Security analysis of WirelessHART communication scheme

Lyes Bayou, David Espes, Nora Cuppens, Frédéric Cuppens

Security analysis of WirelessHART communication scheme

Lyes Bayou[1], David Espes[2], Nora Cuppens-Boulahia[1], and Frédéric Cuppens[1]

[1] Télécom Bretagne-LabSTICC, 2 Rue de la Châtaigneraie, Césson Sévigné, France
[2] University of Western Brittany - LabSTICC, Brest,France

Abstract. Communication security is a major concern in industrial process management. Indeed, in addition to real-time requirements, it is very important to ensure that sensing data sent by field sensors are not altered or modified during their transmission. This is more true in Wireless Sensor Networks where communication can be hijacked and false data injected. Therefore wireless communication protocols include several security mechanisms to ensure data confidentiality and integrity. In this paper, we present an attack against WirelessHART, the leading wireless communication protocol in industrial environment. We show that an insider attacker can bypass security mechanisms and inject false commands in the network. Such attacks can have harmful economical consequences or even more can threaten human lives. We propose also some solutions that can be applied for detecting and mitigating this kind of attacks.

1 Introduction

Industrial Control Systems (ICS) are computed-based systems used for monitoring and managing industrial installations and facilities. We can find such systems in airports, power plants, gas refineries, etc. The architecture of these systems relies on several sensors and actuators deployed throughout the industrial installation. Sensors are responsible for gathering different kinds of information about the industrial process such as temperature, pressure, flow, etc. These information are sent to a controller that processes them and sends back commands to actuators. As results, an actuator can for example open a valve to increase the flow of a chemical component or stop a pump when the oil tank is filled.

The security in Industrial Control Systems is a major concern. Indeed, these systems manage installations that play an important economical role. Even more, targeting these systems can lead not only to economical losses but can also threaten human lives [1].

Therefore and as these systems depend on sensing data, it is important to secure communication channels between these sensors and the main controllers. This issue is more challenging in Wireless Sensor Networks as the use of wireless communications brings its own security weaknesses.

Based on the analysis of the communication scheme, we present in this paper an attack against WirelessHART [2], the leading wireless communication protocol in the industrial environment. We show that although this protocol implements several mechanisms to ensure the integrity and confidentiality of exchanged data, an insider attacker can use its own credential to bypass security mechanisms and inject false commands in the network. Using this weakness, we describe three scenarios that can be used to launch an attack against a WSN. Such attacks can have harmful economical consequences or even more can threaten human lives.

Several tests were conducted on a simulated network to prove the feasibility of these attacks and to assess its potential impact on the functioning of the industrial process.

The rest of the paper is organized as follows. In Section 2, we give a brief description of the functioning of a WirelessHART network, its communication scheme and how data are exchanged and secured. We detail in Section 3, the functioning of the broadcast attack and give three different scenarios that use this attack. Section 4 presents results of the three scenarios on a simulated WSN. Some countermeasures that can be used to detect such attacks are discussed in Section 5. In Section 6, we discuss prior works on the security of WirelessHART. Finally, Section 7 presents the conclusion and future works.

2 Background

WirelessHART [2] is the first standardized wireless communication protocol specially developed for industrial process management. It uses a time-synchronized, self-organized and self-healing mesh architecture to provide a reliable and real-time communication. It is included in version 7 of the HART standard, released in 2007, and was approved as a IEC 62591 standard in 2010.

2.1 Topology of a WirelessHART network

A typical WirelessHART network is composed of the following devices:

- A Gateway that connects the wireless network to the plant automation network, allowing data to flow between the two networks. It can also be used to convert data and commands from one protocol to another one;
- A Network Manager that is responsible for the overall management, scheduling, and optimization of the wireless network. It generates and maintains all of the routing information and also allocates communication resources;
- A Security Manager that is responsible for the generation, storage, and management of cryptographic keys;
- Access Points that connect the Gateway to the wireless network through a wired connection;
- Field devices deployed in the plant field and which can be sensors or actuators;

- Routers used for forwarding packets from one network device to another;
- Handheld devices that are portable equipments operated by the plant personnel used in the installation and during the maintenance of network devices.

2.2 WirelessHART stack

The WirelessHART protocol is composed of 4 layers. It is based in its physical layer upon the IEEE 802.15.4 standard [3]. It defines its own data link layer and network layer and shares the same application layer with the wired HART protocol (in the addition of wireless commands). A brief description of each layer is given below:

- Application Layer (AL): it is a command based layer. It is used to send sensing data from field devices to the Network Manager, and to send commands from the Network Manager to the field devices. It supports both common HART commands (inherited from the wired version) and WirelessHART commands.
- Transport Layer (TL): it provides mechanisms to ensure packets fragmentation and defragmentation. It ensures data delivery without loss, duplication or misordering to its final destination. It supports acknowledged and unacknowledged transactions.
- Network Layer (NL): it ensures end-to-end integrity and confidentiality. It provides routing features. It receives packets from the DLL and checks if they have to be transmitted to the AL or have to be resent to the DLL to be forwarded to the next device.
- Data Link Layer (DLL): it is responsible of preparing packets for transmission, sending and receiving packets, managing time slots and maintaining informations about neighborhood. It provides hop-by-hop authentication.
- Physical Layer (PhL) : it is based on the IEEE 802.15.4-2006 standard and operates in the 2.4 GHz. It is responsible of wireless transmission and reception.

2.3 WirlessHART Communication

The Network Manager is one of the most important devices in a WirelessHART network. It is responsible for the overall management, scheduling, and optimization of the wireless network. It generates and maintains graphs and routing information and also allocates communication resources.

Communication type In WirelessHART there are 05 packet types, called Data Link Protocol Unit (DLPDU), that can be exchanged between devices:

- Data DLPDU: encapsulates packets from the NL. It is used to exchange sensing data and AL commands.
- Ack DLPDU: is used by a device that receives an unicast packet, to send back to the sender device an acknowledgment of the reception of that packet.

- Keep-alive DLPDU: is used by a device that spends a defined time without sending any packets, to inform its neighbors that it is still active.
- Advertise DLPDU: is used for providing information to neighboring devices trying to join the network;
- Disconnect DLPDU: is used by a device to inform its neighboring devices that it is leaving the network.

Ack, Advertise, Keep-Alive and Disconnect DLPDUs are generated and processed in the Data Link Layer and are not propagated to the network layer or forwarded through the network. This means that these DLPDUs are only used in local communication between neighbors. The Data DLPDU is the only kind of packet that is transmitted in an end-to-end communication. During the transmission, data fields in the payload are enciphered.

Communication scheduling To provide reliable and collision free communication, WirelessHART uses *Time Division Multiple Access* (TDMA) and *Channel hopping* to control the access to the wireless medium. The time is divided in consecutive periods of the same duration called slots. Each communication between two devices occurs in one slot of 10 ms. Superframes are collection of slots repeated continuously with a fixed repetition rate.

Typically, two devices are assigned to one time slot (one as the sender and a second as the receiver). Only one packet is transmitted in one slot from the sender to the receiver which has to reply with an acknowledgment packet in the same slot. In the case of a broadcast message, there is one sender and multiple receivers and the message is not acknowledged.

In addition to the TDMA, WirelessHART uses channel hopping to provide frequency diversity and avoid interferences. Thus, the 2.4 GHz band is divided into 16 channels numbered from 11 to 26 which provide up to 15 communications in the same slot (Channel 26 is not used).

Communication routing WirelessHART implements in the Network Layer, two methods of routing packets throughout the network, i.e., graph routing and source routing.

- Graph routing: a graph is a collection of directed paths that connect network devices. It is build by the Network manager based on its knowledge of the network topology and connectivity. Every graph has a unique graph identifier that is inserted in the network packet header. Each device receiving this packet, must forward it to the next hop belonging to that graph. This routing method is used for normal communications, in both upstream (from a device to the network manager) and downstream (from the network manager to a specific device) directions.
- Source routing: it is a single directed route between a source and a destination device. The complete route is completely inserted in the network packet header by the sender device. Each intermediate device propagates the packet to the next device indicated in the source route field. This method of routing is used only for testing routes, troubleshooting network paths or for ad-hoc communications.

Communication security WirelessHART implements several mechanisms to ensure data confidentiality, authenticity and integrity in both hop-by-hop and end-to-end transmissions.

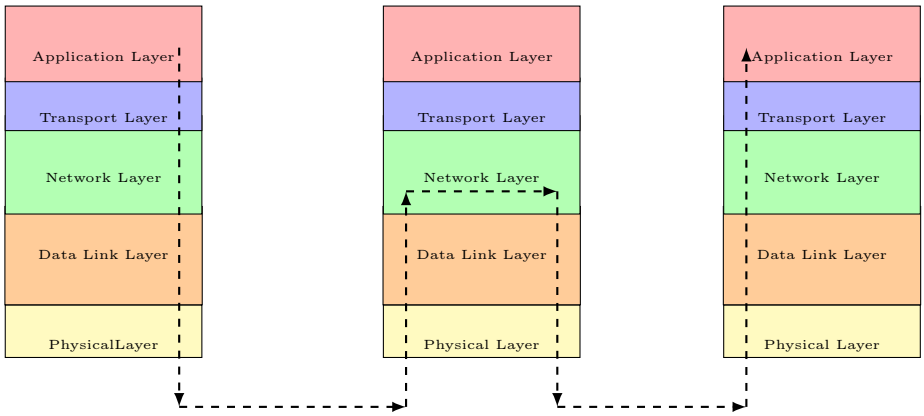


Fig. 1. WirelessHART Communication scheme

Indeed, as WirelessHART builds a mesh network, sensors are located several hops from the network manager. Thus, these sensors rely on their neighbors to forward their packets from/to the network manager. Therefore, as illustrated in Figure 1, the several forwards of packets between neighbor devices are called *the hop-by-hop transmission* and the communication between the sending sensor and the network manager is called *the end-to-end communication.*

Security at Data Link Layer: The hop-by-hop transmission security is provided by the Data Link Layer (DLL) using a cryptographic key called "Network Key" shared by all devices composing the wireless network. It defends against attackers who are outside the network and do not share its secret.

Each DLPDU is authenticated by the sending device using *the network key.* Therefore, before processing any received DLPDU, a device must check the keyed Message Integrity Code (MIC) to authenticate the identity of the sending device. We must note that the DLPDU itself is not enciphered but authenticated by a four-byte MIC generated with CCM* mode (Combined Counter with CBC-MAC) using the AES-128 block cipher.

Security at Network Layer: The end-to-end security is provided by the Network Layer (NL) using a cryptographic key called "Session Key" known only by the two communicant devices. It defends against attackers who may be on the network path between the source and the destination (Inside attacker).

The network layer also uses a keyed Message Integrity Code (MIC) for the authentication of the Network Protocol Data Unit (NPDU). Additionally, it is

used to encrypt and decrypt the NPDU payload. The end-to-end security is session oriented i.e., it provides a private and secure communication channel between a pair of network devices. Each session is defined by two elements:

- the session key: it is a dedicated 128-bits cryptographic key. It is used to encipher the NPDU payload and to authenticate the whole NPDU.
- the session counter: it is a 32 bits value that defends against replay attacks and used as the nounce for generating the NPDU MIC. Each device keeps a history of received nonce counter.

2.4 Communication scheme

WirelessHART implements unicast and broadcast communications in both the Data Link and the Network Layers. In the Data link layer, the unicast or broadcast communication is set by configuring the packet with unicast or a broadcast destination address, by using the unicast or the broadcast graph and also by using the dedicated transmission slots. Indeed, the Network Manager configures each wireless sensor to be at the beginning of each slot either a sender, a receiver or to stay idle.

As illustrated in Figure 2, when a device receives unicast packet, it starts by authenticating it in the Data link layer (DLL) using the network key and then it is transmitted to the Network layer. There, the destination NL address is checked. If it matches the device's address, the packet is authenticated a second time using the unicast session key and its payload is deciphered and sent to the Application Layer to be executed. Otherwise, the packet is sent back to the DLL to be forwarded to the next hop device.

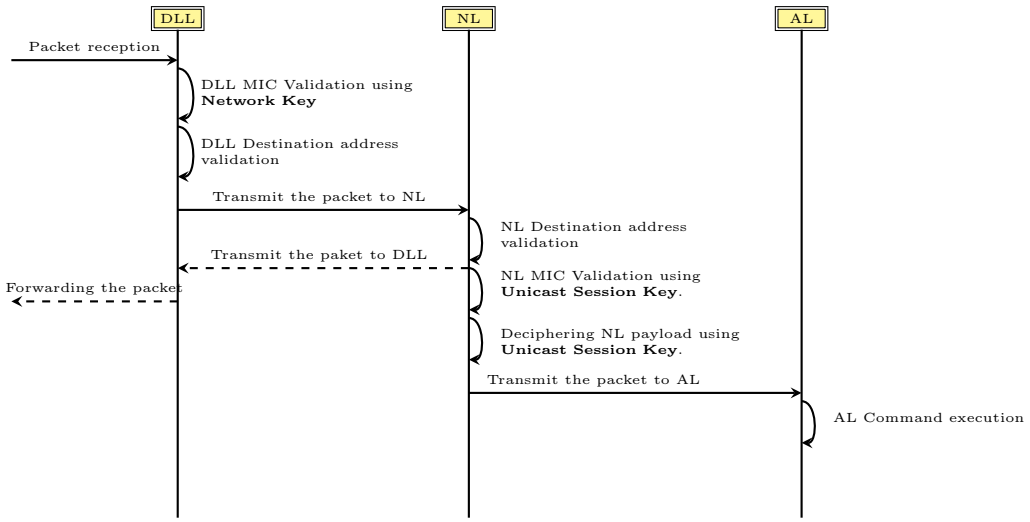


Fig. 2. Unicast packet processing sequence

In a broadcast communication, a packet sent by the Network manager is propagated to all devices in the wireless network. As illustrated in Figure 3, each time a device receives a broadcast packet, it starts by authenticating it firstly in the Data link layer (DLL) using the network key and then in the Network layer (NL) using the broadcast session key. If the packet passes authentication validations, it will be deciphered and sent to the Application Layer (AL) to be executed. A copy of the packet is also sent back to the DLL to be forwarded to other devices.

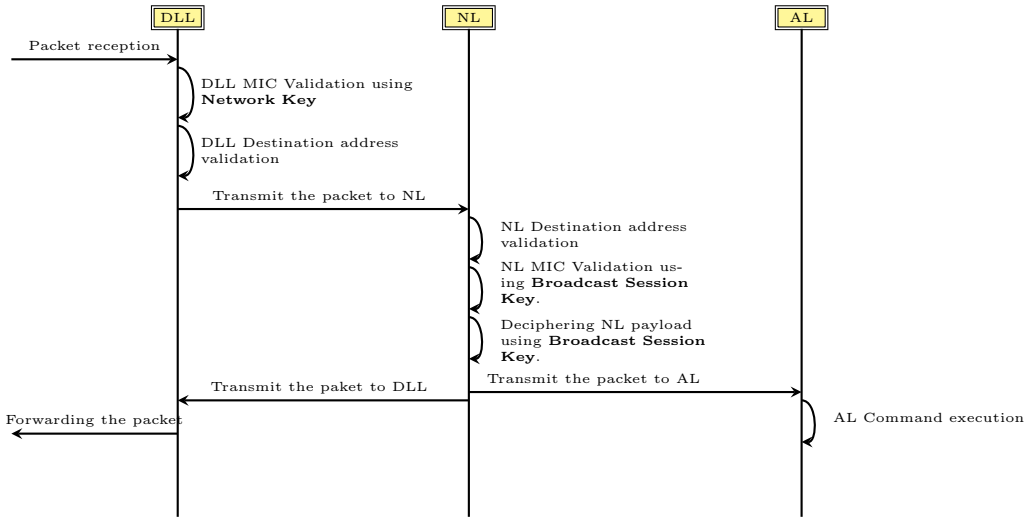


Fig. 3. Broadcast packet processing sequence

On another hand, in the Network Layer, four sessions are set up as soon as any device joins the network. They allow the transmission of sensing data from a device to the Network Manager, and the transmission of commands from the Network Manager to a field device.

1. unicast session with the NM: it is used by the network manager to manage the device.
2. broadcast session with the NM: it is used to globally manage devices. For example this can be used to roll a new network key out to all network devices. All devices in the network have the same key for this session.
3. unicast session with the Gateway: it carries normal communications (for example process data) between the gateway and the device.
4. broadcast session with the Gateway: it is used by the gateway to send the identical application data to all devices.

In addition, each device has a join session key which cannot be deleted. The Join_key is the only key that is written once connecting to the device's

maintenance port. It can also be updated by the Network Manager once the device is successfully connected. All other keys are distributed by the Network Manager.

3 Communication Scheme Attack

The idea of the attack is that a malicious insider attacker uses its own credentials to bypass the authentication mechanism and injects false command into the network. These false commands will be authenticated as legitimate commands and executed by receiving devices. Depending on the nature of injected false commands, consequences on the network can be more or less harmful.

As indicated in the previous Section, end-to-end communications are secured by session keys. In unicast communications, the session key is only known by the two communicant devices while in broadcast communications, the session key is shared by all devices connected to the network.

Therefore to launch the command injection attack, the malicious insider attacker will use Broadcast Session credentials to perform this kind of attacks. Indeed, as part of the network, the malicious node is configured with *the broadcast session key* and the *session counter*.

The command injection attack can be performed in several ways such as: a Direct command injection attack, a Bounced command injection attack and an On-the-fly command injection attack.

3.1 Scenario 1: Direct Command injection attack

In a Direct Command Injection Attack a malicious insider node forges a fake broadcast packet and forwards it to its neighbors.

As illustrated in Figure 4, at the moment T the malicious node *Device*5 uses its knowledge on the broadcast session credential i.e., the broadcast session key and the session counter, to forge a broadcast packet. The source address in the NL is set to the Network Manager address and the destination addresses in both network and data link layers are set to the broadcast address. The malicious insider node will send the forged packet using its own broadcast link in the same way as if it was a legitimate packet sent by the network manager. Receiving nodes, *Device*8 and *Device*9, will authenticate the packet using the broadcast session key and execute the injected false command.

Using this attack, a malicious insider node can inject any false command and send it to its neighbors using the broadcast graph.

3.2 Scenario 2: Bounced Command injection attack

In WirelessHART both DLL and NL destination addresses can be either unicast or broadcast addresses and all combinations are allowed. So, a packet can have unicast DLL destination address and a broadcast NL destination address.

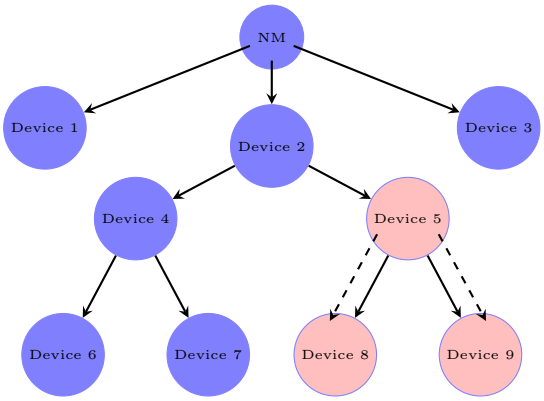


Fig. 4. Direct Broadcast attack

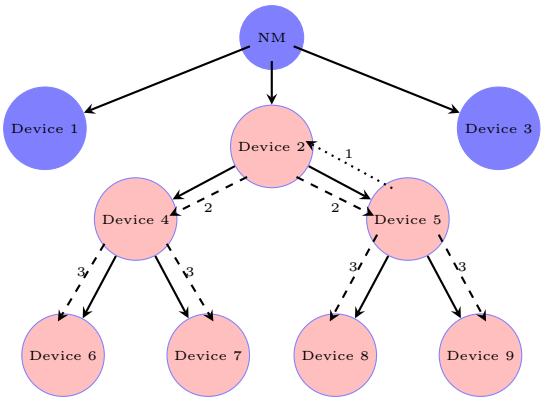


Fig. 5. Bounced Broadcast attack

In a Bounced Command Injection Attack a malicious insider node forges a fake broadcast packet and sends it to its parent node. As illustrated in Figure 5, this kind of attacks is composed of the following steps:

1. At the moment T the malicious node *Device*5 uses its knowledge of the broadcast session credential i.e., the broadcast session key and the session counter, to forge a broadcast packet. The source address in the NL is set to the Network Manager address and the NL destination address is set to the broadcast address.
   In the DLL, the source address is set to the *Device*5 address and the destination address is set to its parent's address i.e., *Device*2. The malicious insider node will send the forged packet using its own normal link between itself and the parent node.
2. The receiving node *Device*2 authenticates the packet in the DLL as a legitimate unicast packet and transmitted it to the upper layer.

In the NL, the packet is identified as a broadcast packet sent by the Network Manager. It is authenticated and deciphered using the broadcast session key. The packet is then transmitted to the application layer to be executed.
A copy of the packet is also transmitted to the DLL to be forwarded to *Device*2 neighbors i.e., *Device*4 and *Device*5.

3. Both *Device*4 and *Device*5 process the received packet as a legitimate broadcast packet sent by the Network manager and propagate it to their neighbors.
4. As results, the injected false command packet is received and executed by *Device*2, *Device*4, *Device*5, *Device*6, *Device*7, *Device*8 and *Device*9.

This scenario allows a malicious insider node by using its parent node as a relay to increase the impact of the attack. By this way, the injected false command is propagated to all parent node's children.

3.3 Scenario 3: On-the-fly Command injection attack

In an On-the-fly command injection attack, a malicious insider node that receives a broadcast packet, will forward to its neighbors a modified version of the received packet.

As illustrated in Figure 6, this attack is performed according to the following steps:

1. The Network Manager sends a broadcast packet.
2. The broadcast packet is forwarded to devices and received by the malicious insider node *Device*5.
3. All receiving node execute the command sent by the network manager and forward it to devices in their neighborhood.
4. The malicious node *Device*5 uses its knowledge of broadcast session credential i.e., the session key and the session counter, to modify the received broadcast packet and send it to its neighbors.
5. As results, the injected false command packet is received and executed by *Device*8 and *Device*9.

As in the direct command injection attack, a malicious insider node can inject any false command and sent it to its neighbors using the broadcast graph. The difference is that an on-the-fly injection command attack is a stealth attack as the injected packet is hidden inside a legitimate communication flow.

3.4 Discussion

Described scenarios showed the feasibility of the broadcast attack and that it can be performed in several ways. We must note that although we can launch the attack at any chosen time T, the malicious node must wait for an appropriate time slot to be able to send the forged packet. For example in the case of the direct command injection, the malicious node must wait for the next broadcast slot to send the false command to its neighbors. But as all devices are configured

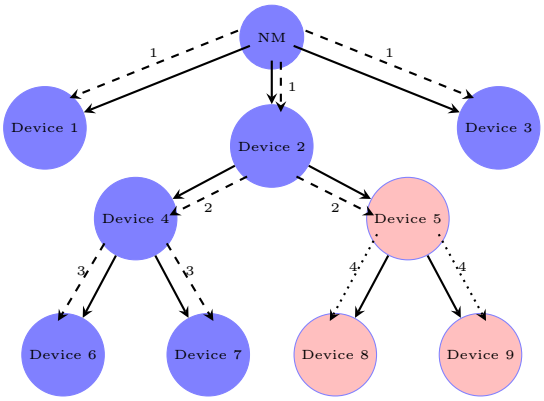


Fig. 6. On-the-fly Broadcast attack

with this kind of slots, it is always possible for a malicious node to send its false command. According to the WirelessHART [2], by default each device is configured with one sending unicast slot and one sending broadcast slot each 1 minute. Thus, the average waiting time T_{Avg} between the attack launching time and the false command injection time is: $T_{Avg} = T_{sending\_broadcast}/2 = 30s$ in the case of a direct attack and $T_{Avg} = T_{sending\_unicast}/2 + T_{sending\_broadcast}/2 = 60s$ for a bounced attack. The on-the-fly attack duration depends on the industrial process and broadcast commands sending frequency. In average, this frequency is around 1 hour.

By comparing the 3 scenarios, we can see that the bounced command injection increases the spreading area of the attack by using the parent of the malicious node as a relay. Also, the on-the-fly command injection attack is interesting as it hides the attacks inside a legitimate flow. Nevertheless, the drawback of this attack is that the malicious node must wait to the transmission by the network manager of a broadcast packet which can take a long time to happen.

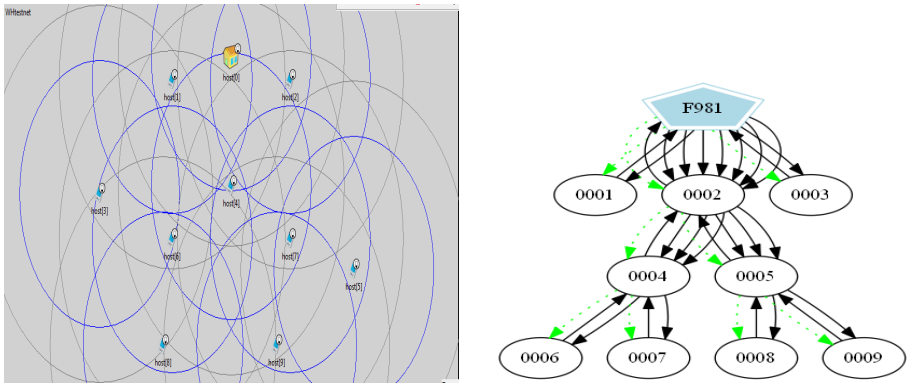
Finally, we must note that in all these scenarios, the malicious insider node has the choice between executing or not the injected false command. Indeed, depending on the attack's goal, the malicious node can launch the attack with or without executing it. For example, by not executing the false command, the malicious node can mislead administrators in their investigations to discover the origin of the network disturbances.

4 Attack implementation

To test the broadcast attack, we use WirelessHART NetSIM [4], a simulator that we develop for assessing the security of WirelessHART SCADA-based systems.

As illustrated in Figure 7(a), the simulated wireless network is composed of a network manager and 9 wireless sensors. Wireless sensors are configured to send periodically each 4s simulated sensing data to the Network Manager. Figure 7(b)

illustrates the routing graphs. The broadcast graph is indicated by dotted green arrows.



(a) Physical topology (b) Logical topology

Fig. 7. Simulation network topology

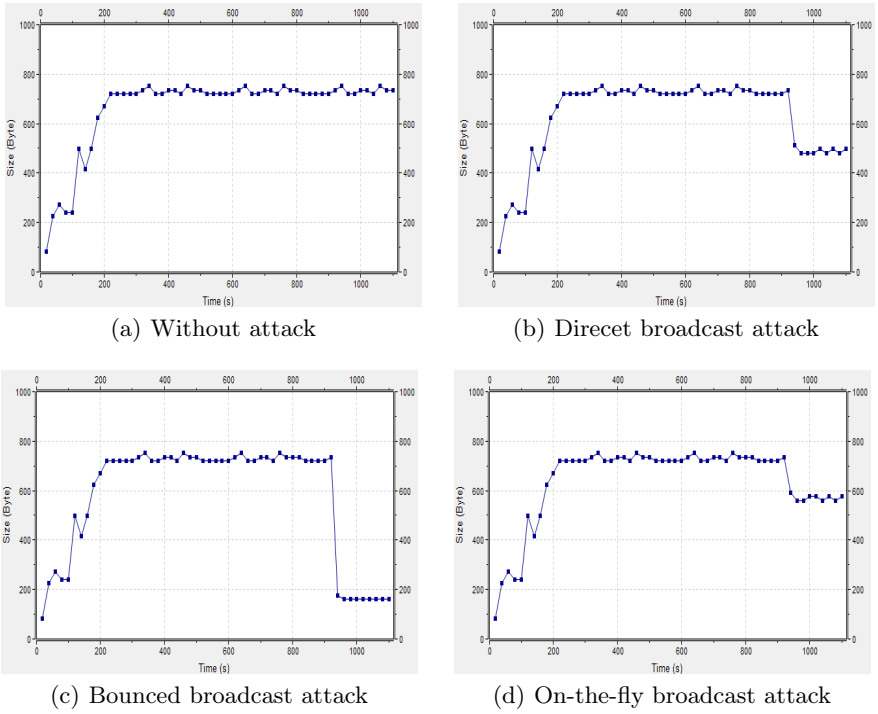
For testing the three scenarios, we launched the broadcast attack at $T = 800s$ and the *Device*5 is configured to be the malicious insider attacker. The injected false command is the command 961 that is used to set a new *network key*. This command has 2 parameters: *the new network key*, and T' the time when it will be changed. In all the three scenarios $T' = 920s$.

As illustrated in Figure 8(a) i.e., in the normal case, the size of sensing data received by the Network Manager is about 720 bytes each 4s. We observe that for the three scenarios of the broadcast attack, the size of received data by the Network Manager falls immediately at $T = 920$. This indicates that the Network manager stops receiving sensing data from some wireless sensors.

Indeed at $T = 920$ *infected devices* will execute the injected false command and start to use the received network key to calculate the DLL MIC. When received by a device that has not been infected by the attack, the packet do not pass the MIC validation step and is rejected. Consequently, packet sent by *infected devices* will be rejected and not received by the Network Manager.

In comparison with the normal case, in the direct command injection attack the data received by the Network Manager, illustrated in Figure 8(b), falls from 720 bytes to 480 bytes. This represents a decrease of 33%. Indeed, 3 devices i.e., *Device*5, *Device*8 and *Device*9, are infected by this attack.

In the case of the bounced command injection attack, shown in Figure 8(c), we record a decrease of 77% in the data received by the Network Manager. This indicates that this kind of attacks, allows a malicious node to use its parent device as a relay to propagate the attack to a great number of devices. As result, 7 devices are infected by the attack, i.e, *Device*5, *Device*2, *Device*4, *Device*6, *Device*7, *Device*8 and *Device*9.



(a) Without attack (b) Direcet broadcast attack

(c) Bounced broadcast attack (d) On-the-fly broadcast attack

Fig. 8. Sensing data received by the Network Manager.

In the on-the-fly command injection attack, we configure the Network Manager to broadcast, at $T = 800s$ to all devices, a command to change the network key at $T' = 920s$. The malicious attacker will modify this command and send a false command to its children devices. This attack has the same impact as in the case of a direct command injection command. As a variant, we choose that the malicious node does not execute the false command, which explains the difference of the impact between the direct and on-the-fly broadcast attacks. As indicated in Figure 8(d), the received data by the network manager decreased by 22% as only 2 devices are infected i.e., *Device*5 and *Device*6.

5 Countermeasures discussion

The broadcast communication is an important feature in WirelessHART. It allows the Network Manager to configure all devices composing the wireless network by only sending a single packet. It avoids a costing time and resources process of sending a single packet to each device. But as shown in this paper, this feature creates a dangerous breach in the communication scheme security. As it is complicated to ban broadcast communications, we propose hereafter, some ideas to reduce the exposition to this vulnerability.

- Broadcast packet validation after the reception of 2 identical packets: this condition aims to stop direct and on-the-fly command injections. Indeed, as WirelessHART builds a meshed network, best practices in industrial sensor networks recommends that each node has at least 2 or 3 parents. Consequently, each sensor will receive the broadcast packet more than once. Thus, according to this rule, each node must wait till the reception of the same packet from another of its parents before it executes and forwards it. Nodes located at one hop do not have to apply this rule as they receive the broadcast packet directly from the Network Manager. This countermeasure adds a latency in the transmission of broadcast packets and can, in some cases, block their forwarding.
- DLL and NL addresses validation: in the case of the bounced command injection attack, DLL and NL headers of the injected packet indicate contradictory informations. Indeed, the source address in the DLL header indicates that the packet has been sent by a children node i.e., the malicious node, while the source address in the NL header indicates that the packet has been sent by a parent node i.e., the network manager. Therefore, implementing in the NL a security mechanisms that rejects packets indicating such contradictory informations can mitigate this kind of attacks. We must note that even if this solution do not complain with the layer separation principle, in practice WirelessHART layers already use information provided by other layers such as addresses.
- Use of an IDS for monitoring node's behavior: indeed, except rethinking deeply the communication scheme of WirelessHART, as implementing asymmetric cryptography for packet's authentication, that is a costly process, the use of an IDS will increase significantly the security of such networks. Indeed, this kind of system by monitoring exchanged packets, are able to detect the injection of a false packet or the modification of a packet during its transmission.

In conclusion, the two first countermeasures are partial solutions that do not prevent all scenarios. The second solution is the costless one as it adds a reduced overhead. The use of an IDS is the more efficient solution. Indeed, although it requires the installation of dedicated equipments for traffic monitoring, it is the only solution that detects all possible scenarios. Nevertheless, given that WSNs are distributed systems, we must pay attention to the scheme used to deploy the IDS as it directly impacts the information gathering capability.

6 Related Works

Most of dedicated studies on WirelessHART focus mainly on the evaluation of the performances of this protocol and its capabilities to operate in an industrial environment and its capacity to meet real-time requirements [5,6,7].

On the other hand, security analysis conducted on WirelessHART are based on the specifications of the standard without conducting any tests. Thus, in [8]

Raza and al. discuss the strengths and the weaknesses of security mechanisms and analyze them against the well known threats in the wireless sensor networks. They conclude that WirelessHART is strong enough to be used in the industrial process control environment. Alcazar and Lopez identify in [9] vulnerabilities and threats in several wireless communication protocols used in industry i.e., ZigBee PRO, WirelessHART and ISA100.11.a. They analyze in detail the security features of each of these protocols. For them, WirelessHART offers strong authentication capabilities before and after deployment. However, they recommend to add a rekeying process to WirelessHART to enforce its resilience to sniffing attacks and thereby key disclosure.

But although WirelessHART implements several security mechanisms, it stays vulnerable to dangerous attacks. Thus, in a previous work, we develop WirelessHART NetSIM [4], a simulator for assessing the security of WirelessHART SCADA-based systems. It can be used to test attacks and countermeasures on WSN. It includes several scenarios for testing simple and complex kinds of attacks. Using this simulator, we give the first description of a Sybil attack specially tailored to target WirelessHART based SCADA systems [10]. We demonstrate that an insider attacker using this weakness can isolate partially or more again totally wireless sensors from the SCADA network. This attack targets the security authentication in the data link layer and is based on the knowledge of the network key by all devices composing the wireless network.

Nevertheless, this attack do not allow the injection of false commands into the network as security mechanisms in the upper layer will stop injection attacks. Therefore, the presented attack in this paper is more dangerous than the previous one, as it permits the injections of any false command by circumventing security mechanisms implemented in the Network Layer.

7 Conclusion

In this paper, we analyze the security of the communication scheme in WirelessHART, the most widely used wireless protocol in SCADA systems. We show that an insider attacker can bypass implemented security mechanisms and inject false commands into the network. These false commands will be authenticated as legitimate commands and executed by receiving devices.

The attack is based on the use of *the broadcast session* credentials that are shared by all devices composing the wireless network. We give also the description of three different scenarios that exploit this weakness. Tests conducted, using a simulator dedicated to WirelessHART security assessment, confirm the feasibility of the attack and its potentially harmful impact. In these tests we choose the network key change command as injected false command. By this way, we were able to break the reception by the network manager of sensing data from wireless sensors. Other scenarios can be developed to take advantage of this weakness. For example, the source routing method can be used to inject false commands to a greater number of sensors.

On the other hand, proposed solutions do not totally mitigate the broadcast attack. Indeed, the broadcast communication is an important feature that cannot be removed. Therefore, and except changing deeply the communication scheme implemented by WirelessHART, the use of an Intrusion Detection System (IDS) is the best operational manner to detect and mitigate this kind of attacks. Further research must be made to study the best way to apply IDS to WSN in industrial environments.

References

1. Huang, Y.L., Cárdenas, A., Amin, S., Lin, Z.S., Tsai, H.Y., Sastry, S.: Understanding the physical and economic consequences of attacks on control systems. International Journal of Critical Infrastructure Protection **2**(3) (2009) 73 – 83
2. HART Communication Foundation: WirelessHART. `http://www.hartcomm.org`
3. IEEE: IEEE 802.15.4-2006: Standard for Local and metropolitan area networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). `http://www.ieee.org`
4. Bayou, L., Espes, D., Cuppens-Boulahia, N., Cuppens, F.: WirelessHART NetSIM: A WirelessHART SCADA-Based Wireless Sensor Networks Simulator. In Bécue, A., Cuppens-Boulahia, N., Cuppens, F., Katsikas, S.K., Lambrinoudakis, C., eds.: Security of Industrial Control Systems and Cyber Physical Systems - First Workshop, CyberICS 2015 and First Workshop, WOS-CPS 2015, Vienna, Austria, September 21-22, 2015, Revised Selected Papers. Volume 9588 of Lecture Notes in Computer Science., Springer (2015) 63–78
5. Han, S., Zhu, X., Mok, A.K., Chen, D., Nixon, M.: Reliable and real-time communication in industrial wireless mesh networks. In: 17th IEEE RTAS, USA, IEEE Computer Society (2011) 3–12
6. Kim, A.N., Hekland, F., Petersen, S., Doyle, P.: When HART goes wireless: Understanding and implementing the wirelesshart standard. In: Proceedings of 13th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Hamburg, Germany, IEEE (2008) 899–907
7. Song, J., Han, S., Mok, A., Chen, D., Lucas, M., Nixon, M.: Wirelesshart: Applying wireless technology in real-time industrial process control. In: Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS '08. IEEE. (April 2008) 377–386
8. Raza, S., Slabbert, A., Voigt, T., Landernäs, K.: Security considerations for the wirelesshart protocol. In: Proceedings of 12th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Spain, IEEE (2009) 1–8
9. Alcaraz, C., Lopez, J.: A security analysis for wireless sensor mesh networks in highly critical systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C **40**(4) (2010) 419–428
10. Bayou, L., Espes, D., Cuppens-Boulahia, N., Cuppens, F.: Security issue of wirelesshart based SCADA systems. In Lambrinoudakis, C., Gabillon, A., eds.: Risks and Security of Internet and Systems - 10th International Conference, CRiSIS 2015, Mytilene, Lesbos Island, Greece, July 20-22, 2015, Revised Selected Papers. Volume 9572 of Lecture Notes in Computer Science., Springer (2015) 225–241