 0x13 (SwitchBinary Get): success                          (a)
[I] [zway] Removing job: SwitchBinary Get     (1)  (2)     (3)
[D] [zway] RECEIVED: ( 04 00 02 0A 20 01 FF FE FE 01 A2 56 C0 05 14)
[D] [zway] SENT ACK
[I] [zway] Job 0x13 (SwitchBinary Get): Delivered
```

```
[D] [zway] Job 0x13 (SwitchBinary Get): success                          (b)
[I] [zway] Removing job: SwitchBinary Get
[D] [zway] RECEIVED: ( 04 00 02 03 20 01 FF 2E )
[D] [zway] SENT ACK
[I] [zway] Job 0x13 (SwitchBinary Get): Delivered
```

Figure 6: The RaZberry Pi log file. (a) Log representation of manipulated Z-Wave packet containing information hidden in the MSDU. (1) Marker signifying that a packet has been injected, (2) Attack type, and (3) IP address in hexadecimal representing 162.86.192.5. (b) Log representation of legitimate Z-Wave Packet

5. Misuse-based intrusion detection system

In this section, we discuss the efficiency of the AFIT Sniffer (Badenhop et al. 2015) as a packet capture device. We then experimentally evaluate the accuracy of our packet monitoring tool as a MBIDS.

5.1 Packet capture device - AFIT Sniffer

Our proposed MBIDS is designed to work within the WLAN that the RaZberry Pi resides on and aims to detect attacks against the Z-Wave gateway and devices. Using the AFIT Sniffer developed in (Badenhop et al. 2015), the MBIDS dissects packets captured by the AFIT Sniffer and extracts the MAC frame for evaluation. The experiment setup is identical to that of the attack setup in Section 4. We test the efficiency of the AFIT Sniffer and RaZberry Pi to determine the best case detection of the MBIDS.

We conduct 500 tests on each device wherein each test consists of sending one attack packet to the gateway controller. The mean packet reception rate for the RaZberry Pi is 99.5% whereas the AFIT Sniffer maintains a mean packet reception rate of 94% (Figure 7). The RF front end on the RaZberry Pi is tuned per manufacturer specifications and is more effective than the AFIT Sniffer SDR.

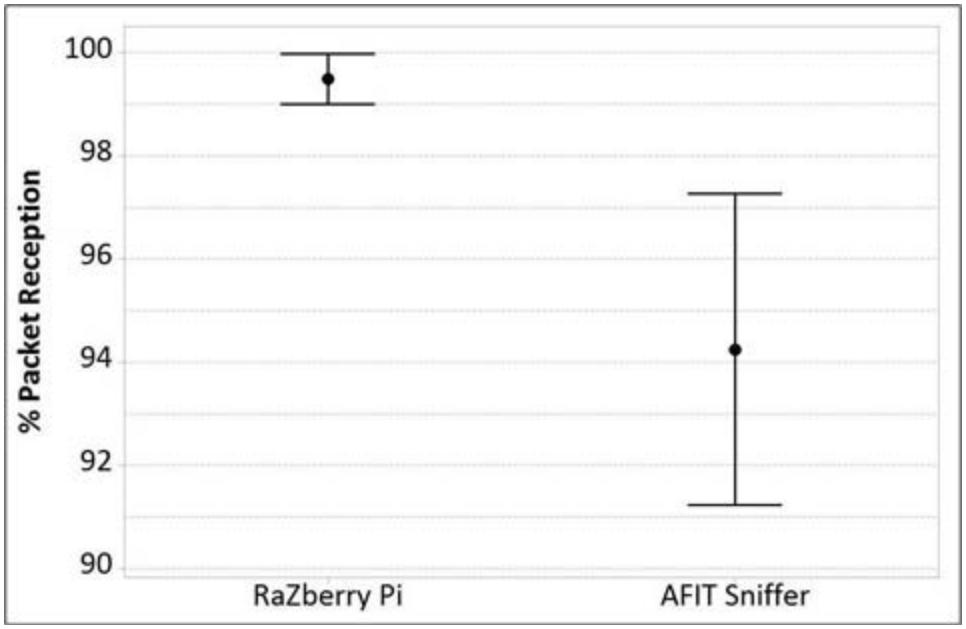

Figure 7: RaZberry Pi and AFIT Sniffer packet reception rates (99 % confidence interval for the mean)

5.2 Packet monitoring tool

Given the packet capture accuracy of the AFIT Sniffer, we develop a packet monitoring tool to evaluate packets with valid checksums that are received. Based on the ITU-T G.9959 MAC frame specification, we differentiate between known-good packets and improperly formed packets. Our System Under Test (SUT) diagram is shown in Figure 8. The SUT consists of the Component Under Test (CUT), *MBIDS.py*, and the aforementioned AFIT Sniffer. Workload parameters are the number of manipulated injected packets with invalid byte fields. Hardware and software system components include those listed in Section 4. The metric used to evaluate the efficiency of MBIDS.py is the percent of misuse packets detected with respect to the packet capture accuracy of the AFIT Sniffer.

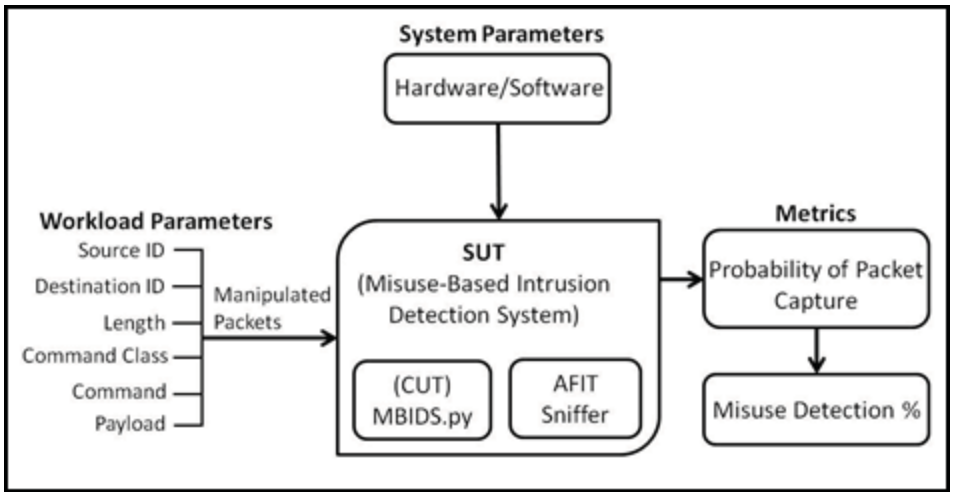

Figure 8: System Under Test: 500 of each workload parameter are used for testing. The resulting metric is the percent of misuse detected

Captured packets are parsed by MBIDS.py for evaluation. As discussed in Section 2.1.1, the MAC frame does not exceed 64B. When MBIDS.py receives packets, the length is checked for validity. If it exceeds 64B, the packet is deemed invalid and the misuse is logged. In this instance, an attacker is likely attempting to inject significant amounts of data in one single packet.

The RaZberry Pi keeps a record of all device node IDs and all command classes of each Z-Wave device. When a packet is received by our sniffer, the MBIDS requests this information from the RaZberry Pi and checks the source ID and destination ID in the MAC frame. If they are invalid, the misuse is logged. Otherwise, the command class in the MAC frame is then compared to the available command classes that the device represented by the source ID supports. If the command class is not supported, the misuse is logged; otherwise, further packet evaluation occurs.

The next step in the evaluation of the MAC frame is the command and remaining payload. After the command class has been verified as supported, the command must be checked. For example, if an attacker injects a packet into the network where the MSDU is [*20-01-FF-FE-FE-02-A2-56-C0-05*], the MBIDS will note it is a Basic command class (*0x20*) with command SET request (*0x01*). If the command is anything but SET, GET, or REPORT, it is considered a misuse case. In this case the command is SET and evaluation of the remaining payload occurs. When the command SET is used, subsequent values should be OFF (*0x00*) or ON (*0xFF*). This requires an MSDU of length 3B (*20-01-FF*) but contains 10B. Therefore, the MBIDS will log the misuse since the payload is abnormal given the specified command class.

The Z-Wave protocol supports over 100 command classes. In order to develop a MBIDS, it is required that every command class be properly understood to detect any violation of their structure. As discussed, the *Basic* command class has three commands, SET, GET, and REPORT. There are five combinations of *Basic* commands supported by the Z-Wave protocol. As the most basic command class, *Basic* has the fewest commands available. At a modest estimation, over 100 known command classes would allow for over 2,000 possible combinations to account for in the MBIDS. Therefore, we provide a proof of concept with 10 of the most common command classes to illustrate the efficacy of this approach. We conduct deep packet inspection of normal payloads to develop known-good packet standard that injected packets can be compared against.

5.3 Test methodology

We engineer a realistic Z-Wave network with a RaZberry Pi and multiple devices. The RaZberry Pi and MBIDS are co-located to ensure that whatever packets are likely received by the RaZberry Pi are also be received by the MBIDS. Using the Scapy-Radio framework, we use a HackRF SDR to inject packets.

For each workload parameter listed in the SUT (Figure 8), we send 500 packets. Given the 6% error rate of the AFIT Sniffer, we know a priori that the MBIDS will not detect 100% of manipulated packets even in a perfect test.

6. Results and analysis

We craft 3,000 packets that violate the ITU-T G.9959 MAC specifications to test the accuracy of our MBIDS. Packets include 500 of each field (Table 1). The HackRF connected to a second laptop is placed one meter from the AFIT Sniffer. Since we know the effectiveness of the AFIT Sniffer as a packet capture device (Figure 7), we are only interested in the effectiveness of the MBIDS to detect misuse cases. Therefore, placing the packet injection tool in close proximity to the MBIDS reduces missed packets and allows for a more accurate MBIDS evaluation.

Table 1 lists a 100% mean detection rate of invalid packet length, source ID, destination ID, command class, and command. All validation checks in the MBIDS are deterministic and are sure to detect violations. As an example, a door lock manufacturer engineers a door lock to only accept one cut of key. Any other cuts used should not work. The manufacturer is sure of the design but still tests non-standard possibilities to ensure the door lock operates as intended. Similarly, we test with values that are not supported knowing the outcome but need to ensure MBIDS.py is engineered correctly. If the attacker is not aware of the source ID, destination ID, valid packet length, command class, or command, they are likely to inject a packet with non-standard values and will certainly be detected and logged as a misuse case. Therefore, our results are as expected.

Table 1: Mean detection rate of injected packets. If a field is invalid, a misuse is logged. (99% confidence interval for the mean)

Field	Detection Rule	Mean Detection Rate
Packet Length (PL)	If PL > 64B, log as misuse.	100%
Source ID (SID)	If parsed SID is invalid, log as misuse.	100%
Destination ID (DID)	If parsed DID is invalid, log as misuse.	100%
Command Class (CC)	If parsed CC is invalid, log as misuse.	100%
Command (CMD)	If parsed CMD is invalid, log as misuse.	100%
Payload (PLD)	If parsed PLD is invalid, log as misuse.	92%

The MBIDS payload mean detection rate is 92% (Table 1) because the specifics of the proprietary Z-Wave protocol command class parameters (payload) are not fully known. We therefore conduct packet inspection and comparison to understand byte fields in the payload. Vendors implement command class parameters differently, resulting in byte field variations. Given the variety, we cannot ascertain all possible combinations are accounted for resulting in the 92% mean misuse detection rate.

False positives are possible and will arise if an attacker crafts a packet using valid packet length, source ID, destination ID, command class, and command. The attacker then pads the payload with hidden information. However, this padded information, although intended to be malicious, does not violate any misuse cases. It is therefore not detected by the MBIDS. This also contributes to the 92% mean misuse detection rate. Lastly, our packet capture device, AFIT Sniffer, has a 6% error rate which means that our MBIDS did not evaluate every packet sent since some were not captured.

Of the six workload parameters we test, only one does not result in 100% detection. Conducting 3000 tests of independent packet transmissions with invalid packet length, source ID, destination ID, command class, command, and payload results in an overall accuracy of at least 92% mean misuse detection rate.

7. Future work and conclusion

In this paper we present the design, implementation, and evaluation of a Z-Wave MBIDS. The necessity of the MBIDS is motiviated by numerous attacks against Z-Wave networks, specificially new packet manipulation attack discussed herein. The malware installed on the Z-Wave gateway controller periodically scans for maliciously injected frames in the Z-Wave network. Upon identification of malicious packets, the malware initiates a R-SSH back to the attacker that allows for Z-Wave network control and WLAN device access. This attack is possible because of vulnerabilities in Z-Wave gateways and the ability to hide information in a Z-Wave packet. Although the attack was only demonstrated against the RaZberry Pi, many gateways use log files that store all packet transmissions within the network, including those previously investigated (Fuller and Ramsey 2015). Therefore, other Z-Wave gateways are susceptible to packet manipulation attacks.

The Z-Wave MAC frame allows for the addition of extra bytes to the payload before transmission and the Z-Wave protocol allows for the reception and execution of commands in the payload, while disregarding additional bytes that seem to serve no purpose. Those extra bytes are not discarded or rejected and enable a covert Z-Wave channel. Manipulating Z-Wave packets and successfully injecting them into the network motivates the need for countermeasures. The MBIDS evaluates pakcets for violations. If violations are detected, a misuse case is logged. Although only a proof-of-concept evaluating 10 of the most common command classes, this work demonstrates the feasibility of an intrustion detection sytem capable of monitoring Z-Wave networks. Using a USRP and a packet monitoring tool, MBIDS.py, we achieve a 92% mean accuracy at identifying manipulated Z-Wave packets. We intend to offer the Z-Wave MBIDS as an open source tool for growth and extension as well as continue to investigate the byte fields in other command classes contributing to the completeness of the MBIDS.

Acknowledgements

This research was supported in part by the Department of Homeland Security ICS-CERT. The views expressed in this work are those of the authors and do not reflect official policy of the United States Army, United States Air Force, Department of Defense, or the U.S. Government.

References

Apa, L. and Hollman, C. (2013) "Compromising Industrial Facilities from 40 Miles Away," *Black Hat Conference*, Nevada, USA, July.

Badenhop, C., Fuller, J., Hall, J., Ramsey, B., and Rice, M. (2015) "Evaluating ITU-T G.9959 Based Wireless Systems used in Critical Infrastructure Assets," in *Critical Infrastructure Protection IX*, Shenoi, S. and Rice, M. Eds. Heidelberg, Germany: Springer International Publishing, pp 209-227.

Barcena, M. and Wueest, C. (2015) "Insecurity in the Internet of Things," [online], http://www.symantec.com/content/en/us/enterprise/iot/b-insecurity-in-the-internet-of-things_21349619.pdf.

Crowley, D., Bryan, D., and Savage, J. (2013) "Home Invasion V2.0 - Attacking Network-Controlled Hardware," *Black Hat Conference*, Nevada, USA, July.

Fouladi, B. and Ghanoun, S. (2013) "Security Evaluation of the Z-Wave Wireless Protocol," *Black Hat Conference*, Nevada, USA, July.

Fuller, J. and Ramsey, B. (2015) "Rogue Z-Wave Controllers: A Persistent Attack Channel," *Tenth IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, October, pp 140-147.

Goodspeed, T., Bratus, S., Melgares, R., Speers, R., and Smith, W. (2012) "Api-do: Tools for Exploring the Wireless Attack Surface in Smart Meters," *45th Hawaii International Conference on System Sciences*, January, pp 2133-2140.

ITU-T G.9959 (2015) "Short Range Narrow-Band Digital Radiocommunication Transceivers - PHY, MAC, SAR and LLC layer specifications," January.

Martins, D. and Guyennet, H. (2010) "Steganography in MAC Layers of 802.15.4 Protocol for securing Wireless Sensor Networks," *International Conference on Multimedia Information Networking and Security*, November, pp 824-836.

Patel, H., Temple, M., Baldwin, R., and Ramsey, B. (2014) "Application of Ensemble Decision Tree Classifiers to ZigBee Device Network Authentication Using RF-DNA Fingerprinting," *9th International Conference on Cyber Warfare and Security*, March, pp 176-186.

Picod, J., Lebrun, A., and Demay, J. (2014) "Bringing Software Defined Radio to the Penetration Testing Community," *Black Hat Conference*, Nevada, USA, August.

Ramsey, B. and Mullins, B. (2013) "Defensive Rekeying Strategies for Physical-Layer-Monitored Low-Rate Wireless Personal Area Networks," *Critical Infrastructure Protection VII*, J. Butts and S. Shenoi, Eds. Heidelberg, Germany: Springer, March, pp 63-79.

RaZberry Project (n.d.) "Z-Way Developer's Documentation," [online], razberry.z-wave.me.

Reaves, B. and Morris, T. (2012) "Analysis and Mitigation of Vulnerabilities in Short-Range Wireless Communications for Industrial Control Systems," *International Journal of Critical Infrastructure Protection*, Vol 5, March, pp 154-174.