Rogue Z-Wave Controllers: A Persistent Attack Channel

Jonathan D. Fuller and Benjamin W. Ramsey
Department of Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433 USA
Email: {jonathan.fuller, benjamin.ramsey}@afit.edu

***Abstract*—The popularity of Wireless Sensor Networks (WSN) is increasing in critical infrastructure, smart metering, and home automation. Of the numerous protocols available, Z-Wave has significant potential for growth in WSNs. As a proprietary protocol, there are few research publications concerning Z-Wave, and thus little is known about the security implications of its use. Z-Wave networks use a gateway controller to manage and control all devices. Vulnerabilities have been discovered in Z-Wave gateways, all of which rely on the gateway to be consistently connected to the Internet. The work herein introduces a new vulnerability that allows the injection of a rogue controller into the network. Once injected, the rogue controller maintains a stealthy, persistent communication channel with all inadequately defended devices. The severity of this type of attack warrants mitigation steps, presented herein.**

***Keywords* - Z-Wave; wireless sensor networks; gateway; home automation networks; security; rogue controller; mitigation**

I. Introduction

Wireless Sensor Networks (WSNs) are technologies where computing, communications, and control are tightly coupled. WSNs consist of multiple sensor nodes that collect information, respond to commands, and report updates to other nodes in the network [1]. WSN use is increasing in critical infrastructure, smart metering, and home automation applications [2-5]. A security analysis of WSNs appears in [2]. Although Reaves et al. [2] focus on the security implications of various protocols when used in critical infrastructure, their focus is the analysis of four attack classes (reconnaissance, injection, denial-of-service, and man-in-the-middle) and vulnerabilities that occur irrespective of protocol. They analyze IEEE 802.11, IEEE 802.15.4 (e.g., WirelessHART, ZigBee, ISA100-11a, and Bluetooth), and proprietary protocols. However, given the closed-nature of proprietary protocols and few security research publications, Reaves et al. are unable to conduct a thorough analysis [2].

Of the numerous commercially available proprietary 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 [6] have significant growth potential in WSNs. The most common implementation of the ITU-T G.9959 recommendation is commercially known as Z-Wave, marketed by the Z-Wave Alliance. The alliance includes more than 300 companies, such as ADT, Ingersoll-Rand, LG Electronics, and Verizon. However, the Z-Wave Alliance requires developers to sign nondisclosure and confidentiality agreements, preventing open source research.

Given the lack of security analyses of the Z-Wave protocol and its implementation, the strengths and weaknesses are not well known. In order to discover vulnerabilities, it is necessary to analyze a variety of vendor implementations.

Previous approaches to uncovering Z-Wave network vulnerabilities focus on identifying points of compromise at its physical layer [7-8]. Meanwhile, minimal research has been done to exploit the Z-Wave network when controlled by a globally accessible gateway. Our approach is an exploration of vulnerabilities that exist in Z-Wave networks where devices are managed by an Internet accessible gateway.

This work begins with an evaluation of security features present on three commercially available Z-Wave gateways, sensors, and secondary controllers. We find that all Z-Wave gateway devices are susceptible to attacks once an attacker compromises the WLAN through physical access, lack of network authentication, or network key compromise. We then experimentally evaluate the even greater security implications of a Z-Wave gateway connected to the Internet. Once the Z-Wave gateway is compromised, point-to-point message encryption to door locks and other secure end devices becomes irrelevant.

In the following section we give a brief background of the Z-Wave protocol and home automation networks and discuss related work in the area of Z-Wave network vulnerabilities. Section III outlines our methodology for evaluating the security features of each gateway device and presents a newly discovered vulnerability, rogue controller injection. In Section IV, we propose mitigation strategies to secure a Z-Wave network and conclude our findings in Section V.

II. Background and Related Work

A. The Z-Wave Protocol

All Z-Wave networks adhere to the ITU-T G.9959 physical (PHY) and medium access control (MAC) layer specification ensuring interoperability between vendor devices, but differentiate based on the network and application layers (Fig. 1). ITU-T G.9959-based networks operate in the industrial, scientific, and medical (ISM) bands (e.g., 908.4 MHz in North America, 860.4 MHz in Europe, and additional frequencies in other

978-1-4673-6773-8/15/$31.00 ©2015 IEEE

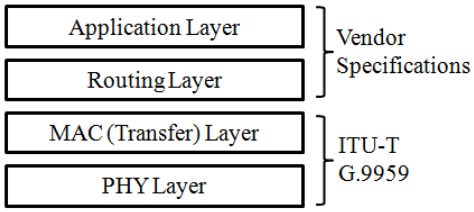

Fig. 1: Z-Wave/ITU-T G.9959 Protocol Stack. The MAC and PHY Layers are specified in the ITU-T G.9959 Recommendation whereas the application and routing layer are specific to vendor implementation.

geographic locations). There are two node types in Z-Wave networks: control nodes (gateways); and slave nodes. Control nodes send commands and request updates and slave nodes responds to commands. As a meshed topology network, slave nodes also forward commands to other nodes not directly reachable by the control node. Message forwarding has a hop limit of 4 nodes and a maximum of 232 nodes are allowed.

To allow for network overlap, each Z-Wave network has a unique 32-bit Home ID specified in the controller. Z-Wave networks can have multiple control nodes but only one of them can be the primary control.

Of note, the Z-Wave protocol supports the Advanced encryption Standard with 128-bit keys. Although supported, encryption is not required. Implementing encryption is left up to the vendor who decides if node communication is sensitive enough to warrant its use. In the low-rate, low-power WSNs, memory and power are limited, discouraging vendors from adding features unless necessary [9]. Surveys of similar protocols have shown that the use of encryption is not universal [10].

B. Home Automation

Many Home Automation Networks (HANs) consist of wireless sensors that exchange control and information messages [11]. Device types include door locks, security sensors, alarms, environmental controls, light modules, and motion sensors (Fig. 2). The user either manages the HAN with a Z-Wave controller from the home (Fig. 3) or a Z-Wave gateway that can be managed either locally or globally (Fig. 4).

Managing the Z-Wave HAN locally lessens the features and span of network control. Being able to only manage the network from within the home does not provide the user with the ability to control locks and alarms while beyond the controller radio frequency transmission range. However, managing the network through a globally connected gateway provides the user with the ability to control their network with a mobile device from anywhere. If a user forgets to lock their door or arm the alarm system when they leave the house, she can login to the gateway and configure devices. With this accessibility comes added vulnerabilities. Not only does the user have access to their Z-Wave HAN, but anyone that can compromise their WLAN defense can also garner access.

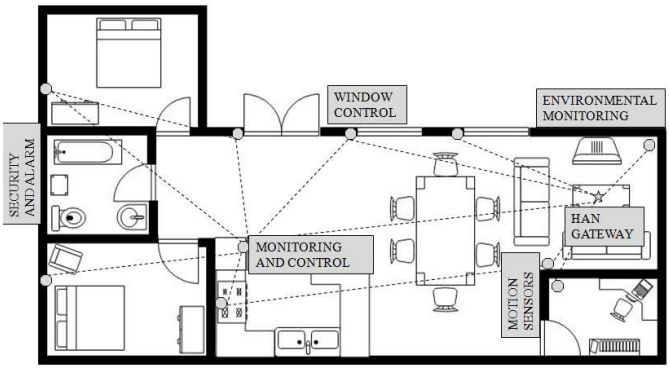

Fig. 2: Home automation network model: Multiple Z-Wave devices communicating with each other and controlled by the Z-Wave gateway.

As reported in [12], 73% of households with Internet access have a wireless local area network (WLAN). Studies show that between 45% and 66% of households in the US have unsecured WLANs [13-14]. Although these studies were not global, it can be inferred that a large percentage of WLANs are unsecured. A survey of IEEE 802.15.4 Wireless Personal Area Networks in ten US cities found similar percentages of unsecured networks [10]. Even if secured, there are weaknesses and proven exploits against wireless protocols [15-17].

C. Related Work

There are published analyses of note that demonstrate the exploitation of Z-Wave HANs. The authors of [7] develop a sniffer project known as *z-force*. Their sniffer consists of two Texas Instruments CC1110 boards, one transmits and receives, and custom firmware. Before implementing the sniffer, the authors conduct an in-depth study of the Z-Wave protocol and uncover the details of frame encryption and authentication algorithms. They discover an implementation error used in

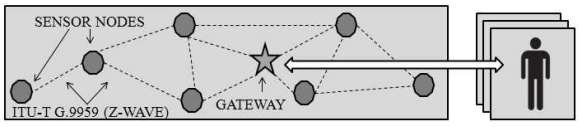

Fig. 3: Local gateway access: Users manage their Z-Wave HAN via their gateway locally. All device control is performed from inside the home.

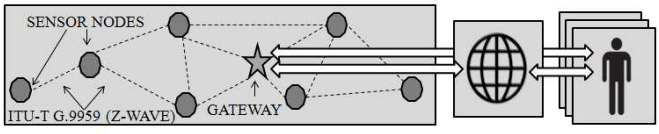

Fig. 4: Global gateway access: Users manage their Z-Wave HAN via their gateway locally or globally. Z-Wave devices can be controlled using mobile applications or any Internet connected device.

a Z-Wave compliant door lock that allows them to reset the established network key. The z-force tool allows them to capture the `homeID` (network ID) of the controller and `nodeID` (device ID) of the device, reset the network key to a different value, and inject unauthorized commands into the Z-Wave HAN. As a result, they are able to open and close the lock by bypassing the controller. The z-force tool has only been demonstrated to work on the 860.4 MHz European frequency and its closed-source nature makes it difficult to implement or extend.

Another analysis of Z-Wave is the *Scapy-Radio* project [8]. Scapy-Radio combines Scapy and gnuRadio to capture traffic and replay packets back into the Z-Wave HAN. To test their device, the authors implement a Z-Wave HAN using a Raspberry Pi and an Aeon Labs Z-Stick programmed using the open source OpenZWave software. They include two Z-Wave devices into the network: an alarm device and a motion sensor. Capturing the network traffic, the authors record and analyze packets, noticing that when the motion sensor was activated, it sent a packet to the controller. Upon receipt, the controller sent a packet to the alarm device with a command to activate. After multiple improvements to their Scapy-Radio, they are now able to detect `ON` commands from the controller to the alarm device and subsequently inject `OFF` commands to the alarm effectively denying service to the device.

The authors of [18] demonstrate the exploitation of several home automation gateways, including the VeraLite Smart Home Controller. The authors find several vulnerabilities that expose sensitive information from the VeraLite while allowing an attacker to fully control devices on the network. Finding insufficient authentication checks, the authors expose a universal plug and play functionality that allows an attacker to execute Lua code as `root` user. They exploit this vulnerability and successfully create a backdoor account on the device. Another vulnerability found was that the VeraLite does not protect against Cross-Site Request Forgery (CSRF). It is therefore possible for an attacker to update the device firmware with their own malicious firmware modifying the operation of the device. It has been shown that remote modification of firmware can severely affect a device and expose confidential information [19].

Oluwafemi et al. [20] investigate the feasibility of causing physical harm to home automation users through the explosion of compact fluorescent lamps (CFLs). Four distinct electronic signals are transmitted to CFLs connected to a Z-Wave device until the CFLs emit a visual or auditory spark or fail completely. Although the authors conclude that harming individuals via their attack vector is difficult, they observe that non-networked devices, such as CFLs, might possibly be connected to networked devices and therefore can be compromised remotely.

Most recently, Barcena et al. [21] discuss multiple insecurities in home automation networks. During their research the authors poison the Address Resolution Protocol on the gateway to redirect firmware update request to their own server. After modifying the firmware, the gateway receives the malicious firmware as a legitimate update and the authors are given full control over the gateway. Barcena et al. are also able directly access the controller application since the device does not require user authentication.

These early studies illustrate the possible attacks resulting from wireless traffic injection or compromise of the Z-Wave gateway.

III. Methodology

There are three phases to this work. First, is the initial reconnaissance of each device, where the default settings and modes of operations are identified. The second phase examines vulnerabilities in the device implementation that enables attackers to take control of the Z-Wave HAN. Phase three exploits a new vulnerability that allows an attacker to create a persistent attack channel by injecting a rogue controller into the Z-Wave HAN.

The equipment used to conduct vulnerability scanning and exploitation are an HP EliteBook 8570w with Kali Linux operating system, Alfa AWUS036H Card, and an Aspire RF Wireless controller (Fig. 5). Software tools used include *aircrack-ng* to defeat WLAN defense, *Zenmap* to scan the WLAN for Z-Wave gateways, *Burp Proxy* to capture and inject packets, and *Putty* to gain backend access via SSH.

A. Reconnaissance

The three Z-Wave gateways under test are the Raspberry Pi with RaZberry Pi General-Purpose Input/Output Daughter Card (henceforth referred to as RaZberry Pi), VeraEdge Home Controller, and the Almond+, which was released in early 2015 (Fig. 6). We build real-world Z-Wave HANs using each gateway, each consisting of a smart switch, a light module, a door lock, and a water valve (Fig. 7) that are dispersed

Fig. 5: Equipment used for vulnerability scanning and exploitation: (a) HP Elitebook 8570w, (b) Alfa AWUS036H Card, and (c) Aspire RF Wireless controller.

TABLE I: Z-Wave Gateway Default Configurations

Gateway Device	Web UI Enabled	SSH Enabled	Web UI Authentication
RaZberry Pi	Yes	No	No
VeraEdge	Yes	Yes	No
Almond+	No	No	Yes

TABLE II: Z-Wave Gateway Fingerprinting

Gateway Device	Open Ports	Unique Feature
RaZberry Pi	22*, 443 8083, 8084	**MAC address:** (Raspberry Pi Foundation)
VeraEdge	80, 22, 53 49451	**OS:** OpenWrt Barrier Breaker
Almond+	80, 22*, 8200**	**OS:** OpenWrt Kamikaze Backfire

*Visible if SSH is enabled.
**Visible if UPnP is enabled.

throughout a house. It is apparent that each gateway device is configured with varying default settings (Table I). The RaZberry Pi and VeraEdge require that the web user interface (UI) be enabled since that is the only way to interact with either. On the other hand, the Almond+ has a touch screen that allows the user to configure all the settings without a web UI. However, as mentioned, in order for a user to access any HAN globally, a web UI is needed.

We enable the Almond+ UI for this experiment to replicate how an actual Almond+ HAN with Internet accessibility would function. We now have three operational web UIs on all devices. The UIs allow us to access the gateways locally by navigating to the IP address of the device or globally through a mobile device application or another Internet connected computer.

As an attacker, the first step of reconnaissance is locating Z-Wave HANs. A likely approach is to use a Z-Wave sniffer [8,22] to conduct Z-Wave warwalking similar to what is done in [23-24]. Next, the attacker gains access to the WLAN by either associating with an unsecured network or using one of the many freely available tools to penetrate the WLAN defense. As previously discussed, the likelihood of an unsecured or poorly secured WLAN is relatively high. Furthermore, the likelihood of an attacker gaining access to the target WLAN is proportional to its insecurity.

After gaining access to the target network, we perform a network scan using *Zenmap* to locate the IP address of the target gateway device. However, before doing so we ensure we know unique fingerprints to identify each device (Table II). If an attacker prefers stealth over speed the fingerprints can be used to reduce the search space, limiting the amount of traffic on the network. This tactic creates less "noise" and improves the chance that the attack goes unnoticed. After locating the devices, we are able to navigate to the UIs. The RaZperry Pi Developer Documentation, freely available on their website [25], states that the UI is located at `http://x.x.x.x:8083` (`x.x.x.x` denotes the IP address of the RaZberry Pi), which confirms our Zenmap scan results (Fig. 8). The UIs for the VeraEdge and Almond+ are located at Uniform Resource Locator `http://y.y.y.y` or `http://z.z.z.z`, where `y.y.y.y` denotes the IP address of the VeraEdge and `z.z.z.z` denotes the IP address of the Almond+.

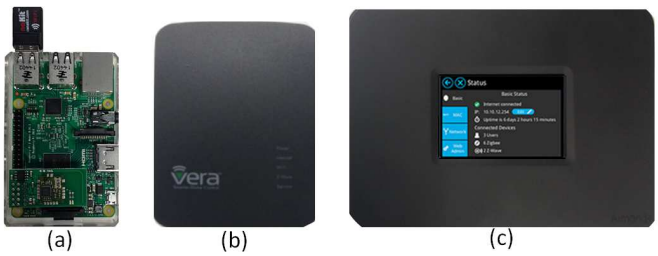

Fig. 6: Gateways devices: (a) RaZberry Pi, (b) VeraEdge, and (c) Almond+.

After navigating to each UI, we have access to the Z-Wave HAN for each gateway and proceed to explore all three UIs to discover potential vulnerabilities.

We discover that it is possible to actuate all Z-Wave devices. This approach seems somewhat naive. If an attacker has access to the UI simultaneously with the user, their attack will be discovered and the owner will take immediate action. We explore additional options to remain inconspicuous while compromising the Z-Wave HAN.

B. Gateway Vulnerabilities

During the reconnaissance, we observe that all gateway devices use HTTP `POST` and HTTP `GET` requests to send commands to their server, which in turn relays information to the embedded Z-Wave chip that transmits wireless packets. We are able to capture the request packets and replay them to the server using Burp Proxy. The packets are accepted as legitimate requests as if from the UI. Modifying the packets before retransmission allows us to actuate all the devices on the network. For instance, capturing requests sent to the smart switch could be easily modified to trigger a motion detector or open a water valve. Some Z-Wave devices (e.g., door locks) use encryption to transmit packets and receive packets from their controller. However, since the gateway controller is compromised, any packets we capture from the controller and replay or inject will be accepted actuating any device despite their use of encryption. Similar attacks are demonstrated in [18, 20, 21]. UI authentication credentials for the Almond+ are also captured and give us full access to the gateway. Some vendors have taken action to fix discovered vulnerabilities [21], but multiple devices can still be exploited.

(a) (b) (c) (d)

Fig. 7: Z-Wave devices that were connected to each gateway network: (a) smart switch, (b) light module (c) door lock, and (d) water valve.

For the first time, we demonstrate exploits, as found in [18] on the VeraEdge. The authors of [18] are able to successfully retrieve the VeraLite (same manufacturer of the VeraEdge) backup file for the entire system. The backup file is used in case of a system malfunction. The user can reimage the system with the backup file to restore it to the original user settings. An attacker can use the backup file to retrieve passwords and sensitive system information that would aide their exploitation of the target network. After navigating to `http://y.y.y.y/cgi-bin/cmh/backup.sh?external=1`, the backup file is successfully downloaded from the VeraEdge. The WLAN password for the VeraEdge is located in the backup file and stored in plain text. By default, the WLAN password is also the root password for SSH access into the VeraEdge. Since SSH is enabled (Table II), we use *Putty* to login to the backend of the VeraEdge. We also successfully replicate all other exploits demonstrated in [18] on the VeraEdge. This confirms the hypothesis that some manufactures either cannot or will not find solutions to existing vulnerabilities and that many of the same vulnerabilities exist in devices from the same manufacturer. We also find that we could apply this exploit to the RaZberry Pi. By navigating to `http://x.x.x.x:8083/ZWaveAPI/Backup`, the attacker will download the backup files that contain Z-Wave device information. This is useful if the attacker wants to determine what types of devices are connected to the target Z-Wave HAN.

From testing exploits on the RaZberry Pi, VeraEdge, and Almond+ that Crowley et al. demonstrate on their VeraLite [18], it is evident that an attacker will need to exercise some caution while manipulating the gateway UI. If the user becomes suspicious of an attack, they can check the gateway log file that exists on all devices under test. The log file keeps a record of all actions on the network. If an attacker attempts to perform an unauthorized firmware update [18, 21], exploit insufficient authentication checks on the gateway [18], or other gateway attacks [20-21], all of the activity will be recorded in the log file. Every time the attacker takes an action, the log file will capture the event. Once confirming their suspicion, the user will take action and possibly disconnect the gateway from the Internet or disable it completely. Interestingly, we find that there is a way to gain access to a Z-Wave gateway

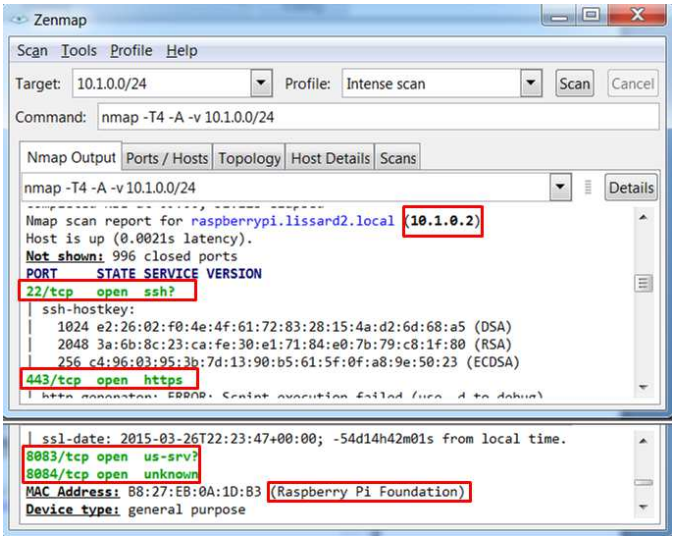

Fig. 8: Zenmap Scan showing RaZberry Pi fingerprints and the IP address.

and create a persistent connection to all Z-Wave devices while creating minimal log entries.

C. Rogue Controller Injection

In a Z-Wave HAN, if the user wants to add a new device to their network, the controller must be placed in *inclusion* mode. Inclusion occurs when the primary controller (gateway device) in the Z-Wave HAN includes other devices in the network by assigning them its `homeID` (network identifier). When the network owner sets the device they intend to add in inclusion mode, the device accepts the `homeID` of the controller and a new `nodeID` and joins the HAN. In Z-Wave HANs, the user must physically activate the inclusion mode on devices to add it to a network. However, there are two ways to put the controller into inclusion mode. The Almond+ requires the user to physically put the controller into inclusion mode, whereas the user can place the RaZberry Pi and VeraEdge Controller in inclusion mode from the UI.

We viewthis as a major vulnerability. This feature allows an attacker to gain access to the WLAN once and make a copy of the Z-Wave HAN configuration, including `homeID` and `nodeIDs`. Doing so will allow stealthy persistent access and full control of devices even if the user decides to remove their gateway from the Internet.

To test this hypothesis, we put the VeraEdge into inclusion mode by injecting a previously captured HTTP packet creating one log entry. Once in inclusion mode, we use an Aspire RF Wireless controller and set it to *Replicate* mode. Replicate mode allows one controller to copy information from another controller. The Aspire RF Wireless controller automatically connects to the gateway and, since the gateway is in inclusion mode, it transfers all of its information to the Aspire RF Wireless controller creating a second log entry. The device settings on the Aspire RF Wireless controller now lists many of the devices of the target network. After adding the rogue

controller to the network, the UI displays the newly added device. We navigate to the UI and delete the rogue controller to cover our tracks so it is not visible to the home owner. This creates a third log entry. Deleting the rogue controller from the HAN does not affect its access since the VeraEdge does not validate device deletion or exclusion. We still maintain full control of most devices while only creating three log entries.

We also attempt this exploit on the RaZberry Pi. After injecting an inclusion packet to the gateway, an attempt to replicate the Z-Wave HAN configuration is made. It successfully connects to the RaZberry Pi and replicates the entire Z-Wave HAN configuration. We use the rogue controller to actuate devices on the target network. We attempt to remove the rogue controller from the UI, but are unsuccessful. The RaZberry Pi, unlike the VeraEdge, validates that device deletion or exclusion occurs. Therefore, a device can only be removed from the UI if it is actually removed from the Z-Wave HAN. We found a way to circumvent this RaZberry Pi feature. For our second attempt, we send a inclusion packet to the gateway. Once in inclusion mode, we activate the inclusion button on a secondary rogue controller, then immediately activate replicate mode on the primary rogue controller. While the secondary rogue controller is being included in the Z-Wave HAN, the primary rogue controller sniffs all traffic between the gateway device and the secondary rogue controller. Therefore, the primary rogue controller is never actually added to the Z-Wave HAN; it only captures all its network information. We then permanently exclude the secondary rogue controller from the RaZberry Pi which removes it from the UI. The primary rogue controller has full access to most devices on the Z-Wave HAN.

The new rogue controller only communicates in the sub-GHz spectrum and not via HTTP requests. Any communication between the new controller and the Z-Wave devices goes undetected by both the HAN gateway and legitimate users. Unlike previous exploits, even if the gateway looses power, Internet connection, or the user willingly disconnects it from the WLAN, the rogue Aspire RF Wireless controller has persistent access to the Z-Wave devices (Fig. 9).

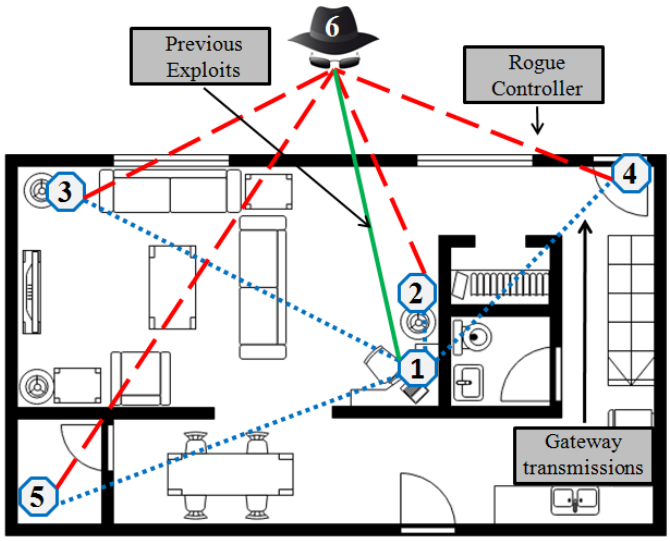

Fig. 9: Actual test environment for the rogue controller exploitation. (1) through (6) represent Z-Wave devices. Unlike previous exploits, the rogue controller transmits commands directly to Z-Wave devices, bypassing the gateway.

IV. Mitigation Strategies

In this section, we propose a checklist that will aide users in securing their Z-Wave HANs.

A. Hide WLAN SSID

The first step in exploiting the HAN is gaining access to the target WLAN. Hiding the service set identifier (SSID) prevents attackers using passive network scans to locate the WLAN. This option can be configured in the wireless access point settings. If an attacker conducts Z-Wave warwalking and locates a Z-Wave network, but is not able to identify the SSID, she will not be able authenticate to the WLAN and attack the globally accessible Z-Wave gateway.

B. WLAN security and choose a robust password

Some WLANs are left unsecured, granting attackers uninhibited access. Secondly, some WLANs use WEP to secure their network because they require backwards compatibility with legacy devices or the user is not aware of existing vulnerabilities. If possible, WPA2 should be used in place as WEP is easily exploitable. A 128-bit WEP key can be cracked in just three minutes with freely available tools such as aircrack-ng [26]. Weak passwords should be avoided because numerous free dictionaries containing precomputed passwords can be found on the Internet. An attacker armed with an Alfa AWUS036H Card, aircrack-ng, and a large password dictionary can crack weak WPA2 passwords. Although longer and more complex passwords are breakable, they take significantly longer to crack than weak passwords. A simple calculation can be used to estimate the maximum time required to brute force crack passwords (1).

$$timetaken(seconds) = \frac{combinations}{guesses/second} \quad (1)$$

Using case-sensitive passwords consisting of 94 available characters, Table III lists the maximum cracking times, assuming the computer calculates a modest 5,000,000 guesses per second. Password dictionaries can significantly reduce the time taken to crack passwords, so users should choose passwords that are as long and complex as practicable.

C. MAC address filtering

MAC address filtering allows the WLAN administrator to select specific devices that are trusted on the WLAN. Although the attacker can spoof MAC addresses, this is one more barrier they will have to breach in order to compromise the WLAN security. This can also be configured in the wireless access point settings. If MAC address filtering is enabled, an attacker without knowledge of spoofing MAC addresses will not be

able to authenticate to the WLAN, even if armed with the WLAN access key.

D. Enable UI authentication on the Z-Wave gateway

Not all Z-Wave gateways have UI authentication. However, if UI authentication is available, the user must immediately enable it and create a strong password. Many of the vulnerabilities exploited in [18] do not work on the recently released Almond+. With mandatory UI authentication, an attacker would have to intercept HTTP packets while the user is logging in to capture credentials. UI authentication is clearly not enough, but it encumbers an attacker. Even though it is possible to capture the Almond+ UI authentication credentials, the exploits available are limited because the Almond+ architecture provides extra security. Not being able to put the Almond+ into inclusion mode from the UI prevents the injection of a rogue controller into the network.

E. Use a Reverse Proxy Server

A Reverse Proxy Server (RPS) is an intermediary application between the user and the gateway. It provides an added layer of security by intercepting packets sent to the gateway and performs additional authentication before allowing the user to access the gateway. This is a effective option for gateways that do not have UI authentication and provides increased UI security for those that already have authentication. RPS tools such as *NGINX* are free and available for download.

F. Disable unused network services

Gateway administrators should also ensure that any unused network services are disabled. If there is not a need to have continual access to the backend of the gateway, SSH should be disabled when not in use. SSH is enabled by default on the VeraEdge. As mentioned, after downloading the backup file and locating the WLAN password, we successfully log into the backend of the VeraEdge using *Putty* with username *root* and password *<wlan_password>*. This vulnerability can be avoided if SSH is disabled.

TABLE III: Brute Force Password Cracking

Characters	**Combinations**	**Time taken @ 5×10^6 guesses/sec**
1	94	0.00002 seconds
2	$8,836$	0.00177 seconds
3	$830,534$	0.16612 seconds
4	$78,074,896$	15.6145 seconds
5	7.3×10^9	25 minutes
6	6.9×10^{11}	38 hours
7	6.4×10^{13}	150 days
8	6×10^{15}	38 years

G. Inspect log files

Although it is possible for an attacker to clear log files to cover their tracks after compromising the Z-Wave HAN, it is still worth inspecting the log files for unusual activity. Any actuation of devices at abnormal hours and failed login attempts will be recorded in the log. If this is found, the user should reset their gateway and change their passwords.

Applying these mitigation strategies will not only deter an attacker but decrease the attack surface, lessening the chance of a successful exploitation of the Z-Wave HAN.

H. Enable End-Device Encryption

When a rogue controller is added to the network, it can control most devices except those that use encryption. When a device that uses encryption is included into the Z-Wave network, the device sends its encryption key to the gateway. A chain of trust is then established between those two devices. When a rogue controller is added, the chain-of-trust does not extend to the rogue controller because the end device is not aware of its inclusion. Some Z-Wave devices, including Gen5 models, come with the option to enable encryption with the press of a button. Enabling end-device encryption may disallow rogue controller access.

V. Conclusion

Given the growing popularity of Z-Wave, vendors are continuously introducing new gateway devices. Many consumers are unaware of the lack of security in gateway devices and the vulnerabilities therein. The ability for unauthorized individuals to access networks and take control of Z-Wave devices poses a serious threat to homes, office buildings, and factories that use WSNs for control and communications. A security aware user may discover that their gateway has been compromised if subjected to gateway exploitation or UI attacks. Of even greater concern, explained here for the first time, rogue controllers pose a significant threat because all transmissions to Z-Wave devices go unnoticed by the victim. Consumers should follow the mitigation steps in Section IV to strengthen their security posture and prevent their WLAN and Z-Wave HAN from being compromised.

VI. Acknowledgment

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

[1] L. Hormann, P. Glatz, C. Steger, and R. Weiss, "Measure the State-of-Charge - Analysis and Impact on Wireless Sensor Networks," *6th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, pp. 982-985, Oct 2011.

[2] B. Reaves and T. Morris, "Analysis and Mitigation of Vulnerabilities in Short-Range Wireless Communications for Industrial Control Systems," *International Journal of Critical Infrastructure Protection*, vol. 5, pp. 154-174, Dec 2012.

[3] E. Lee, "Cyber Physical Systems: Design Challenges," *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pp.363-369, May 2008

[4] T. McCourt, S. Leopold, and F. Louthan, "The Internet of Things: A Study in the Hype, Reality, Disruption, and Growth," Raymond James Technology & Communications Industry Report, Jan 2014.

[5] B. Ramsey and B. Mullins , "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, pp 63-79, Mar 2013.

[6] Recommendation ITU-T G.9959, "Short Range Narrow-Band Digital Radiocommunication Transceivers - PHY and MAC Layer Specifications," Jan 2015.

[7] B. Fouladi and S. Ghanoun, "Security Evaluation of the Z-Wave Wireless Protocol," *Black Hat Conference*, Las Vegas, NV, Jul 2013.

[8] J. Picod, A. Lebrun, and J. Demay, "Bringing Software Defined Radio to the Penetration Testing Community," *Black Hat Conference*, Las Vegas, NV, Aug 2014.

[9] *Freescale Beestack: Application Development Guide*, Freescale Semiconductor, Chandler, AZ, Jan 2008.

[10] B. Ramsey, B. Mullins, R. Speers, and A. Batterton, "Watching for Weaknesses in Wild WPANs," *IEEE Military Communications Conference (MILCOM)*, pp. 1404-1409, Nov 2013.

[11] N. Langhammer and R. Kays, "Performance Evaluation of Wireless Home Automation Networks in Indoor Scenarios," *IEEE Transactions on Smart Grid*, vol.3, no.4, pp. 2252-2261, Dec 2012.

[12] D. Watkins, "Broadband and Wi-Fi Households Global Forecast 2012," https://www.strategyanalytics.com, Mar 2012.

[13] K. Jones, L. Liu, "What Where Wi: An Analysis of Millions of Wi-Fi Access Points," *IEEE International Conference on Portable Information Devices*, pp. 1-6, May 2007.

[14] A. Zafft and E. Agu, "Malicious WiFi Networks: A First Look," *37th IEEE Conference on Local Computer Networks (LCN)*, pp. 1038-1043, Oct 2012.

[15] H. Lane, "Security Vulnerabilities and Wireless LAN Technology," SANs Institute InfoSec Reading Room, Feb 2005.

[16] S. Fahmy, A. Nasir, and N. Shamsuddin, "Wireless Network Attack: Raising the Awareness of Kampung WiFi Residents," *International Conference on Computer and Information Sciences (ICCIS)*, pp. 736-740, Jun 2012.

[17] M. Ahmad, "WPA Too!," *DEF CON*, vol. 18, Aug 2010. Available: http://www.defcon.org/images/defcon-18/dc-18-presentations/Ahmad/DEFCON-18-Ahmad-WPA-Too-WP.pdf.

[18] D. Crowley, D. Bryan, and J. Savage, "Home Invasion V2.0 - Attacking Network-Controlled Hardware," *Black Hat Conference*, Las Vegas, NV, Jul 2013.

[19] A. Cui and S. Stolfo, "Print Me If You Dare: Firmware Modification Attacks and the Rise of Printer Malware," in the 28th Chaos Communication Congress, Dec 2011.

[20] T. Oluwafemi, S. Grupta, S. Patel, and T. Kohno, "Experimental Security Analyses of Non-Networked Compact Fluorescent Lamps: A Case Study of Home Automation Security," *2013 Workshop on Learning from Authoritative Security Experiment Results*, pp. 13-14, Arlington, VA, Oct 2013.

[21] M. Barcena and C. Wueest, "Insecurity in the Internet of Things," https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/insecurity-in-the-internet-of-things.pdf, Mar 2015.

[22] C. Badenhop, J. Fuller, J. Hall, B. Ramsey, and M. Rice, "Evaluating ITU-T G.9959 Based Wireless Systems used in Critical Infrastructure Assets," in *Critical Infrastructure Protection IX*, S. Shenoi and M. Rice, Eds. Heidelberg, Germany: Springer, 2015, to be published.

[23] B. Ramsey, B. Mullins, and E. White, "Improved Tools for ZigBee WarWalking," *7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, pp. 921-924, Oct 2012.

[24] C. Kiraly and P. Picco, "Where's the mote? Ask the MoteHunter!," *7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, pp. 982-990, Oct 2012.

[25] The RaZberry Project (n.d.) [Online]. Available: razberry.z-wave.me

[26] S. Reddy, K. Ramani, K. Rijutha, S. Ali, and P. Reddy, "Wireless Hacking - A WiFi Hack by Cracking WEP," *2nd International Conference on Education Technology and Computer (ICETC)*, pp. 189-193, Jun 2010.